Channel Assignment Using Genetic Algorithm Based on Geometric Symmetry

Sasthi C. Ghosh, Bhabani P. Sinha, *Fellow, IEEE*, and Nabanita Das, *Member, IEEE*

***Abstract*—This paper deals with the channel assignment problem in a hexagonal cellular network with two-band buffering, where the channel interference does not extend beyond two cells. Here, for cellular networks with homogeneous demands, we find some lower bounds on minimum bandwidth required for various relative values of s_0, s_1, and s_2, the minimum frequency separations to avoid interference for calls in the same cell, or in cells at distances of one and two, respectively. We then present an algorithm for solving the channel assignment problem in its general form using the elitist model of genetic algorithm (EGA). We next apply this technique to the special case of hexagonal cellular networks with two-band buffering. For homogeneous demands, we apply EGA for assigning channels to a small subset of nodes and then extend it for the entire cellular network, which ensures faster convergence. Moreover, we show that our approach is also applicable to cases of nonhomogeneous demands. Application of our proposed methodology to well-known benchmark problems generates optimal results within a reasonable computing time.**

***Index Terms*—Two-band buffering, cellular network, channel assignment problem, genetic algorithm, optimal bandwidth.**

I. Introduction

When a mobile cellular network is designed, each cell of the network is assigned a set of channels to provide services to the individual calls of the cell. The task of assigning frequency channels to the cells satisfying some frequency separation constraints with a view to avoiding channel interference and using as small bandwidth as possible is known as the *channel assignment problem*. For a network, the available radio spectrum is divided into nonoverlapping frequency bands. We assume that the frequency bands are of equal length and are numbered as 0, 1, 2,... from the lower end. Each such frequency band is termed a *channel*. In this context, the terms *channel assignment* and *frequency assignment* will be used interchangeably in our discussions. The highest numbered channel required in an assignment problem is termed the *required bandwidth.* We are considering here the static model of the channel assignment problem, where the number of calls to each cell is known a priori. Three types of interference [23] are generally taken into consideration in the form of constraints:

1) *cochannel constraint*, due to which the same channel is not allowed to be assigned to certain pairs of cells simultaneously;
2) *adjacent channel constraint,* for which adjacent channels are not allowed to be assigned to certain pairs of cells simultaneously;
3) *cosite constraint,* which implies that any pair of channels assigned to the same cell must be separated by a certain number.

In its most general form, the channel assignment problem (CAP) is equivalent to the generalized graph-coloring problem, which is a well-known NP-complete problem [2]. As a result, earlier researchers attempted to solve the problem from a graph theoretic view point and proposed many heuristics [9], [25]. Later improved approximate algorithms using neural networks, simulated annealing, and genetic algorithms have been proposed to solve this problem. In the neural network approach [6], [22], [26], [28], an inherent disadvantage is that it easily converges to local optima and hence optimal solutions cannot always be guaranteed. The simulated annealing approach [4], [5] guarantees global optimal solution asymptotically, but the rate of convergence is rather slow. The genetic algorithm approach [18], [19], [21], however, provides a global optimal solution with a relatively faster rate of convergence.

Earlier works on approximate algorithms for channel assignment can be classified into two categories. One of these [9], [25] first determines an ordered list of all calls and then assigns channels deterministically to the calls so as to minimize the required bandwidth. Given the bandwidth of the system, the other category [4]–[6], [19], [21], [22], [26], [28] formulates a cost function, such as the number of interference constraints violated by a given channel assignment, and then tries to minimize this cost function. The advantage of the first category of algorithms is that the derived channel assignment always fulfills all the interference constraints for a given demand; but it may be hard to find an optimal solution in the case of large and difficult problems, even with quite powerful optimization tools. On the other hand, for the second group of algorithms, it seems to be impossible to minimize the cost function to the desired value of zero with the minimum number of channels, in the case of very hard problems.

In [18], the authors combined both of the above methods and proposed the combined genetic algorithm (CGA), which generates a call list in each iteration and evaluates the quality of the generated call list following the frequency exhaustive assignment strategy. The authors started the procedure by estimating the lower bound Z on bandwidth [7], [8], [23]. If the algorithm does not find a solution with Z, the value of Z is incremented by one and the CGA algorithm is repeated until a valid solution is derived. Thus, in this approach, the computation time will be highly dependent on the proximity of the prior estimation of the

Manuscript received November 29, 2001; revised May 31, 2002, August 22, 2002, and October 4, 2002.

The authors are with the Advanced Computing and Microelectronics Unit, Indian Statistical Institute, 700 108 Kolkata, India (e-mail: sasthi_r@isical.ac.in; bhabani@isical.ac.in; ndas@isical.ac.in).

lower bound on bandwidth to its optimal value. But the problem of finding an achievable lower bound is itself *NP-complete* [26].

In [7], Gamst presented some lower bounds on the bandwidth for channel assignment problems in general. Improving the results by Gamst in [8], Tcha *et al.* presented some new results on the lower bound on bandwidth. The authors in [1] proposed some new lower bounds on channel bandwidth, taking the regular geometry of the cellular network [11]–[14] into account. They considered hexagonal cellular networks where every cell has a demand of only one channel with *two-band buffering,* i.e., the channel interference does not extend beyond two cells, with s_0, s_1, and s_2 as the minimum frequency separations between the calls in the same cell and in cells at distances one and two apart respectively.

In this paper, we present an algorithm for solving the channel assignment problem in its general form using the elitist model of genetic algorithm (EGA) [3]. We then show how this general approach can conveniently be applied to the special case of the network model of a hexagonal cellular structure with two-band buffering, on which most of the known benchmark instances (including the Philadelphia problems) of the CAP have been defined. Our proposed approach falls in the first category of approximate algorithms discussed above, which do not depend on a prior estimation of the lower bound on the bandwidth. However, in order to judge the performance of our proposed algorithm, we have also found new lower bounds on the bandwidth for a hexagonal cellular network with homogeneous demands (where w_i, the number of channels required for cell i, is same for all i) and two-band buffering. These lower bounds have been based on different relative values of s_1 and s_2, and for $w_i = 1$, these bounds are improved over that in [1]. It may also be noted that these new lower bounds are either equal to or tighter than those in [7], [8], and [23] when applied to this special case of hexagonal cellular networks with homogeneous demand and two-band buffering.

For hexagonal cellular networks with homogeneous demand of w channels per cell, our approach essentially selects a small subset of cells of the network, on which we apply the EGA to find its assignment and next repeat the assignment for the whole network. As a result of this, the proposed technique has a faster rate of convergence. For $w = 1$, our approach improves the bandwidth requirement by 25% at best over that in [1]. We then show how our method can also be used to solve the channel assignment problem with nonhomogeneous demand vector $W = (w_i)$ on these networks. Most interestingly, it shows that in some cases, depending on the relative values of s_0, s_1, and s_2, the required bandwidth is mainly determined by the maximum demand $w_{\max}$ in W. It reveals the fact that in terms of bandwidth requirement, the case is almost equivalent with the cellular networks having homogeneous demand $w_{\max}$. Application of our approach to well-known benchmark instances (including the most difficult two) always results in assignments with the optimal bandwidth, in a reasonable amount of computation time on a DEC Alpha station.

The rest of this paper is organized as follows. Section II describes the general model of the cellular network. Section III presents new lower bounds on the required bandwidth for homogeneous demands. Section IV includes the genetic algorithm

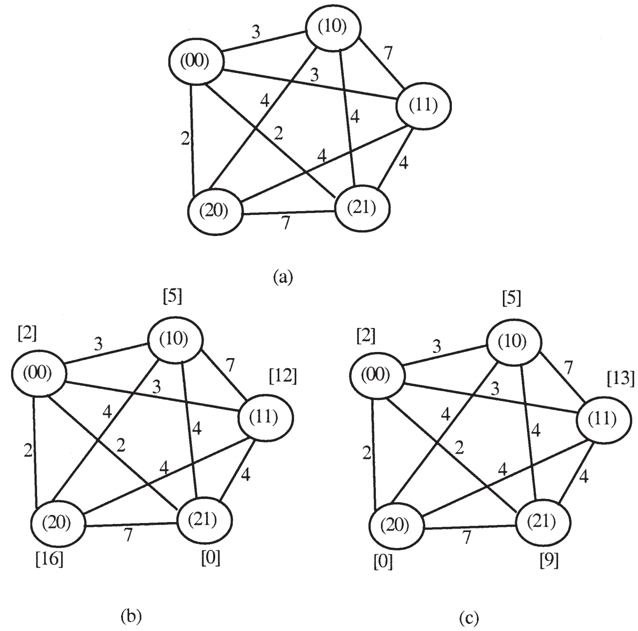


Fig. 1. (a) A typical CAP graph and (b)–(c) two different frequency assignments on it.

implementation of the problem in its most general form. In Section V, we show how this general technique can effectively be applied to the special case of hexagonal cellular network with two-band buffering. In this section, we propose several schemes for assigning channels to the cells of the entire network. In Section VI, we consider nonhomogeneous channel demands and report the results of our approach on well-known benchmark problems. Section VII concludes this paper.

II. Model of the Cellular Network

We have used here the same model as described in [1], [9], and [15]. This model is described by the following components.

1) The number of distinct cells, say, n, with cell numbers as $0, 1, \ldots, n-1$.
2) A demand vector $W = (w_i)$, $(0 \le i \le n-1)$, where w_i represents the number of channels required for cell i.
3) A frequency separation matrix $C = (c_{ij})$, where c_{ij} represents the minimum frequency separation requirement between a call in cell i and a call in cell j, $0 \le i, j \le n-1$.
4) A frequency assignment matrix $\Phi = (\phi_{ij})$, where ϕ_{ij} represents the frequency assigned to call j in cell i, $0 \le i \le n-1, 0 \le j \le w_i - 1$. The assigned frequencies ϕ_{ij}s are assumed to be evenly spaced and can be represented by integers ≥ 0.
5) A set of frequency separation constraints specified by the frequency separation matrix:
$|\phi_{ik} - \phi_{jl}| \ge c_{ij}$ for all i, j, k, l (except when both $i = j$ and $k = l$).

The goal of the channel assignment problem is to assign frequencies to the cells satisfying the frequency separation constraints as specified by component 5) above, in such a way that the required system bandwidth becomes optimal.

Each call to a cell is represented as a node of a graph, and the nodes v_i and v_j are connected by an edge with weight c_{ij}

if $c_{ij} > 0$. We call this graph a CAP graph following the terminology in [1]. In our model, we assume that the channels are assigned to the nodes of the CAP graph in a specific order and that a node will be assigned the channel corresponding to the smallest integer that will satisfy the frequency separation constraints with all the previously assigned nodes.

Example 1: Fig. 1(a) shows a CAP graph with three cells having demands $w_0 = 1$, $w_1 = 2$, and $w_2 = 2$, respectively. Each node in Fig. 1 is labeled as (rs), where r is the cell number at which a call is generated and s is the call number to this cell r. That is, node (10) represents call 0 in cell 1. The frequency separation requirements for this example is given by the following matrix:

$$C = \begin{array}{c|ccc} cell \quad no. \rightarrow & 0 & 1 & 2 \\ \downarrow & & & \\ 0 & 7 & 3 & 2 \\ 1 & 3 & 7 & 4 \\ 2 & 2 & 4 & 7. \end{array}$$

The edges of the CAP graph are labeled with weights according to matrix C. The label $[\alpha]$ associated with each node of the CAP graph of Fig. 1(b) and (c) indicates that the frequency channel α is assigned to that node. Now, if the channels are assigned to nodes in the order ((21), (00), (10), (11), (20)), as shown in Fig. 1(b), the minimum bandwidth required will be 16. But, if the channels are assigned to nodes in the order ((20), (00), (10), (21), (11)), as shown in Fig. 1(c), the minimum bandwidth required will be just 13.

It is clear from the above example that the ordering of the nodes has strong impact on the required channel bandwidth. Suppose there are m nodes in the CAP graph. Therefore, the nodes can be ordered in $m!$ ways and, hence, for sufficiently large m, it is impractical to find the best ordering by an exhaustive search. Instead, we use the genetic algorithm approach to find an optimal or near-optimal solution to the problem.

III. Lower Bounds on Bandwidth for Homogeneous Demand

The cellular graph is a graph where each cell of the cellular network is represented by a node and two nodes have an edge between them if the corresponding cells are adjacent to each other (i.e., when the two cell boundaries share a common segment) [14]. Note that this cellular graph simply represents the topology of the cellular structure, without any regard to the demand per cell, and is different from the CAP graph mentioned above. We assume that the cellular graph is of hexagonal structure with two-band buffering, i.e., the interference extends only up to two cells from the call originating cell.

Let us now consider a seven-node subgraph of the cellular graph as shown in Fig. 2. Every node in this subgraph is within at distance of two from each other. Therefore, they are going to interfere with each other in their frequency assignment. In other words, no frequency reuse is possible within this subgraph. Hence, the bandwidth requirement of this subgraph will give a lower bound on the bandwidth requirement of the whole cellular network.

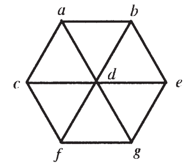


Fig. 2. Seven-node subgraph of hexagonal cellular network.

In all our later discussions, a frequency $(is_1 + js_2)$ assigned to a node will be denoted by a two-tuple (i, j). Similarly, a frequency $(is_0 + js_1 + ks_2)$ will be denoted by a three-tuple (i, j, k). We now state the following result on the minimum bandwidth requirement for assigning channels to this subgraph.

Lemma 1: The minimum bandwidth required for assigning channels to the seven-node subgraph of a hexagonal cellular network having homogeneous demand of a single channel and two-band buffering with frequency separation $s_1 \geq s_2$ is $(s_1 + 5s_2)$ when $s_2 \leq s_1 \leq 2s_2$ and $(2s_1 + 3s_2)$ when $s_1 \geq 2s_2$.

Proof: See Appendix I.

Remark: The results of Lemma 1 show that the minimum bandwidth requirement for assigning channels to the seven-node subgraph of Fig. 2 is actually lower (except when $s_1 = s_2$) than $(2s_1 + 4s_2)$ reported in [1].

We now generalize Lemma 1 and state the following two theorems on the minimum bandwidth requirement for assigning w (≥ 2) frequency channels to each of the nodes of the subgraph.

Theorem 1: The minimum bandwidth required for assigning channels to the seven-node subgraph of a hexagonal cellular network with homogeneous demand w (≥ 2) and two-band buffering with frequency separation $s_2 \leq s_1 \leq 2s_2$ is:

1) $(2s_1 + 5s_2) + (w-2)(s_0 + 6s_2) + 6s_2$, when $s_1 \leq s_0 \leq (2s_1 - s_2)$;
2) $(w-1)(2s_1 + 5s_2) + 6s_2$, when $(2s_1 - s_2) \leq s_0 \leq 6s_2$;
3) $(w-1)(2s_1 + 5s_2) + s_0$, when $6s_2 \leq s_0 \leq (s_1 + 5s_2)$;
4) $(w-1)(2s_1 + 5s_2) + (s_1 + 5s_2)$, when $(s_1 + 5s_2) \leq s_0 \leq (2s_1 + 5s_2)$;
5) $(w-1)s_0 + (s_1 + 5s_2)$, when $s_0 \geq (2s_1 + 5s_2)$.

Proof: See Appendix II.

Theorem 2: The minimum bandwidth required for assigning channels to the seven-node subgraph of a hexagonal cellular network with homogeneous demand w (≥ 2) and two-band buffering with frequency separation $s_1 \geq 2s_2$ is:

1)
   a) $(3s_1 + 3s_2) + (w-2)(s_0 + 6s_2) + 6s_2$, when $s_1 \leq s_0 \leq 3s_2$;
   b) $(3s_1 + 3s_2) + (w-2)(3s_0) + 2s_0$, when $3s_2 \leq s_0 \leq (s_1 + s_2)$;
2) $(w-1)(3s_1 + 3s_2) + (2s_1 + 2s_2)$, when $(s_1 + s_2) \leq s_0 \leq (2s_1 + 2s_2)$;
3) $(w-1)(3s_1 + 3s_2) + s_0$, when $(2s_1 + 2s_2) \leq s_0 \leq (2s_1 + 3s_2)$;
4) $(w-1)(3s_1 + 3s_2) + (2s_1 + 3s_2)$, when $(2s_1 + 3s_2) \leq s_0 \leq (3s_1 + 3s_2)$;
5) $(w-1)s_0 + (2s_1 + 3s_2)$, when $s_0 \geq (3s_1 + 3s_2)$.

Proof: See Appendix III.

Note that the results presented in the theorems above assumed homogeneous demand. However, in Section VI, we show that some nonhomogeneous cases also can be solved by applying the technique for homogeneous demand. For those cases, these lower bounds help us to check the optimality of the solutions achieved. It may be noted that the lower bounds derived above are either exactly equal to or tighter than those in [7], [8], and [23] in the special cases of hexagonal cellular networks with homogeneous demand and two-band buffering.

IV. Genetic Algorithm for Channel Assignment

While solving an optimization problem using the genetic algorithm (GA), it is required that the parameter set of the optimization problem be coded as a finite-length string or chromosome over some finite alphabet Q [16]. A collection of M (finite) such strings or chromosomes is called a *population*. A simple genetic algorithm is composed of three basic operators: 1) reproduction or selection, 2) crossover, and 3) mutation [16]. GA starts with an initial population (randomly generated). In each iteration, a new (hopefully improved) population of the same size is generated from the current population by applying the above-mentioned three operators on the strings of the current population. The newly generated population is then used to generate the next (possibly improved) population, and so on.

Let S_b be the best string (with respect to the fitness value) of the population generated up to iteration t. In the EGA, if S_b or any string better than S_b is not in the population generated in iteration (t+1), then include S_b in the (t+1)st population [16]. We apply this technique for solving the channel assignment problem, which ensures that in successive iterations the population is improved.

A. Problem Formulation

Let us assume that the channel assignment problem is represented by a CAP graph and the frequency separation constraints are given by the matrix $C = (c_{ij})$, as described in Section II. Suppose the CAP graph has n nodes. A random order of these nodes is considered as a string S or chromosome. For example, $S = ((20), (21), (10), (11), (00))$ is a string corresponding to the CAP graph of Fig. 1(a). Let M be the population size (we have taken M as an even integer). Let *cp* be the crossover probability. We set *cp* to a high value, say, 0.95, in our algorithm.

Let q be the mutation probability and T be the total number of iterations. Usually, T is a very large positive integer. We divide the total number of iterations into five equal intervals (assuming that T is a multiple of five) as $[0, T/5]$, $[T/5, 2T/5]$, $[2T/5, 3T/5]$, $[3T/5, \text{and}, 4T/5][4T/5, T]$. We start with a mutation probability of $q = 0.5$. It is then varied with the number of iterations in the way shown in Fig. 3. The way in which the mutation probability is varied here is similar to that used in [17]. The reason for varying the mutation probability q in this fashion is that, to maintain the diversity of the population, we are to increase the value of q. Similarly, as the optimal string is approached, q is to be reduced. Fig. 3 shows one such technique for varying q with the number of iterations.

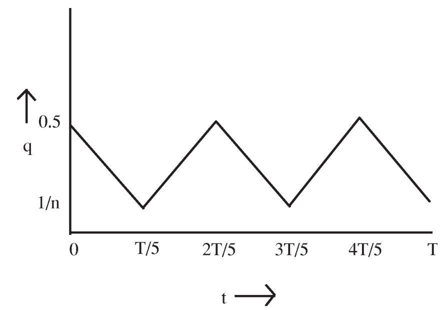


Fig. 3. Variation of mutation probability (q) with number of iterations (t).

The mutation probability at the tth iteration is given by the following function.

```
Function m_probability (int t)
begin
  W ← 5(1 − 0.5n)/(nT)
  for i = 0 to 2 do
    begin
    V ← 2i + 1
    if (t ≤ VT/5) do
    begin
      U ← t − (V − 1)T/5
      return (0.5 + WU)
      break
    end;
    if (t ≤ (V + 1)T/5) do
    begin
      U ← (V + 1)T/5 − t
      return (0.5 + WU)
      break
    end;
  end
end.
```

The fitness function $Fit(S)$ used in our algorithm is described by the following function.

```
function Fit(S) // S is a string.//
begin
   t[0] ← 0
   // t[i] is the frequency assigned to the
   ith node
   node_i (0 ≤ i ≤ n − 1) of S //
   for i = 1 to n − 1 do
   begin
      Set t[i] to smallest integer without
      violating the frequency separation
      requirements specified by the matrix
      C with all the previously assigned
      values t[0], t[1], ..., t[i − 1].
   end
return max(t, t[1], ..., t[n − 1])
end .
```

Next, we describe the algorithm to resolve the channel assignment problem.

B. Algorithm GA

Step 1: Set the iteration number $t \leftarrow 0$; set $cp \leftarrow 0.95$; set $M \leftarrow 20$.

Step 2 (Initial population):

```
begin
  for i = 0 to M − 1 do
  begin
    Generate a random order of the nodes
    in the CAP graph and consider it as a
    string S_i.
  end.
set q_t ← {S_0, S_1, ..., S_{M−1}} as the initial
population.
end
```

Example: For the CAP graph shown in Fig. 1(a), with $M = 4$, $\{S_0, S_1, S_2, S_3\}$ is an initial population where $S_0 = S_1 = ((21),(20),(11),(00),(10))$, $S_2 = ((00),(20),(11),(21),(10))$, and $S_3 = ((11),(20),(21),(00),(10))$.

Note that the initial population may contain multiple copies of one or more strings. In particular, M copies of a single string may also constitute an initial population.

Step 3: Compute $\text{Fit}(S_i)$ for each string S_i $(0 \le i \le M-1)$ of q_t. Find the best string S_{best1} (i.e., the string with the least fitness value) and the worst string S_{worst1} (i.e., the string with the highest fitness value) of q_t. If S_{best1} or S_{worst1} is not unique, choose one arbitrarily.

Step 4: (Selection or Reproduction): Apply the selection operation as described below on the strings of q_t to generate a mating pool q_{mat} of size M.

Selection Operation

- Calculate the probability p_i of selection of S_i $(i = 0,\ 1,\ \ldots,\ M-1)$

$$p_i = \frac{\frac{1}{\text{Fit}(S_i)}}{\sum_{i=0}^{M-1} \frac{1}{\text{Fit}(S_i)}}.$$

  Note that with this choice of p_is, the probability of selection of S_i will be higher if $\text{Fit}(S_i)$ is smaller. This is required because we are dealing with the minimization problem.
- Calculate the cumulative probability q_i for S_i $(i = 0, 1, \ldots, M-1)$

$$q_i = \sum_{j=0}^{i} p_j.$$

- Generate a random number r_j from [0, 1] for $j = 0,\ 1,\ \ldots,\ M-1$. Now, if $r_j \le q_0$, select S_0; otherwise, select S_i $(1 \le i \le M-1)$ if $q_{i-1} < r_j \le q_i$.

Note that $p_0 = q_0$ and $p_i = q_i - q_{i-1}$ for $1 \le i \le M-1$.

Remark: With this implementation, if p_i is higher, i.e., the interval $(q_{i-1}, q_i]$ is of higher length, the probability that the random number generated falls within the interval $(q_{i-1}, q_i]$ will be higher. In other words, the probability of selection of S_i will be higher if $\text{Fit}(S_i)$ is smaller. In this way, from a current population of size M, a mating pool q_{mat} of the same size with potential strings is constructed. Since we are dealing with a minimization problem, we will say the string S_i is more potential than the string S_j if $\text{Fit}(S_i) < \text{Fit}(S_j)$. The strings of the mating pool then undergo crossover and mutation. This strategy for construction of a mating pool is close to that used in [3].

Step 5: (Crossover): Perform a crossover operation as described below on the strings of q_{mat} to obtain a population q_{temp1} of size M.

```
Crossover operation
Form M/2pairs by pairing the ith and
(M/2+i)th string from q_mat(i = 0, 1, ..., (M/2–1)).
for each pair of strings do
begin
  generate a random number R from [0, 1].
    if ( R ≤ cp) do
    //to produce two offsprings for the
    next generation.//
    begin
      generate two random numbers
      from [0, n − 1] to define a matching
      section. Use this matching
      section to effect a cross through
      position-by-position exchange
      operation.
    end.
end.
```

Note that the above reordering operator, known as partially matched crossover (PMX) [16], has the effect of both inversion and crossover.

Example: Let us assume that the two random numbers generated are 2 and 3. Then, PMX on $S_1 = ((00),\ (10),\ (11),\ (20),\ (21))$ and $S_2 = ((20),\ (00),\ (21),\ (10),\ (11))$ will produce the two offsprings $S_1' = ((00),\ (20),\ (21),\ (10),\ (11))$ and $S_2' = ((10),\ (00),\ (11),\ (20),\ (21))$.

Remark: We have used two-point crossover operation and PMX operator. Multipoint crossover and other forms of reordering operator such as order crossover (OX) and cycle crossover (CX) [16] can also be used.

Step 6: (Mutation): Perform a mutation operation as described below on the strings of q_{temp1} to obtain a population q_{temp2} of size M.

```
Mutation operation
Set q ← m_probability(t)
for each string S_i of q_temp1 (0 ≤ i ≤ M − 1) do
begin
  for each node node_j (0 ≤ j ≤ n − 1) of
  string S_i do
  begin
    generate a random number from [0,1],
    say m.
```

```
    if (m ≤ q) do
    begin
      exchange node_j of S_i with any other
      randomly selected node node_k of
      S_i (0 ≤ k ≤ n − 1,  k ≠ j)
    end
  end
end.
```

Example: If node (00) is mutated by node (21), after mutation, string $S_1 = ((00), (10), (11), (20), (21))$ be changed to string $S_1' = ((21), (10), (11), (20), (00))$.

Step 7: Calculate Fit(S_i) for each string S_i, $0 \leq i \leq M-1$ of q_{temp2}. Find the best string S_{best2} and the worst string S_{worst2} of q_{temp2}. If S_{best2} or S_{worst2} is not unique, choose one arbitrarily.

Step 8: (Elitism): Compare S_{best1} of q_t and S_{best2} of q_{temp2}. If Fit$(S_{\text{best2}}) >$ Fit(S_{best1}), then replace S_{worst2} with S_{best1}. Rename q_{temp2} as q_t.

Step 9: $t \leftarrow t+1$; Go to Step 3) if $t < T$, else terminate.

Bhandari *et al.* [3] provided the proof that an EGA converges to the global optimal solution with any choice of initial population as the number of iterations goes to infinity. They also proved that no finite stopping time can guarantee the optimal solution. But in practice, we must terminate the process after finitely many iterations with a high probability that the process has achieved the global optimal solution. In our problem, the optimal string is not necessarily unique. There may be many strings that provide the optimal value. It was shown in [3] that if the number of strings having the optimal value is larger, the probability of fast convergence is higher. Practically, GA either runs for a fixed number of iterations or terminates if no improvement is found at some stage within the fixed number of iterations.

V. GA on Special Cases of CAP

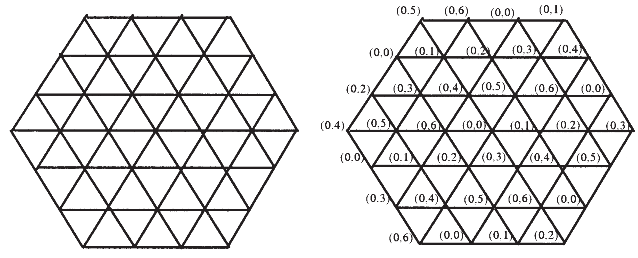
The GA presented in the previous section is applicable to any arbitrary CAP graph. We now consider the application of this algorithm to some special cases of the channel assignment problem. We assume that the cellular graph is of hexagonal structure and the channel interference extends only up to two cells from the call originating cell. First, we consider the case where every cell of the network has a demand of only one channel. In fact, we apply GA on a small subset of nodes to find its assignment and next extend the assignment regularly to cover the whole network, for faster convergence of GA.

Here, we propose three frequency-assignment schemes depending on the relative values of s_1 and s_2.

A. Scheme 1: For the Case of $s_1 = s_2$

Lemma 2: In a hexagonal cellular network of two-band buffering with a demand of only one channel per cell, there exists a frequency-assignment scheme that requires a bandwidth of $6s_2$ for $s_1 = s_2$.

Proof: Consider the subgraph shown in Fig. 4(a). In this scheme, the frequency separation constraint is specified by the matrix whose (i, j)th element is $s_1(= s_2)$ if cells i and j are within distance two, and zero otherwise. With this frequency separation matrix, we apply the GA over the subgraph shown in Fig. 4(a). The resulting assignment is shown in Fig. 4(b). Observe that the frequency channels (0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5), and (0, 6) are assigned repeatedly to the nodes over the subgraph in a regular and symmetrical way. The symmetry is clear from Fig. 4(b). Using this symmetry, frequencies can be assigned to the nodes over the entire network. This assignment is shown in Fig. 5 and is characterized as follows.



(a) (b)

Fig. 4. (a) A subgraph of a cellular graph and (b) frequency assignment to it when $s_1 = s_2$.

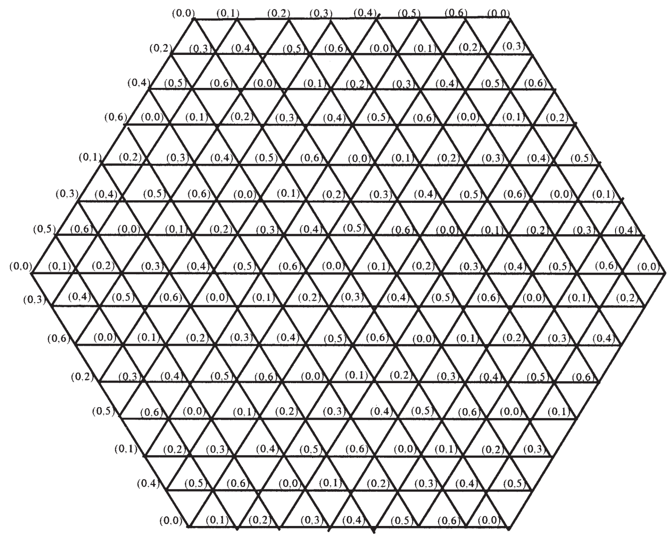


Fig. 5. Frequency assignment of a two-band buffering system for $s_1 = s_2$.

We first consider three directions x, y, and z in the cellular graph as shown in Fig. 6. In each direction we have definite types of lines, with each type of line being identified by the sequence of frequencies assigned to the nodes lying on that line.

In x direction, there is only one type of line (*type* x) identified by the repetitive sequence of period seven as: (0, 0) (0, 1) (0, 2) (0, 3) (0, 4) (0, 5) (0, 6) . . ..

In y direction, there is only one type of line (*type* y) identified by the repetitive sequence of period seven as : (0, 0) (0, 5) (0, 3) (0, 1) (0, 6) (0, 4) (0, 2) . . . .

In z direction, there is only one type of line (*type* z) identified by the repetitive sequence of period seven as: (0, 0) (0, 4) (0, 1) (0, 5) (0, 2) (0, 6) (0, 3) . . . .

It follows from Fig. 5 that along any of the lines of *type* x, *type* y, and *type* z, there is no conflict. Therefore, this assignment is conflict-free over the entire network.

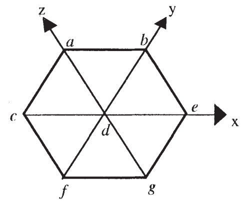


Fig. 6. Three directions in the hexagonal cellular graph.

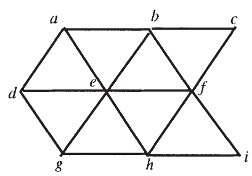


Fig. 7. A nine-node subgraph of hexagonal cellular network.

Hence, by using the scheme described above, the frequencies can be assigned to the nodes of the entire network using a bandwidth of $6s_2$, when $s_1 = s_2$. □

We now generalize Lemma 2 and state the following theorem for the assignment of w channels to every node of the entire cellular network.

Theorem 3: In a hexagonal cellular network with homogeneous demand w and two-band buffering, there exists a frequency-assignment scheme for $s_1 = s_2$, which requires a bandwidth of:

Case 1) $(w-1)7s_2 + 6s_2$ when $s_0 \le 7s_2$;
Case 2) $(w-1)s_0 + 6s_2$ when $s_0 \ge 7s_2$.

Proof: Follows from Lemma 2. □

Remark: For the second case of Theorem 3, i.e., when $s_1 = s_2$ and $s_0 \ge 7s_2$, the required bandwidth is close to $(w-1)s_0$.

Remark: For $s_1 = s_2$, the above assignment requires a bandwidth equal to the corresponding lower bound presented in Theorem 1 for different relative values of s_0, s_1, and s_2. Hence this assignment is optimal.

B. Scheme 2: For the Cases of i) $s_2 < s_1 \le 2s_2$ and ii) $s_1 \ge 2s_2$

Consider the nine-node subgraph with nodes $a, b, c, d, e, f, g, h,$ and i of the cellular graph as shown in Fig. 7.

Each node of this subgraph represents a cell of the cellular network. Suppose that $\phi(\alpha)$ is the frequency assigned to node α, $\alpha \in \{a, b, c, d, e, f, g, h, i\}$. We propose to use only these nine frequencies to be assigned to the nodes of the entire cellular graph in a regular and symmetrical fashion as shown in Fig. 8. The idea behind this assignment is explained as follows.

We again consider the three directions x, y, and z in the cellular graph as shown in Fig. 6.

In x direction, there are three different types of lines identified by the repetitive sequence of period three as:

type x_1: $\phi(a)\ \phi(b)\ \phi(c)\ \phi(a)\ \phi(b)\ \phi(c)\ \ldots$;
type x_2: $\phi(d)\ \phi(e)\ \phi(f)\ \phi(d)\ \phi(e)\ \phi(f)\ \ldots$;
type x_3: $\phi(g)\ \phi(h)\ \phi(i)\ \phi(g)\ \phi(h)\ \phi(i)\ \ldots$.

In y direction, there is only one type of line (*type* y) identified by the repetitive sequence of period nine as: $\phi(a)\ \phi(h)\ \phi(f)\ \phi(c)\ \phi(g)\ \phi(e)\ \phi(b)\ \phi(i)\ \phi(d)\ \ldots$.

TABLE I
FREQUENCY SEPARATION MATRIX FOR SCHEME 2

$Nodes \rightarrow$ $\downarrow$	a	b	c	d	e	f	g	h	i
a	0	s_1	s_1	s_1	s_1	s_2	s_1	s_1	s_2
b	s_1	0	s_1	s_2	s_1	s_1	s_2	s_1	s_1
c	s_1	s_1	0	s_1	s_2	s_1	s_1	s_2	s_1
d	s_1	s_2	s_1	0	s_1	s_1	s_1	s_2	s_1
e	s_1	s_1	s_2	s_1	0	s_1	s_1	s_1	s_2
f	s_2	s_1	s_1	s_1	s_1	0	s_2	s_1	s_1
g	s_1	s_2	s_1	s_1	s_1	s_2	0	s_1	s_1
h	s_1	s_1	s_2	s_2	s_1	s_1	s_1	0	s_1
i	s_2	s_1	s_1	s_1	s_2	s_1	s_1	s_1	0

In z direction, there is only one type of line (*type* z) identified by the repetitive sequence of period nine as: $\phi(c)\ \phi(i)\ \phi(f)\ \phi(b)\ \phi(h)\ \phi(e)\ \phi(a)\ \phi(g)\ \phi(d)\ \ldots$.

Based on this assignment scheme, we construct a table of frequency separation constraints between every pair of these nine nodes, as shown in Table I (assuming that only one channel is assigned to each node), such that

1) along each of the lines of *type* x_1, *type* x_2, and *type* x_3 in x direction, there is no conflict;
2) along all the lines of *type* y in y direction, there is no conflict;
3) along all the lines of *type* z in z direction, there is no conflict.

Therefore, these nine frequencies can be repeated over the entire network without any conflict.

It is important to note that Table I has been constructed so that only nine frequency channels assigned to the nodes of Fig. 7 would be sufficient for the frequency assignment to nodes of the entire cellular network following the assignment patterns as shown in Fig. 8, without any conflict.

We now use the GA over the nine-node subgraph of Fig. 7 with the frequency separation requirements given by Table I. Depending on the relationship between s_1 and s_2, the results of the algorithm can be stated by the following two lemmas.

Lemma 3: In a hexagonal cellular network of two-band buffering with a demand of only one channel per cell, there exists a frequency assignment scheme that requires a bandwidth of $8s_2$ for $s_2 < s_1 \le 2s_2$.

Proof: For the case when $s_2 < s_1 \le 2s_2$, GA gives the result such that $\phi(a) = (0,0)$, $\phi(b) = (0,6)$, $\phi(c) = (0,3)$, $\phi(d) = (0,5)$, $\phi(e) = (0,2)$, $\phi(f) = (0,8)$, $\phi(g) = (0,7)$, $\phi(h) = (0,4)$, and $\phi(i) = (0,1)$. Note that the inequality $2s_2 \ge s_1$ has been used here, because the frequencies (0, 6) and (0, 8) are assigned to nodes distance one apart from each other. In this case, the maximum frequency assigned is (0, 8). This assignment is shown in Fig. 9, which is obtained by replacing the values of $\phi(a)$, $\phi(b)$, $\phi(c)$, $\phi(d)$, $\phi(e)$, $\phi(f)$, $\phi(g)$, $\phi(h)$, and $\phi(i)$ in Fig. 8by (0, 0), (0, 6), (0, 3), (0, 5), (0, 2), (0, 8), (0, 7), (0, 4), and (0, 1) respectively.

Hence, by using the frequency-assignment scheme described above, the frequencies can be assigned to the nodes of the entire network using a bandwidth of $8s_2$ when $2s_2 \ge s_1 > s_2$. □

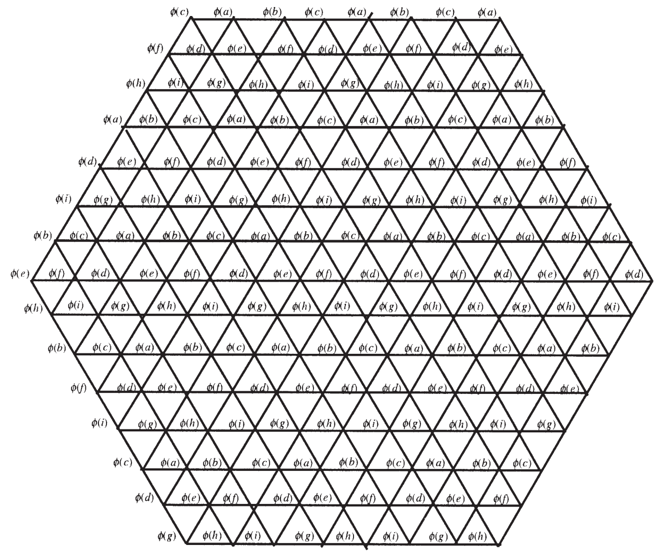


Fig. 8. Frequency assignment of a two-band buffering system using nine frequency channels.

Lemma 4: In a hexagonal cellular network of two-band buffering with a demand of only one channel per cell, there exists a frequency assignment scheme that requires a bandwidth of $4s_1$ for $s_1 \geq 2s_2$.

Proof: Follows exactly in the same way as in Lemma 3. This assignment is shown in Fig. 10. □

Next we consider the case of homogeneous demand with w channels. The following two theorems simply extend the results of Lemmas 3 and 4, respectively.

Theorem 4: In a hexagonal cellular network with homogeneous demand w and two-band buffering, there exists a frequency-assignment scheme for $s_2 < s_1 \leq 2s_2$, which requires a bandwidth of:

Case 1) $(w-1)9s_2 + 8s_2$ when $s_0 \leq 9s_2$;
Case 2) $(w-1)s_0 + 8s_2$ when $s_0 \geq 9s_2$.

Proof: Follows from Lemma 3. For $w = 2$, the assignments are illustrated in Fig. 11(a) and (b) for the cases $s_0 \leq 9s_2$ and $s_0 \geq 9s_2$, respectively. □

Remark: For the second case of Theorem 4, i.e., when $s_2 < s_1 \leq 2s_2$ and $s_0 \geq 9s_2$, the required bandwidth is close to $(w-1)s_0$.

Theorem 5: In a hexagonal cellular structure with homogeneous demand w and two-band buffering, there exists a frequency-assignment scheme for $s_1 \geq 2s_2$, which requires a bandwidth of:

Case 1) $(w-1)(4s_1 + s_2) + 4s_1$ when $s_0 \leq (4s_1 + s_2)$;
Case 2) $(w-1)s_0 + 4s_1$ when $s_0 \geq (4s_1 + s_2)$.

Proof: Follows from Lemma 4. For $w = 2$, the corresponding assignments are shown in Fig. 12(a) and (b) for the cases $s_0 \leq (4s_1 + s_2)$ and $s_0 \geq (4s_1 + s_2)$, respectively. □

Remark: For the second case of Theorem 5, i.e., when $s_1 \geq 2s_2$ and $s_0 \geq (4s_1 + s_2)$, the required bandwidth is close to $w(-1)s_0$.

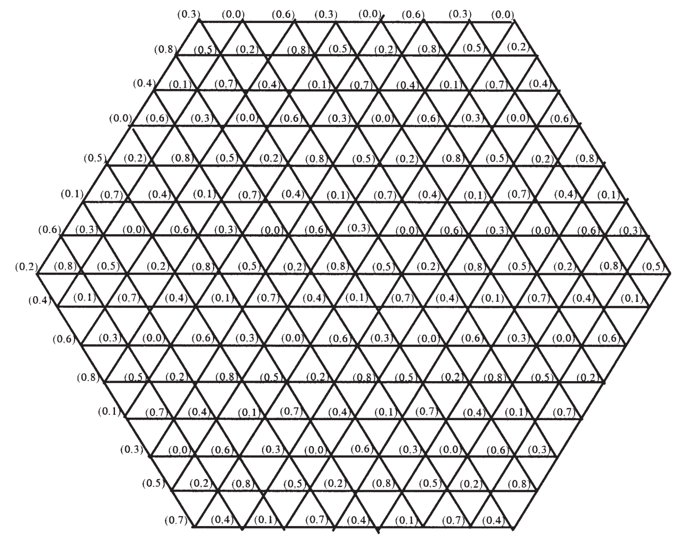


Fig. 9. Frequency assignment of a two-band buffering system for $s_2 < s_1 \leq 2s_2$.

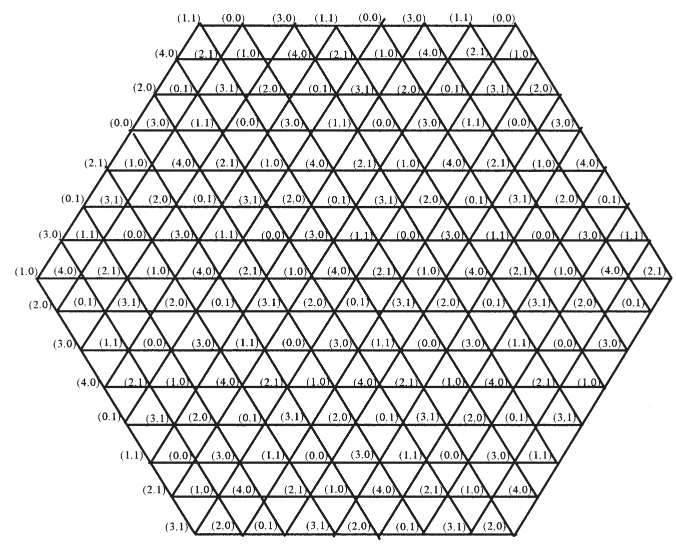
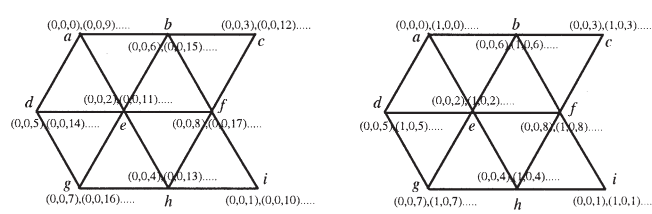
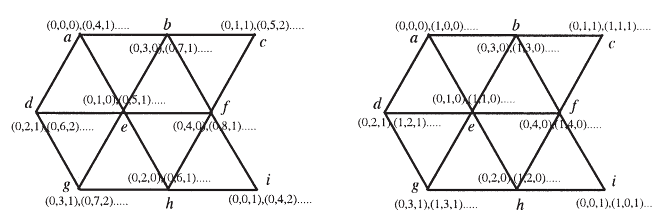


Fig. 10. Frequency assignment of a two-band buffering system for $s_1 \geq 2s_2$.



(a) (b)

Fig. 11. Frequency assignments to nine-node subgraph for $s_2 < s_1 \leq 2s_2$.



(a) (b)

Fig. 12. Frequency assignments to nine-node subgraph for $s_1 \geq 2s_2$.

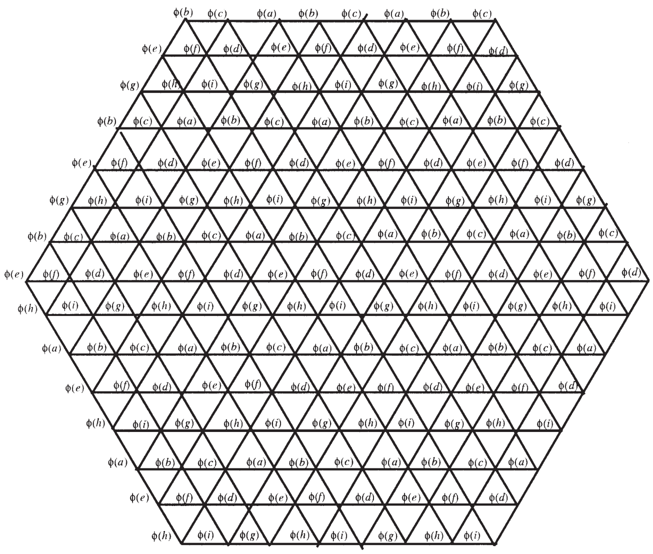


Fig. 13. Frequency assignment of a two-band buffering system using nine frequency channels.

C. Scheme 3 : For the Case of $s_1 \geq s_2$

Consider again the nine-node subgraph of Fig. 7. In this scheme as well, we propose to use only nine frequencies assigned to these nine nodes in a regular and symmetric fashion for the whole cellular graph as shown in Fig. 13. However, we would use here a different frequency separation matrix as given in Table II. The difference between Tables I and II is in the submatrix corresponding to the rows for a, b, c and columns for g, h, and i . Because of this change, the repetition pattern to be used here will be slightly different from that of scheme 2, as discussed below.

We again consider the three directions x, y, and z in the cellular graph as shown in Fig. 6.

In x direction, there are three different types of lines (*types* x_1, x_2, and x_3) as in scheme 2 above.

In y direction, there are three different types of lines identified by the repetitive sequence of period three as:

type y_1: $\phi(b)\ \phi(g)\ \phi(e)\ \phi(b)\ \phi(g)\ \phi(e)\ \ldots$;
type y_2: $\phi(a)\ \phi(i)\ \phi(d)\ \phi(a)\ \phi(i)\ \phi(d)\ \ldots$;
type y_3: $\phi(c)\ \phi(h)\ \phi(f)\ \phi(c)\ \phi(h)\ \phi(f)\ \ldots$.

In z direction, there are also three different types of lines identified by the repetitive sequence of period three as:

type z_1: $\phi(g)\ \phi(d)\ \phi(c)\ \phi(g)\ \phi(d)\ \phi(c)\ \ldots$;
type z_2: $\phi(b)\ \phi(i)\ \phi(f)\ \phi(b)\ \phi(i)\ \phi(f)\ \ldots$;
type z_3: $\phi(h)\ \phi(e)\ \phi(a)\ \phi(h)\ \phi(e)\ \phi(a)\ \ldots$.

It follows from Fig. 13 and the frequency separation matrix given by Table II that

1) along each of the lines of *type* x_1, *type* x_2, and *type* x_3 in x direction, there is no conflict;
2) along each of the lines of *type* y_1, *type* y_2, and *type* y_3 in y direction, there is no conflict;
3) along each of the lines of *type* z_1, *type* z_2, and *type* z_3 in z direction, there is no conflict.

Therefore, these frequencies can be repeated over the entire network without any conflict.

TABLE II
FREQUENCY SEPARATION MATRIX FOR SCHEME 2

$Nodes \rightarrow$ $\downarrow$	a	b	c	d	e	f	g	h	i
a	0	s_1	s_1	s_1	s_1	s_2	s_2	s_1	s_1
b	s_1	0	s_1	s_2	s_1	s_1	s_1	s_2	s_1
c	s_1	s_1	0	s_1	s_2	s_1	s_1	s_1	s_2
d	s_1	s_2	s_1	0	s_1	s_1	s_1	s_2	s_1
e	s_1	s_1	s_2	s_1	0	s_1	s_1	s_1	s_2
f	s_2	s_1	s_1	s_1	s_1	0	s_2	s_1	s_1
g	s_2	s_1	s_1	s_1	s_1	s_2	0	s_1	s_1
h	s_1	s_2	s_1	s_2	s_1	s_1	s_1	0	s_1
i	s_1	s_1	s_2	s_1	s_2	s_1	s_1	s_1	0

It is important to note that Table II has been constructed so that only nine frequency channels assigned to the nodes of Fig. 7 would be sufficient for the frequency assignment to nodes of the entire cellular network following the assignment patterns as shown in Fig. 13 without any conflict.

We now use the GA over the nine-node subgraph of Fig. 7 with the frequency separation requirements given by Table II. The results of the algorithm can be stated by the following lemma.

Lemma 5: In a hexagonal cellular network of two-band buffering with a demand of only one channel per cell, there exists a frequency-assignment scheme that requires a bandwidth of $(2s_1 + 6s_2)$ for all values of s_1 and s_2 where $s_1 \geq s_2$.

Proof: For this case, the GA gives a result such that $\phi(a) = (0, 0)$, $\phi(b) = (2, 4)$, $\phi(c) = (1, 3)$, $\phi(d) = (2, 5)$, $\phi(e) = (1, 4)$, $\phi(f) = (0, 2)$, $\phi(g) = (0, 1)$, $\phi(h) = (2, 6)$, and $\phi(i) = (1, 2)$. Note that the maximum frequency assigned is (2, 6). This assignment is shown in Fig. 14, which is obtained by replacing the values of $\phi(a)$, $\phi(b)$, $\phi(c)$, $\phi(d)$, $\phi(e)$, $\phi(f)$, $\phi(g)$, $\phi(h)$, and $\phi(i)$ in Fig. 13, by (0, 0), (2, 4), (1, 3), (2, 5), (1, 4), (0, 2), (0, 1), (2, 6), and (1, 2), respectively.

Hence, by using the scheme described above, assignment to the nodes of the entire network can be done using a bandwidth of $(2s_1 + 6s_2)$ for all cases of $s_1 \geq s_2$. □

We now state the following result for assigning w frequency channels to each of the nodes of the entire cellular network by simply extending the results of Lemma 5.

Theorem 6: In a hexagonal cellular network of two-band buffering with a homogeneous demand w, there exists a frequency assignment scheme for all values of s_1 and s_2 where $s_1 \geq s_2$, which requires a bandwidth of:

Case 1) $(w-1)(3s_1 + 6s_2) + (2s_1 + 6s_2)$, for $s_0 \leq (3s_1 + 6s_2)$;

Case 2) $(w-1)s_0 + (2s_1 + 6s_2)$, for $s_0 \geq (3s_1 + 6s_2)$.

Proof: Follows from Lemma 5. For $w = 2$, the corresponding assignments are illustrated in Fig. 15(a) and (b) for the cases $s_0 \leq (3s_1 + 6s_2)$ and $s_0 \geq (3s_1 + 6s_2)$, respectively. □

Remark: For the second case of Theorem 6, i.e., when $s_0 \geq (3s_1 + 6s_2)$, the required bandwidth is close to $(w-1)s_0$.

Remark: The results of Lemmas 3–5 are based on assigning frequencies to a nine-node subgraph of the entire network. This subgraph with the corresponding Tables I and II has been chosen

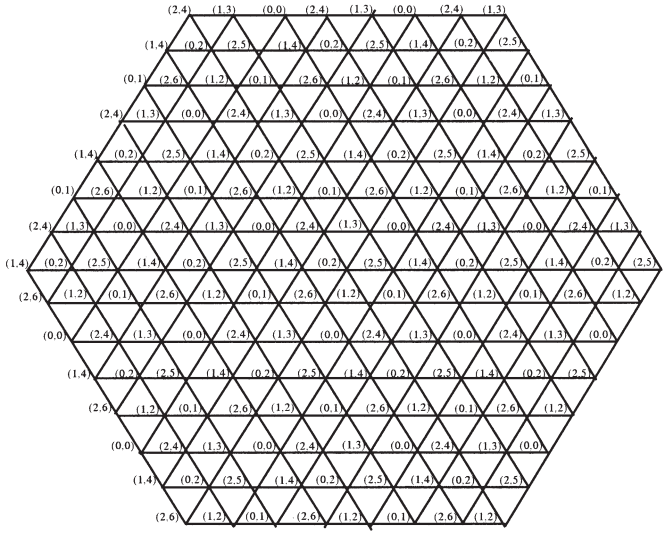


Fig. 14. Frequency assignment of a two-band buffering system for $s_1 \geq 3s_2$.

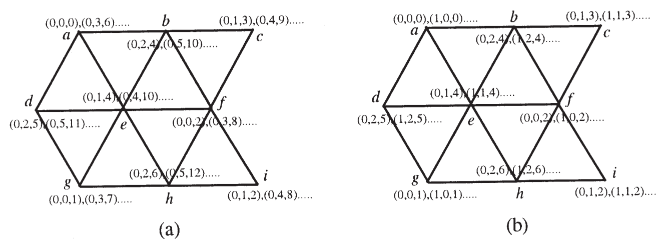


Fig. 15. Frequency assignments to nine-node subgraph for $s_1 \geq s_2$.

so that only nine frequency channels would be sufficient for frequency assignment to the nodes of the entire cellular network. Different choices of the subgraph or the frequency separation matrix would give different assignment patterns. Besides that, we obtained the results of Lemmas 3–5 by applying GA. Therefore, it is likely that we would not always get the best solution. Since the results of Theorems 4–6 are based on these lemmas, we cannot claim that Theorems 4–6 give minimal bandwidth for assignment. However, for different relative values of s_1 and s_2, the bandwidth required by our assignment schemes is either less than or equal to that reported in [1].

D. Overall Bandwidth Requirement

A frequency-assignment scheme was described in [1] for a hexagonal cellular network with a demand of only one channel per cell and two-band buffering, requiring a bandwidth of $(2s_1 + 6s_2)$ for all cases of $s_1 \geq s_2$. Let I be the factor of improvement in bandwidth achieved by our proposed schemes over that in [1]. Summarizing all the results of Lemmas 2–5, we now consider the following cases.

Case 1) When $s_1 = s_2$.

$$I = \frac{2s_1 + 6s_2}{6s_2} = \frac{4}{3}$$

leading to an improvement by 25%.

Case 2) When $s_2 < s_1 \leq 2s_2$.

$$I = \frac{2s_1 + 6s_2}{8s_2}.$$

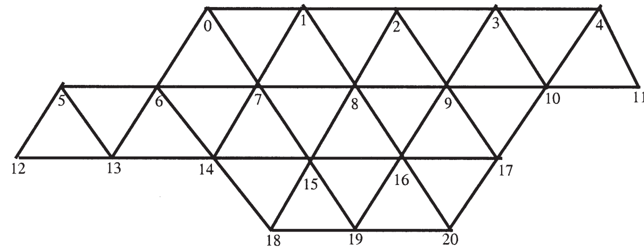


Fig. 16. The benchmark cellular network.

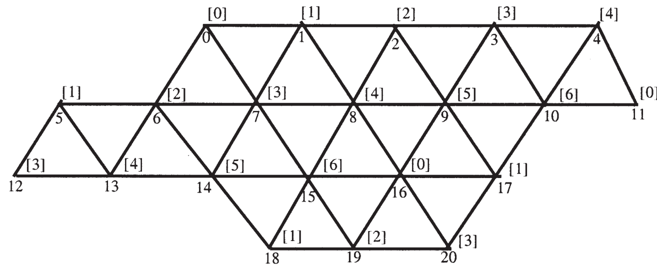


Fig. 17. Initial assignment of problem 7′.

Note that I lies within the interval (1, 5/4], leading to an improvement of 20% at best.

Case 3) When $2s_2 \leq s_1 \leq 3s_2$

$$I = \frac{2s_1 + 6s_2}{4s_1}.$$

Here I lies within the interval [1, 5/4], leading to an improvement of 20% at best.

Case 4) When $s_1 \geq 3s_2$.

$$I = \frac{2s_1 + 6s_2}{2s_1 + 6s_2} = 1.$$

Considering all the cases, we conclude that our proposed schemes result in at most a 25% improvement in bandwidth at best over that in [1].

For homogeneous demands of w channels per cell, we summarize our results below (Theorems 3–6) on bandwidth requirement for different relative values of s_0, s_1, and s_2.

1) When $s_1 = s_2$ (Theorem 3):
   a) $(w-1)7s_2 + 6s_2$ for $s_0 \leq 7s_2$ (Case 1 of Theorem 3);
   b) $(w-1)s_0 + 6s_2$ for $s_0 \geq 7s_2$ (Case 2 of Theorem 3).
2) When $s_2 < s_1 \leq 2s_2$ (Theorem 4):
   a) $(w-1)9s_2 + 8s_2$ for $s_0 \leq 9s_2$ (Case 1 of Theorem 4);
   b) $(w-1)s_0 + 8s_2$ for $s_0 \geq 9s_2$ (Case 2 of Theorem 4).
3) When $s_1 \geq 2s_2$, Theorem 5 or 6 or whichever gives the minimum bandwidth:
   a) $(w-1)(4s_1 + s_2) + 4s_1$ for $s_0 \leq (4s_1 + s_2)$ (Case 1 of Theorem 5);
   b) $(w-1)s_0 + 4s_1$ for $s_0 \geq (4s_1 + s_2)$ (Case 2 of Theorem 5);

TABLE III
TWO DIFFERENT DEMAND VECTORS FOR BENCHMARK PROBLEMS

Cell nos.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
D_1	8	25	8	8	8	15	18	52	77	28	13	15	31	15	36	57	28	8	10	13	8
D_2	5	5	5	8	12	25	30	25	30	40	40	45	20	30	25	15	15	30	20	20	25

c) $(w-1)(3s_1+6s_2)+(2s_1+6s_2)$ for $s_0 \leq (3s_1+6s_2)$ (Case 1 of Theorem 6);

d) $(w-1)s_0+(2s_1+6s_2)$ for $s_0 \geq (3s_1+6s_2)$ (Case 2 of Theorem 6).

Example 2: Let $s_0 = 8$, $s_1 = 1$, and $s_2 = 1$. In this case, the bandwidth requirement will be given by Case 2 of Theorem 3, which corresponds to item 1b) above.

VI. CHANNEL ASSIGNMENT FOR NONHOMOGENEOUS DEMANDS

So far, we have considered a cellular network with homogeneous demand only. However, we can extend our proposed schemes to handle cases with nonhomogeneous demands as well. Given a network with nonhomogeneous demand vector $W = (w_i)$, with w_i being the demand of cell i, it is evident that a rough estimate of the minimum bandwidth requirement is given by $w(-1)s_0$, where $w = \max(w_i)$. We also observe that when the relative values of s_0, s_1, and s_2 are such that they satisfy the condition given by case 2 of any of Theorems 3–6, the required bandwidth following our assignment schemes is very close to $(w-1)s_0$ (see remarks of Theorems 3–6). Hence, at least for these situations, we can conclude that using our assignment schemes for homogeneous demand of w channels per cell, as discussed in the previous section, we may obtain a solution to the problem with nonhomogeneous demand vector $W = (w_i)$ where $w = \max(w_i)$, keeping the bandwidth requirement very close to the optimal one.

To establish this finding, we consider eight well-known benchmark problems widely used in the literature [9], [10], [18]–[20], [22]–[27], [22].

A. Benchmark Instances

These benchmark problems have been defined on a hexagonal cellular network of 21 cells as shown in Fig. 16, with either of two nonhomogeneous demand vectors D_1 and D_2, as shown in Table III. Column-i of Table III indicates the channel demand from cell i corresponding to D_1 or D_2. Table IV shows the specifications of these eight problems (problems 1–8) in terms of the specific values of s_0, s_1, and s_2 for the two-band buffering system and the corresponding demand vector used for each of them.

First let us consider benchmark problems 3 and 7 described above, where the nonhomogeneous demand vectors $W = (w_i)$ are given by D_1 and D_2, respectively, and the frequency separation constraints for both of the problems are specified as $s_0 = 7$, $s_1 = 1$, and $s_2 = 1$. If we start with frequency 0, the minimum frequencies by which the assignments of problem 3 and 7 can be completed are at least $76 \times 7 = 532$ and $44 \times 7 = 308$, respectively, because 77 and 45 are the highest demands in D_1 and D_2, respectively, and $s_0 = 7$ for both problems.

TABLE IV
THE SPECIFICATION OF BENCHMARK PROBLEMS

Problems		1	2	3	4	5	6	7	8
Frequency	s_0	5	5	7	7	5	5	7	7
separation	s_1	1	2	1	2	1	2	1	2
constraints	s_2	1	1	1	1	1	1	1	1
Demand vector		D_1	D_1	D_1	D_1	D_2	D_2	D_2	D_2

Let us now derive a modified problem $7'$ from problem 7, such that all other specifications remain the same and only the demand vector D_2' for $7'$ is homogeneous with demand 45, i.e., the maximum demand of problem 7. According to the relative values of s_0, s_1, and s_2, we are to apply the result of Case 2 of Theorem 3. It shows that the assignment of problem $7'$ can be completed with bandwidth $44\times7+6\times1 = 314$. For problem $7'$, using Lemma 2, the initial assignment chosen for allocating one channel per node is shown in Fig. 17. The label $[\alpha]$ associated with each node of Fig. 17 indicates that the frequency channel α is assigned to that node.

Subsequent channels can be assigned to each node following the scheme corresponding to Theorem 3 in the previous section, leading to the frequencies assigned to various cells as shown in Fig. 18. The label $[\alpha-\beta]$ associated with each node of Fig. 18 indicates that the frequency channels $\alpha, \alpha+7, \ldots, \beta-7, \beta$ are assigned to that node. There are 45 frequency entries per node in Fig. 18, and in effect, we have also solved problem 7 by this process. For the assignment corresponding to problem 7, we need to select only the first w_i entries from the ith node $(0 \leq i \leq 20)$ of Fig. 18 , where $w_i(\leq 45)$ is the channel requirement for cell i given by D_2.

It can be checked from Fig. 18 that the benchmark problem 7 can actually be solved using frequencies from 0 to 308 only. Essentially, the idea was to initially assign 0 frequency to node 11 (with the maximum demand of 45) for the first channel, as shown in Fig. 17, so that we end up with the maximum frequency of 308 at this node. In other words, we have solved benchmark problem 7 with an optimal assignment.

Note that, to get the assignments of Fig. 17, we need to run GA on a nine-node subgraph (as shown in Fig. 7) with single demand per node. The assignments of Fig. 18 are obtained using the results of Theorem 3. As a result, the computation time needed to solve problem 7 is a fraction of a second on a DEC Alpha station. Likewise, it can be verified that benchmark problem 3 can also be optimally solved using frequencies only from 0 to 532.

For the other six benchmark problems, if we convert the nonhomogeneous assignment problem to its corresponding homogeneous counterpart with $w = \max(w_i)$ demand per node, the required minimum bandwidth as given by Theorems 3–6 will be well above ws_0. For each of these problems, we have applied

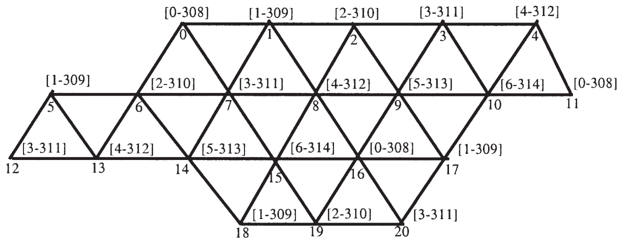


Fig. 18. Complete assignment of problems 7′ and 7.

our proposed algorithm (Algorithm GA) to get the bandwidths as shown in Table V.

B. Comparison of Results

In order to evaluate the performance of the algorithms for *static channel assignment problem,* the quality of derived solutions is generally considered to be much more important than the computation time [18], [29].

Regarding the optimality of the solutions, results from earlier works along with that from our proposed approach are shown in Table V. The row *Lower Bound* in Table V corresponds to the lower bound for each of the problems as reported in [18].

As we see from this table, our proposed scheme can solve each of the eight benchmark problems optimally. Most of the other algorithms (except that presented in [18]) determined an optimal assignment only for six of the benchmark problems, excluding problems 2 and 6.

In fact, problems 2 and 6 are regarded as the most difficult in the literature [18], [27]. For example, the assignment algorithm given in [19] required 165 h of computing time for problem 6 on an unloaded HP Apollo 9000/700 workstation, but giving only a nonoptimal solution with 268 channels (optimality requires only 253 channels). Later, however, the authors in [18] proposed an algorithm that provided an optimal solution for problem 6 with a running time of 10 min on the same workstation. Among the later works, the frequency exhaustive strategy with rearrangement algorithm in [20] and the randomized saturation degree (RSD) heuristic presented in [27] also produce only nonoptimal solutions to benchmark problems 2 and 6. However, combining their RSD heuristic with a local search algorithm, the authors in [27] were able to find an optimal solution for problem 2 but not for problem 6. Most recently, an efficient heuristic algorithm has been proposed in [29], which also produced nonoptimal results for problems 2 and 6 with 463 and 273 channels, respectively.

The exact computation times have not been mentioned in most of the earlier works for benchmark problems 1, 3, 4, 5, 7, and 8. Some papers only mentioned the order of computation time, e.g., a few seconds, etc. Also, the workstations on which the algorithms run are different. However, to have an idea about the respective computation times, we have included Table VI, showing the times of our approach and some recent works that mentioned the computing times as well as produced optimal results.

For fair comparison, we may normalize the speeds of different machines mentioned in Table VI by using the SPECint95 and SPECfp95 parameters.[1] For example, the SPECint95 and SPECfp95 parameters of the DEC Alpha station 200 4/233 used by us are given by 3.39 and 4.32, respectively, whereas those for HP 9000 Series 700 Model 712/100 are 3.76 and 4.03, respectively. It shows that these two systems are of comparable performance.

[1]See http://www.specbench.org/osg/cpu95/results/cint95(cfp95).html.

TABLE V
PERFORMANCE COMPARISONS

Problems	1	2	3	4	5	6	7	8
Lower Bound	381	427	533	533	221	253	309	309
Our approach	381	427	533	533	221	253	309	309
(2001)[29]	381	463	533	533	221	273	309	309
(2001)[27]	381	427	533	533	221	254	309	309
(2000)[20]	381	433	533	533	-	260	-	309
(1999)[18]	381	427	533	533	221	253	309	309
(1998)[19]	-	-	-	-	221	268	-	309
(1997)[22]	381	-	533	533	221	-	309	309
(1997)[23]	381	436	533	533	-	268	-	309
(1996)[24]	381	-	533	533	-	-	-	-
(1996)[10]	381	433	533	533	221	263	309	309
(1994)[25]	381	464	533	536	-	293	-	310
(1992)[26]	381	-	533	533	221	-	309	309
(1989)[9]	381	447	533	533	-	270	-	310

Finally, it is to be noted that our technique finds optimal results for all benchmark instances. However, the computation times required by our technique for problems 3 and 7 are much better; and for problems 1, 4, 5, and 8, the computation times are comparable with those reported earlier. For problems 2 and 6, however, our computation time for the optimal assignment varied between 12–80 h for different runs on DEC Alpha station. Hence, our results are better with respect to either optimality and/or computation time for all benchmark instances except problems 2 and 6, for which optimal results have been reported in [18] with lesser computation time. It is to be noted that our proposed approach and that in [18] are both based on the applications of genetic algorithm. However, the approach in [18] starts by estimating the lower bound Z on bandwidth and attempts to satisfy all the constraints. If no solution is found, then it increments Z by one and repeats the procedure. Therefore, it is evident that the computation time will be highly dependent on the proximity of the estimated lower bound for the given problem to the optimal value of the bandwidth. But the problem of finding an achievable lower bound itself is an *NP-complete* one [26]. Our proposed technique, however, does not depend on any initial estimate of the lower bound and achieves optimal bandwidths for all benchmarks.

VII. CONCLUSION

In this paper, we have derived tighter lower bounds on the required bandwidth for assigning w channels to each node of a hexagonal cellular network with two-band buffering for different relative values of s_0, s_1, and s_2. Next, we presented an algorithm based on GA for solving the channel-assignment problem in its general form. We then showed how this technique can be effectively applied to the special cases of hexagonal cellular networks with homogeneous demands and two-band buffering. We have also shown that our methodology can easily be applied to cases of nonhomogeneous demands. The application of our algorithm to the eight well-known benchmark

TABLE VI
COMPUTATION TIME COMPARISONS BETWEEN THE EXISTING CAP ALGORITHMS AND OUR APPROACH

Problems	1	3	4	5	7	8	*Workstation*
Our approach	$2-5$ *sec*	$0.5-1.0$ *sec*	$6-12$ *sec*	$2-7$ *sec*	$0.5-1.0$ *sec*	$6-17$ *sec*	*DEC Alpha station*
(2001)[29]	7.5 *sec*	8.2 *sec*	11.1 *sec*	6.9 *sec*	6.0	10.2 *sec*	*DEC Alpha station*
(2001)[27]	<1 *sec*	<1 *sec*	<1 *sec*	<1 *sec*	<1 *sec*	<1 *sec*	*DECAlphaServer* 2100
(2000)[20]	–	–	–	–	–	–	–
(1999)[18]	*few sec*	*few sec*	*few sec*	*few sec*	*few sec*	*few sec*	*HP Apollo* 9000/7000
(1998)[19]	–	–	–	–	19365 *sec* ($\pm$16782)	89196 *sec* ($\pm$64846)	*HP Apollo* 9000/7000

problems, including the most difficult two, produces an optimal solution in each case. It performs better in terms of optimality of bandwidth and/or computation time, in general, compared to those presented in earlier works, except the computation time for the benchmark problems 2 and 6. However, we present a technique for improving the computation time for these two difficult problems following this approach in [30].

APPENDIX I
PROOF OF LEMMA 1

Since any two nodes of the subgraph shown in Fig. 2 are either at distance one or two from each other, the frequencies assigned on any two nodes of this subgraph must be separated by at least s_1 or s_2 depending on whether they are distance one or two apart, respectively. In Fig. 2, d is the central node and all the remaining nodes a, b, c, e, f, and g of this subgraph are distance one apart from d. We divide the six nodes a, b, c, e, f, and g into two sets A and B in such a way that any two nodes of a set are at distance two from each other. Let $A = \{a, e, f\}$ and $B = \{b, c, g\}$. The minimum frequency that can be assigned to any node of this subgraph is $(0, 0)$. We now have the following cases.

Case 1) Let frequency $(0, 0)$ be assigned to the central node d. Now, we assign frequencies to the nodes of set A (or B). After the assignment of all the nodes of set A (or B), we will assign frequencies to the nodes of set B (or A). The minimum frequency that can be assigned to any node of set A is $(1, 0)$ because any node of set A is distance one apart from d, which has already been assigned a frequency $(0, 0)$. Without loss of generality, let (1,0) be assigned to node a. Now, the minimum frequency that can be assigned to either node f or node e is (1, 1), because both nodes e and f are distance two apart from a [which has already been assigned a frequency (1, 0)] and distance one apart from d with assigned frequency (0, 0). Without loss of generality, let (1, 1) be assigned to f. Then, it follows from a similar argument that the minimum frequency that can be assigned to e is (1, 2). We can now start assigning frequencies to the nodes of set B. Note that e has been assigned the highest frequency among all the previously assigned nodes d, a, f, and e. Node c of set B is at distance two from node e, i.e., it requires less frequency separation from e. To satisfy the frequency separation requirements with all the previously assigned nodes d, a, f, and e, the minimum frequency that can be assigned to node c of set B is i) (1, 3) when $s_2 \le s_1 \le 2s_2$ and ii) (2, 1) when $s_1 \ge 2s_2$. By similar arguments, it follows that the minimum frequencies that can be assigned to nodes b and g are : i) (1, 4) and (1, 5), respectively, when $s_2 \le s_1 \le 2s_2$ and ii) (2, 2) and (2, 3), respectively, when $s_1 \ge 2s_2$. The corresponding assignments for $s_2 \le s_1 \le 2s_2$ and $s_1 \ge 2s_2$ are shown in Fig. 19(a) and (b), respectively.

Note that, in Fig. 19(a), the inequality $s_1 \le 2s_2$ has been used because the frequencies (1, 1) and (1, 3) are assigned to nodes f and c, respectively, which are at distance one from each other.

Similarly, in Fig. 19(b), the inequality $s_1 \ge 2s_2$ has also been used, because frequencies (1, 2) and (2, 1) are assigned to nodes e and c, respectively, which are at distance two from each other.

It shows clearly that $s_1 = 2s_2$ is the critical point to determine the minimum bandwidth requirement for the seven-node subgraph.

Case 2) Let channel (0, 0) be assigned to any node other than the central node. Without loss of generality, let (0, 0) be assigned to node e. As $e \in A$, our next attempt is to assign frequencies to a or f. The minimum frequency that can be assigned to a or f is (0, 1) because both of them are at distance two from e, which has already been assigned a frequency $(0, 0)$. Without loss of generality, let (0, 1) be assigned to a. Then, the minimum frequency that can be assigned to f is (0, 2) because f is at distance two from both a and e. We now need to assign frequencies to the central node d and the nodes of set B. Note that f has been assigned the highest frequency among all the previously assigned nodes a, e, and f. Node b of set B is at distance two from node f, i.e., it requires minimum frequency separation from f. Now, satisfying the frequency separation requirements with all the previously assigned nodes a, e, and f, the minimum frequency that can be assigned to node b is i) (0, 3) when $s_2 \le s_1 \le 2s_2$ and ii) (1, 1) when $s_1 \ge 2s_2$. By similar arguments, we can say that the minimum frequencies that can be assigned to nodes g and c are : i) (0, 4) and (0, 5), respectively, when $s_2 \le s_1 \le 2s_2$ and ii) (1, 2) and (1, 3), respectively, when $s_1 \ge 2s_2$.

Since the central node d is at distance one from all the remaining nodes, the minimum frequency that can

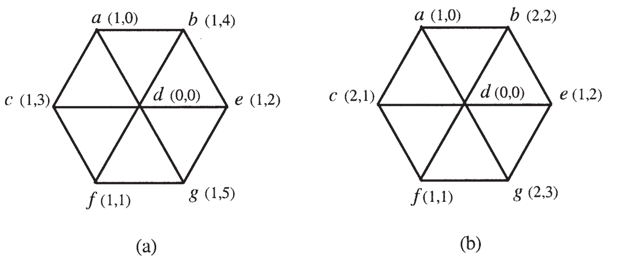


Fig. 19. Different frequency assignments to seven-node subgraph with minimum channel at central node for (a) $s_2 \leq s_1 \leq 2s_2$ and (b) $s_1 \geq 2s_2$.

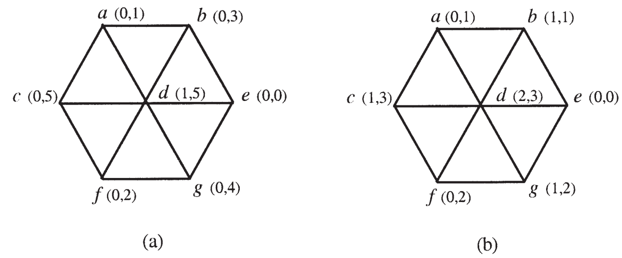


Fig. 20. Different frequency assignments to seven-node subgraph with maximum channel at central node for (a) $s_2 \leq s_1 \leq 2s_2$ and (b) $s_1 \geq 2s_2$.

be assigned to d is (1, 5) if $s_2 \leq s_1 \leq 2s_2$ and (2, 3) if $s_1 \geq 2s_2$.

If, after assigning frequencies to the nodes of set A, we first assign frequencies to the central node d followed by the nodes of set B, then, proceeding in a similar way, we see that the minimum bandwidth required will be (2, 4) for both the cases of i) $s_2 \leq s_1 \leq 2s_2$ and ii) $s_1 \geq 2s_2$.

Fig. 20(a) shows the assignment with a bandwidth of (1, 5) for $s_2 \leq s_1 \leq 2s_2$ and Fig. 20(b) shows the assignment with a bandwidth of (2,3) for $s_1 \geq 2s_2$.

Hence, whatever the case, all the nodes of the subgraph can be assigned frequency channels without any interference by using a bandwidth of $(s_1 + 5s_2)$ when $s_2 \leq s_1 \leq 2s_2$; or $(2s_1 + 3s_2)$ when $s_1 \geq 2s_2$. □

APPENDIX II
PROOF OF THEOREM 1

Let us consider the seven-node subgraph of a hexagonal cellular network as shown in Fig. 2. Suppose, using the frequency channels within the closed interval [0, p], it is possible to assign w frequency channels to each of the nodes of the subgraph satisfying all other constraints. Therefore, our objective is to find *minimum* p.

Note that any two nodes of the subgraph are within distance of two from each other. Therefore, any two frequencies assigned to two nodes of the subgraph must be separated by at least s_2. Suppose that the frequency channel y ($\in [0,p]$) is assigned to any node of the subgraph. Then, to satisfy the interference criteria, no frequency channel within the open interval $(y - s_2, y)$ and $(y, y + s_2)$ can be assigned to any node of the subgraph (including the node itself, since $s_0 \geq s_2$). That is, there would be an unusable gap of $(y - s_2, y)$ before y and also another such gap of $(y, y + s_2)$ after y, on the frequency spectrum line, where gap (a, b) implies that the integers within the open interval (a, b) cannot be assigned to any node of the subgraph. We refer to $(b - a)$ as the length of gap (a, b). Obviously, for $y = 0$ (or p), one of these gaps of unusable frequencies, e.g., $(y - s_2, y)$ [or $(y, y + s_2]$ will be beyond the interval [0, p].

In addition to that, the central node d is at distance one from each of the other nodes of the subgraph. Suppose, during the assignment, that a frequency channel x ($\in [0,p]$) is assigned to the central node d. Then, to satisfy the interference criteria, no frequency channel within the open interval $(x - s_1, x)$ and $(x, x + s_1)$ can be assigned to any node of the subgraph (including the central node itself, since $s_0 \geq s_1$). That is, there must be unusable gaps $(x - s_1, x)$ and $(x, x + s_1)$ before and after x, respectively. If $x = 0$ (or p), one of these gaps of unusable frequencies, e.g., $(x - s_1, x)$ [or $(x, x + s_1)$] falls beyond the interval [0, p].

Since $s_1 \geq s_2$, it follows from the above that we can have more usable frequency channels if the minimum and maximum frequencies are assigned to the central node d rather than to any other node.

It is always possible to assign either the minimum frequency or the maximum frequency to the central node d, but depending on the relative values of s_0, s_1, and s_2, it may or may not be possible to assign both of these to the central node for keeping p minimum.

Let us now consider the cases where we can do that and where we cannot.

Lemma 1 states that when $s_2 \leq s_1 \leq 2s_2$, $(s_1 + 5s_2)$ is the minimum bandwidth required to assign only one frequency channel to each of the nodes of the subgraph. Note that, for this assignment, the minimum frequency 0 [or, the maximum frequency $(s_1 + 5s_2)$] is assigned to the central node d as shown in Fig. 19(a) [or Fig. 20(a)]. Without loss of generality, let us start with the single channel assignment of Fig. 19(a). Next, we are left to assign $(w-1)$ additional frequency channels. In this process, we are to see whether the maximum frequency channel can also be assigned to the central node d, keeping p minimum.

We see that in Fig. 19(a), node g has been assigned the frequency $(s_1 + 5s_2)$. The frequency assigned to node a is the minimum of all assigned frequencies to the nodes at distance two from node g, and the frequency assigned to node d is the minimum of all assigned frequencies to the nodes at distance one from g. To assign the second channel to each node, we can start with assigning the next available minimum frequency either to a (a peripheral node) or to d (the central node). For the first case, the minimum frequency that can be assigned to a is $(s_1 + 5s_2) + s_2$, if $s_0 \leq 6s_2$ (to avoid interference between two channels assigned to a) and $(s_0 + s_1)$, otherwise. For the second case, the minimum frequency that can be assigned to d is $(s_1 + 5s_2) + s_1$, if $s_0 \leq (2s_1 + 5s_2)$, and s_0, otherwise. It follows from these observations that for $6s_2 \leq s_0 \leq (s_1 + 5s_2)$, it is more economical to start with assigning the minimum frequency for the second channel to node a than to node d. We now have the following cases.

Case 1) When $s_0 \leq 6s_2$ [or $6s_2 \leq s_0 \leq (s_1 + 5s_2)$]: To assign the second and successive channels to the nodes in Fig. 19(a), we first start with assigning the frequency $(s_1 + 6s_2)$ (for $s_0 \leq 6s_2$) or $(s_0 + s_1)$ (for $6s_2 \leq s_0 \leq (s_1 + 5s_2)$) to node a. Our objective is to end up with the maximum frequency at the central node

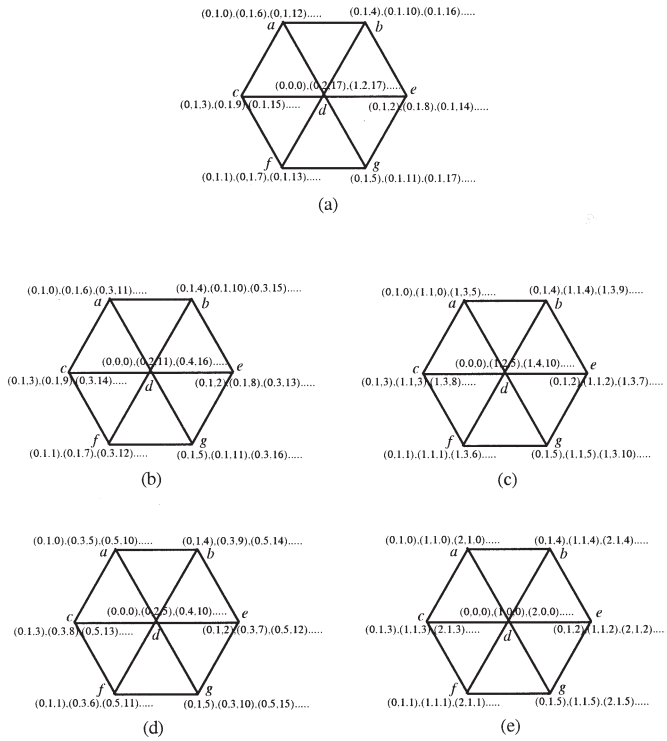


Fig. 21. Different frequency assignments to seven-node subgraph for $s_2 \leq s_1 \leq 2s_2$ when (a) $s_1 \leq s_0 \leq (2s_1 - s_2)$, (b) $(2s_1 - s_2) \leq s_0 \leq 6s_2$, (c) $6s_2 \leq s_0 \leq (s_1 + 5s_2)$, (d) $(s_1 + 5s_2) \leq s_0 \leq (2s_1 + 5s_2)$, and (e) $s_0 \geq (2s_1 + 5s_2)$.

d along with minimum gaps between successively assigned frequencies to keep the required p minimum. Let us now consider the following two subcases.

i) When $s_1 \leq s_0 \leq (2s_1 - s_2)$: After the completion of second-round channel assignment to all peripheral nodes $\{a, b, c, e, f, g\}$, if we go to assign a channel to the central node, then the minimum gap would be s_1. Also for the next round, we are to keep another gap of length s_1 before assigning to node a. Instead, it would be more economical to complete the assignments of all w channels to the peripheral nodes only, and after that assign the remaining $(w-1)$ channels to the central node.

As shown in Fig. 21(a), we assign the second-round channels to all the nodes except the central node as (0, 1, 7), (0, 1, 8), (0, 1, 9), (0, 1, 10), and (0, 1, 11) to the nodes f, e, c, b, and g, respectively. To assign the remaining $(w-2)$ channels to each of nodes $\{a, b, c, e, f, g\}$ of the subgraph, we follow the same ordering of nodes, i.e., a, f, e, c, b, and g so that each node is assigned with the next minimum frequency available at that time. Proceeding similarly for the third, fourth, $\ldots$, and wth channel, we can end up at node g with a frequency of $(s_1 + 5s_2) + (w - 1)(6s_2)$, for $s_1 \leq s_0 \leq (2s_1 - s_2)$. Now we are left to assign $(w-1)$ frequencies to the central node d. For that, the minimum frequency by which we can start at node d is $(s_1 + 5s_2) + (w - 1)(6s_2) + s_1$. Therefore, after the assignment of remaining $(w-2)$ frequencies, we can end up with a frequency of $(s_1 + 5s_2) + (w - 1)(6s_2) + s_1 + (w - 2)s_0$ at node d. Since we finally end up with the assignment of both the minimum and maximum frequencies to the central node d along with minimum gaps between successively assigned frequencies, the required p will be minimum whose value is given by: $(s_1 + 5s_2) + (w - 1)(6s_2) + s_1 + (w - 2)s_0$ $= (2s_1 + 5s_2) + (w - 2)(s_0 + 6s_2) + 6s_2$, when $s_1 \leq s_0 \leq (2s_1 - s_2)$.

Fig. 21(a) illustrates the assignments up to the third channel to each node following this method.

ii) When $(2s_1 - s_2) \leq s_0 \leq 6s_2$ [or $6s_2 \leq s_0 \leq (s_1 + 5s_2)$]: It would be more economical to complete the assignments of channels in each round to all the nodes, including the central node.

As shown in Fig. 21(b), (for $(2s_1 - s_2) \leq s_0 \leq 6s_2$), we assign the second-round frequencies (0, 1, 7), (0, 1, 8), (0, 1, 9), (0, 1, 10), (0, 1, 11), and (0, 2, 11) to the nodes f, e, c, b, and d, respectively. To assign the remaining $(w-2)$ channels to each of the nodes of the subgraph, we follow the same ordering of nodes, i.e., a, f, e, c, b, and g, so that each node is assigned with the next available minimum frequency at that time. Fig. 21(b) illustrates the assignments up to the third channel to each node following this method. Since we finally end up with the assignment of both the minimum and maximum frequencies to the central node d along with minimum gaps between successively assigned frequencies, the required p will be minimum whose value is given by:

a) $(s_1 + 5s_2) + (s_1 + 6s_2) + (w - 2)(s_1 + (s_1 + 5s_2)) = (w - 1)(2s_1 + 5s_2) + 6s_2$, for $(2s_1 - s_2) \leq s_0 \leq 6s_2$;
b) $(s_1 + 5s_2) + (s_0 + s_1) + (w - 2)(s_1 + (s_1 + 5s_2)) = (w - 1)(2s_1 + 5s_2) + s_0$, for $6s_2 \leq s_0 \leq (s_1 + 5s_2)$.

Fig. 21(c) illustrates the assignment when $6s_2 \leq s_0 \leq (s_1 + 5s_2)$ for $w = 3$.

Case 2) When $(s_1 + 5s_2) \leq s_0 \leq (2s_1 + 5s_2)$ [or $s_0 \geq (2s_1 + 5s_2)$]: To assign the second and successive channels to the nodes in Fig. 19(a), we first start with assigning the frequency $(2s_1 + 5s_2)$ (for $(s_1 + 5s_2) \leq s_0 \leq (2s_1 + 5s_2)$) or s_0 (for $s_0 \geq (2s_1 + 5s_2)$) to the central node d. However, in this case, after the assignment of the second channel to every node, we end up with assigning the maximum frequency to a noncentral node (say, g). This is illustrated in Fig. 21(d) (for $(s_1 + 5s_2) \leq s_0 \leq (2s_1 + 5s_2)$) and Fig. 21(e) (for $s_0 \geq (2s_1 + 5s_2)$), respectively, for $w = 3$. Proceeding similarly for the third, fourth, $\ldots$, and wth channel, we can end up at node g with a frequency of:

a) $(s_1 + 5s_2) + (2s_1 + 5s_2) + (w - 2)(s_1 + (s_1 + 5s_2)) = (w - 1)(2s_1 + 5s_2) + (s_1 + 5s_2)$, for $(s_1 + 5s_2) \leq s_0 \leq (2s_1 + 5s_2)$;
b) $(s_1 + 5s_2) + (s_0) + (w - 2)(s_0) = (w - 1)(s_0) + (s_1 + 5s_2))$, for $s_0 \geq (2s_1 + 5s_2)$.

Hence the proof. □

Appendix III
Proof of Theorem 2

We start from Fig. 19(b) and proceed in a similar way as in the proof of Theorem 1 to get the desired result.

Acknowledgment

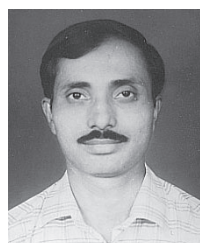
The authors would like to thank the anonymous reviewers for their help to improve this paper.

References

[1] A. Sen, T. Roxborough, and B. P. Sinha, "On an optimal algorithm for channel assignment in cellular network," in *Proc. IEEE Int. Conf. Communications*, Vancouver, Canada, June 6–10, 1999, pp. 1147–1151.
[2] W. K. Hale, "Frequency assignment: theory and application," *Proc. IEEE*, vol. 68, pp. 1497–1514, 1980.
[3] D. Bhandari, C. A. Murthy, and S. K. Pal, "Genetic algorithm with elitist model and its convergence," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 10, pp. 731–747, 1996.
[4] M. Duque-Anton, D. Kunz, and B. Ruber, "Channel assignment for cellular radio using simulated annealing," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 14–21, Feb. 1993.
[5] R. Mathar and J. Mattfeldt, "Channel assignment in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 647–656, Nov. 1993.
[6] D. Kunz, "Channel assignment for cellular radio using neural networks," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 188–193, Feb. 1991.
[7] A. Gamst, "Some lower bound for a class of frequency assignment problems," *IEEE Trans. Veh. Technol.*, vol. VT-35, pp. 8–14, Feb. 1986.
[8] D. Tcha, Y. Chung, and T. Choi, "A new lower bound for the frequency assignment problem," *IEEE/ACM Trans. Networking*, vol. 5, Feb. 1997.
[9] K. N. Sivarajan, R. J. McEliece, and J. W. Ketchum, "Channel assignment in cellular radio," in *Proc. 39th IEEE Vehicular Technology Conf.*, May 1989, pp. 846–850.
[10] W. Wang and C. K. Rushforth, "An adaptive local-search algorithm for the channel-assignment problem," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 459–466, Aug. 1996.
[11] S. Khanna and K. Kumaran, "On wireless spectrum estimation and generalized graph coloring," in *Proc. IEEE INFOCOM '98*, Apr. 1998.
[12] R. A. Leese, "A unified approach to the assignment of radio channels on a rectangular hexagonal grid," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 968–980, Nov. 1997.
[13] L. Narayanan and S. Shende, "Static frequency assignment in cellular networks," in *Proc. 4th Int. Colloq. Structural Information and Communication Complexity*, 1997.
[14] A. Sen, T. Roxborough, and S. Medidi, "Upper and lower bounds of a class of channel assignment problems in cellular networks," in *Proc. IEEE INFOCOM '98*, Apr. 1998.
[15] S. Kim and S. Kim, "A two-phase algorithm for frequency assignment in cellular mobile systems," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 542–548, Aug. 1994.
[16] D. E. Goldberg, *Genetic Algorithm: Search, Optimization and Machine Learning*. Reading, MA: Addison Wesley, 1989.
[17] C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithms," *Pattern Recognit. Lett.*, vol. 17, pp. 825–832, 1996.
[18] D. Beckmann and U. Killat, "A new strategy for the application of genetic algorithms to the channel-assignment problem," *IEEE Trans. Veh. Technol.*, vol. 48, pp. 1261–1269, July 1999.
[19] C. Y. Ngo and V. O. K. Li, "Fixed channel assignment in cellular radio networks using a modified genetic algorithm," *IEEE Trans. Veh. Technol.*, vol. 47, pp. 163–172, Feb. 1998.
[20] D.-W. Tcha, J.-H. Kwon, T.-J. Choi, and S.-H. Oh, "Perturbation-minimizing frequency assignment in a changing TDMA/FDMA cellular environment," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 390–396, Mar. 2000.
[21] W. K. Lio and G. G. Coghill, "Channel assignment through evolutionary optimization," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 91–96, Feb. 1996.
[22] J. S. Kim, S. H. Park, P. W. Dowd, and N. M. Nasrabadi, "Cellular radio channel assignment using a modified hopfield network," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 957–967, Nov. 1997.
[23] C. W. Sung and W. S. Wong, "Sequential packing algorithm for channel assignment under co-channel and adjacent-channel interference constraint," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 676–686, Aug. 1997.
[24] J. S. Kim, S. H. Park, P. W. Dowd, and N. M. Nasrabadi, "Channel assignment in cellular radio using genetic algorithm," *Wireless Personal Commun.*, vol. 3, no. 3, pp. 273–286, Aug. 1996.
[25] T.-M. ko, "A frequency selective insertion strategy for fixed channel assignment," *Proc. 5th IEEE Int. Symp. Personal, Indoor and Mobile Radio Commun.*, pp. 311–314, Sept. 1994.
[26] N. Funabiki and Y. Takefuji, "A neural network parallel algorithm for channel assignment in cellular radio network," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 430–437, Nov. 1992.
[27] R. Battiti, A. Bertossi, and D. Cavallaro, "A randomized saturation degree heuristic for channel assignment in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 50, pp. 364–374, Mar. 2001.
[28] K. Smith and M. Palaniswami, "Static and dynamic channel assignment using neural network," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 238–249, Feb. 1997.
[29] G. Chakraborty, "An efficient heuristic algorithm for channel assignment problem in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 50, pp. 1528–1539, Nov. 2001.
[30] S. C. Ghosh, B. P. Sinha, and N. Das, "A new approach to efficient channel assignment for hexagonal cellular networks," *Int. J. Foundations Comp. Sci.*, vol. 14, pp. 439–463, June 2003.

Sasthi C. Ghosh received the B.Sc. degree in mathematics from the University of Calcutta, India, in 1996 and the M.C.A. degree from Jadavpur University, India, in 1999.

He is currently a Senior Research Fellow in the Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Calcutta. His current research interests include mobile computing, wireless communication, distributed computing, and ad hoc networks.



Bhabani P. Sinha (M'84–SM'94–F'02) received the M.Tech. and Ph.D. degrees from the University of Calcutta, India, in 1975 and 1979, respectively.

He joined the Faculty of the Electronics Unit, Indian Statistical Institute, Calcutta, in 1976, where he is currently a Professor in the Advanced Computing and Microelectronics Unit. He has published more than 100 research papers in the areas of computer architecture, fault diagnosis, graph algorithms, interconnection networks, parallel processing, and mobile computing. His current research interests are in the areas of parallel architectures and algorithms and mobile computing. He was an Alexander von Humboldt Fellow of Germany during 1979–1981 and during 1990–1991. He was a Visiting Faculty Member of Southern Illinois University, Carbondale, (1986–1987), University of Central Florida, Orlando (1998), and Clemson University, Clemson, SC (2001–2002). He held the Indian Space Research Organization Chair Professor position at the Indian Institute of Technology, Kharagpur, in 2000.

Nabanita Das (M'00) received the B.Sc. (Hons.) degree in physics and the B.Tech. degree in radio physics and electronics from the University of Calcutta, India, in 1976 and 1979, respectively. She received the M.E. degree in electronics and telecommunication engineering and the Ph.D degree in computer science from Jadavpur University, India, in 1981 and 1992, respectively.

Since 1986, she has been on the Faculty of the Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Calcutta. She visited the Department of Mathematik and Informatik, University of Paderborn, Germany, under the INSA scientists' exchange program. Her research interests include parallel processing, interconnection networks, wireless communication, and mobile computing.