Solving the Fixed Channel Assignment Problem in Cellular Communications Using An Adaptive Local Search

Graham Kendall and Mazlan Mohamad

Automated Scheduling, Optimisation and Planning (ASAP) Research Group, School of Computer Science and IT, University of Nottingham, Nottingham NG8 1BB, UK

Abstract. The Channel Assignment Problem can be defined as assigning a minimum number of radio frequencies to a set of transceiver/receiver units without violating given constraints, in particular the frequency separation that must exist between two given channels to avoid interference. Being an NP-complete problem, finding good quality solutions increases in difficulty as the number of transceiver/receiver units increase. Previous approaches for solving the channel assignment problems have used graph colouring, heuristic approaches, local search, meta-heuristics and genetic algorithms. In this paper, we present a greedy local search, combined with a monte carlo algorithm as an acceptance criteria. Our results are able to match lower bound conditions and beat existing approaches. Computational results are given.

1. Introduction

The use of cellular communication technology continues to grow and shows no signs of slowing up in the foreseeable future. With the ability to provide instant connectivity at anytime, anywhere in the world, cellular communications is gradually becoming the communication method of choice for many users. This can be witnessed by the proliferation of mobile handsets and the prediction that many people will dispense with their landline, preferring instead to rely on their mobile handset. As the requirement for cellular communication increases, efficient usage of the limited radio channels available is necessary in order to cope with the additional services and additional mobile subscribers. To meet the increase in channel demand, mobile operators have to increase the number of base stations. This, in turn, leads to the need to utilise cell splitting (a cell is the area covered by a base station) techniques and optimise frequency reuse (use the same channel repeatedly). Of course, the operators also have to maintain a minimum level in the quality of service that they offer [1].

In cellular communication, a duplex traffic channel is established for receiving and transmitting a signal between a base station and a mobile terminal. The mobile terminal, situated within the base station coverage area (cell), is able to use that channel for a voice call or for data communication. The same channel (co-channel) can be simultaneously used in other base stations subject to a minimum reuse distance between channels. A complete cellular communication network consists of thousand of cells (each with a base station) in order to cover the large, geographic, area required to give as a wide coverage as possible to the end users. In fact, the network is split into clusters (typically containing 21, 25 or 55 base stations) and the frequency assignments in one cluster can be applied to other clusters in that network.

With a limited frequency spectrum available, the main task of cell design is to optimise the use of the available frequency bandwidth. This is known as the channel assignment problem and is an active area of research (for example, see [2]).The channel assignment problem can be defined as assigning a minimum number of radio frequencies to a set of transceiver/receiver units without violating given constraints. In particular a frequency separation must exist between two given channels to avoid interference. Being an NP-complete problem, finding good quality solutions increases in difficulty as the number of transceiver/receiver units increase.

The channel assignment problem can be categorised into three different forms, these being Fixed Channel Assignment, Dynamic Channel Assignment and Hybrid Channel Assignment (a combination of fixed and dynamic) [3]. In fixed channel assignment, the channels are permanently assigned to the base stations based on predetermined traffic demand and interference constraints. Therefore, in the fixed problem, it is difficult to adapt to any changes in either channel demand or interference. In dynamic channel assignment, the channels are placed into a central pool and are dynamically assigned upon request by a base station. Once the call is completed, the channel will be returned to central pool and can be used by another base station. Dynamic assignment provides flexibility and traffic adaptability at the cost of higher complexity. Also, under heavy traffic conditions, dynamic strategies are less efficient when compared to fixed strategies [3]. Since heavy traffic is expected in the future, the efficiency of fixed schemes is highly desirable [4]. The approach discussed in this work utilises a fixed channel assignment method.

2. Related Work

Many researchers have investigated the fixed channel assignment problem, utilising approaches such as graph theory [5], heuristic approaches [6,7,8], local search [9] and meta-heuristics [10,11,12,13], including genetics algorithms [14].

Zoellner and Bell [5] used node-colour and node-degree to rank the ordering of the cells. They used two assignment strategies, frequency exhaustive strategy and requirement exhaustive strategy to assign the channels. In the frequency exhaustive strategy, each call is assigned with the least the possible channel assignment. In the requirement exhaustive strategy, the first call is assigned to channel 1 and every other call is checked to see if it can also be assigned channel 1. The channel number is then incremented and the process repeats until all calls have been assigned to a frequency. Box [6] used a simple iterative technique to assign channels according to assignment difficulty. His approach ranked the channels based on how difficult it would be to assign that channel to a base station. Sivarajan et al. [7], used the same approach as [6], namely a sequential cell ordering to give eight channel assignment algorithms. They express the channel assignment problem as a minimum span problem which can be viewed as a generalised graph colouring problem. They used two ordering of the calls, row-wise ordering and column-wise ordering with the combination of cells ordering proposed by Zoellner and Bell [5]. Their approach gave better solutions than existing algorithms.

Wang and Rushforth [9] implemented a two-phase adaptive local search algorithm. They used a deterministic-probability neighbour-generation method to create a new neighbour configuration. They select a call with maximum frequency, and swap with another call which is randomly selected from the call list, to give a new neighbour configuration. The algorithm has been applied to several existing benchmark problems (see table 6) and the solutions obtained outperformed existing algorithms. Chakraborty [8] proposed a fast heuristic algorithm that created a pool of valid solutions using a quandary representation. He tested the algorithm against twenty benchmarks problems (see table 6) and set the current benchmarks that we compare against in this paper.

Kunz [10] used a neural network algorithm based on Hopfield and Tank's model to minimise the cost function. Funabiki and Takefuji [11] proposed a parallel algorithm based on neural networks. The algorithm, which does not require a rigorous synchronisation procedure, runs on sequential and parallel machines. Kim et. al [12] used a modified discrete Hopfield neural network to avoid getting stuck in local minima.

Duque-Anton et. al [13] proposed a simulated annealing approach. The interference relationship between cells was represented by a cost function, which they attempted to minimise. Lai and Coghill [14] used a genetic algorithm approach. They used a string structure to represent the channel required for each base station, where the total length of each string is the sum of channels required. They used partially matched crossover and a basic mutation probability, with two extra parameters in order to bias co-site and co-channel constraints in their fitness function. They claimed that their approach is elegant and simple. In addition, new rules can be easily added without corrupting existing rules.

3. Problem Description

Radio channels are represented by the positive integers 1, 2, 3,…,m where m is a maximum allocation of the spectrum bandwidth. The basic model of the channel assignment problem can be represented as follows (mostly adopted from [7,9])

a. N : The number of cells in the network.

b. d_i : The number of radio channel required in cell i ($1 \le i \le N$) in order to satisfy channel demand.

c. C : Compatibility matrix, $C=(c_{ij})_{NxN}$ denotes the frequency separation required between cell i and cell j

d. $Call_{ik}$: Cell i with call k where $1 \le i \le N$, $1 \le k \le d_i$.

e. f_{ik} : A radio channel is assigned to $Call_{ik}$, where $f_{ik} \in$ a set of radio channel F.

f. Frequency separation constraint - $| f_{ik} - f_{jm} | \geq c_{ij,}$ for all i,j,k,m ($i \neq j$,$k \neq m$), c_{ij} is defined in Compatibility Matrix, C. If $i=j$, it's become co-site constraint.

g. *TotalAssignCh:* The total of radio channel to be assigned in the system can be shown as

$$TotalAssignCh = \sum_{i=1 to N} d_i$$

Therefore, the objective of MS-CAP is [15]

Minimise m

Subject to

$\sum_{k=1}^{m} f_{ik} = d_i$ for $1 \leq i \geq N$

$|k - l| \geq C_{ij}$ for $1 \leq k,l \geq m$ and $1 \leq i,j \geq N$ such that $f_{ik}=f_{jl}=1$

$f_{ik} = 0$ if channel k is not assigned to cell i, otherwise 1, for $1 \leq k,l \geq m$ and $1 \leq i,j \geq N$

3.1 Example

We consider here an example taken from Sivarajan et. al [7], in order to further explain the notation presented above. We also show how this example can be represented as a graph colouring problem. This problem consists of four cells.

$$C = \begin{pmatrix} 5 & 4 & 0 & 0 \\ 4 & 5 & 0 & 1 \\ 0 & 0 & 5 & 2 \\ 0 & 1 & 2 & 5 \end{pmatrix} \qquad D = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 3 \end{pmatrix}$$

Where C is compatibility matrix and D is the traffic demand vector.

In order to make it clearer, the following descriptions can be read in conjunction with the problem description above.

a. The number of cells in the network, N=4.

b. The number of radio channels required in each cell

Cell *1*	Cell *2*	Cell *3*	Cell *4*
1 channel	1 channel	1 channel	3 channels

Table 1. Channel demand

c. Minimum frequency separation required between cell *i* and cell *j*

	Cell *1*	Cell *2*	Cell *3*	Cell *4*
Minimum Frequency separation distance	Co-site : 5 Cell 2/4	Co-site : 5 Cell 1/4 Cell 4/1	Co-site : 5 Cell 4/2	Co-site : 5 Cell 3/2 Cell 2/1

Table 2. Constraints based on compatibility matrix C

Notation: The co-site constraint shows the minimum separation between two frequencies assigned to the same cell. *i/j* indicates that for the cell to which that column refers, there must be a separation of *j* between that cell and cell *i*.

d. The call list that needs to be assign to each cell

	Cell *1*	Cell *2*	Cell *3*	Cell *4*
Call list	$Call_{11}$ ($i=1,k=1$)	$Call_{21}$ ($i=2,k=1$)	$Call_{31}$ ($i=3,k=1$)	$Call_{41}$, $Call_{42}$, $Call_{43}$ ($i=4, 1\leq k\geq 3$)

Table 3. Call list

e. A radio channel is assigned to serve a call in the each cell

	Cell *1*	Cell *2*	Cell *3*	Cell *4*
Call list	1 call	1 call	1 call	3 calls
Radio channel, f_{ik}	f_{11}	f_{21}	f_{31}	$f_{41} f_{42} f_{42}$

Table 4. Radio channel at every cell

f. Frequency separation constraint

	Cell *1*	Cell *2*	Cell *3*	Cell *4*
Cell 1	-	$\|f_{21}-f_{11}\|\geq 4$	-	-
Cell 2	$\|f_{11}-f_{21}\|\geq 4$	-	-	$\|f_{4k}-f_{21}\|\geq 1$
Cell 3	-	-	-	$\|f_{4k}-f_{31}\|\geq 2$
Cell 4	-	$\|f_{21}-f_{4k}\|\geq 1$	$\|f_{31}-f_{4k}\|\geq 2$	$\|f_{4k}-f_{4m}\|\geq 5$

note: $1\leq k,m\geq 3$

Table 5. Frequency channel separation constraint

g. The total of radio channel that is to be assigned

$TotalAssignCh= (d_1+d_2+d_3+d_4)=6$

We can model the above example as a graph coluring problem. Let define the vertices C of graph G consist of set of calls to be assigned (in this case 6 calls i.e. $call_{11}$, $call_{21}$, $call_{31}$, $call_{41}$, $call_{42}$, $call_{43}$). An edge, E → e($call_{ij}$,$call_{kl}$) for graph G can be defined as a constraint between $call_{ij}$ and $call_{kl}$. These is shown in figure 1:

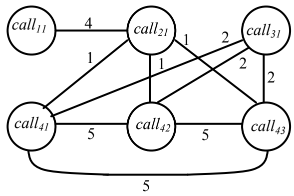

Figure 1: CAP Graph representation

The task is to schedule the calls such that each 'clashed' call will be assigned a different channel. Using the concept of graph colouring, the channel is represented by the *colour* and our objective function is to minimise the number of colours used. Therefore, this problem can be solved using graph colouring method which is widely used in scheduling and timetabling as well as in the telecommunication industry.

In general, we have two objectives in solving the channel assignment problem, namely (from [16]):

(1). Given a traffic demand, base station number and compatibility matrix, find the minimum number of frequency channels with free engineering interference i.e.

Minimise the total bandwidth (span) of radio channels
s.t. traffic demand and interference constraint (CAP1)

(2). Given a number of radio channels, a number of base stations, traffic demand and compatibility matrix, minimise severity of channel interference i.e.

Minimise severity of channel interferences
s.t. demand constraints (CAP2)

Two solution examples, based on the above example are shown below [16]:

a) Solution of CAP1

Channel Number

Cell Number		1	2	3	4	5	6	7	8	9	10	11
	1										■	
	2			■								
	3				■							
	4	■					■					■

Figure 2: An interference free assignment

b) Solution of CAP2 (given only 10 channels)

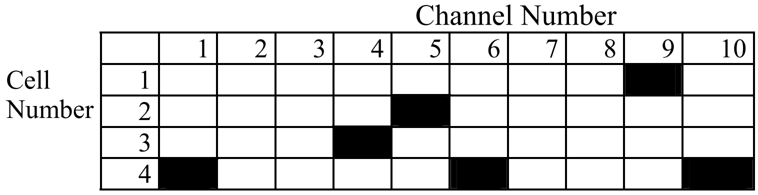

Figure 3. A near interference free assignment

At cell 4, there is weak interference between channel 6 and channel 10. The difference between channel 6 and channel 10 is 4 but the co-site constraints at cell 4 requires a separation of 5 (c_{44}=5).

In our approach, our objective function is to obtain a minimum span (CAP1). The minimum span is determined by call or vertex ordering. For example, referring to the above example, if the call ordering is {$call_{11}$, $call_{21}$, $call_{31}$, $call_{41}$, $call_{42}$, $call_{43}$}, the span is 13 but if call ordering is {$call_{41}$, $call_{42}$, $call_{43}$, $call_{21}$, $call_{31}$, $call_{11}$}, the span is 11.

4. The Algorithm

We use the local search framework proposed by Wang and Rushforth [9] to define the solution space and objective function.

Solution Space S : The set of possible ordered list of calls.
Objective function : $f(S)_{max}$ Є *m*, the maximum frequency for solution *S* Є S.
n : neighbourhood structure.

In our proposed method, we have a two stage algorithm, namely a probabilistic stage and a new neighbour generation stage.

4.1 Probabilistic stage

In this stage, all the information such as network size, *N*, channel demand, *D* and compatibility matrix *C* are initialised. The next step is to generate a random number of calls to be assigned first and then use a frequency exhaustive strategy to assign the channel to the selected call. This step will be repeated until all the call assignments are completed to give an initial call order list S_0 Є S and initial objective function $f(S_0)$ Є *m.* The algorithm is shown below:

Step 1 : (Initialisation)
(A) Choose initial solution S_0 Є S;
(B) Record the best obtain solution $S_{best} = S_0$ and $f(S_{best}) = f(S_0)$;

4.2 New neighbour generation stage

With the initial call list, s_o, from the probabilistic stage, the algorithm now uses a deterministic approach and selects the last call, $call_{ik\ max}$ (that is the call with the highest assigned channel number, $f(S_0)_{max}$). This call is deleted from the call list and a new location is sought where it can be re-inserted, starting from the beginning of the list. It will create new neighbour, S_{new} with objective function, $f(S_{new})$. This step will be repeated until a stopping criteria is met, which is either:

- The lower bound from the benchmark problem is obtained.
- The maximum number of iterations, $iter_{max}$, is reached. We defined , $iter_{max} =$ Cell size * Minimum demand.
- RunTime is expired (we use 400 seconds).

The algorithm is shown below (continue from *section 4.1 step 1*):

Step 2 : (Choice and termination)
(A) Choose an S_{new} Є $n(S_0)$;
(B) Compute $\delta = f(S_{new}) - f(S_0)$;
(C) If the acceptance criteria is true, then accept S_{new} (and proceed to Update);
(D) If S_{new} is rejected and stopping condition=false, then return to Step2;
(E) Terminate by a stopping condition.

(Update)
Re-set $S_0 = S_{new}$, and if $f(S_{new})<f(S_{best})$, perform step 1(B) : Initialisation;
Return to step 2 if stopping condition=false.

4.3 Acceptance Criteria

The new solution acceptance criterion is important in order to escape from local minima. In our approach, we use an Exponential Monte Carlo with counter (EMCq) as an acceptance criteria. This 'parameter free' acceptance criteria is extended from Monte Carlo method discuss in [17] has shown to be effective in another domain (printed circuit board assembly) [18].

The basic EMCq acceptance criteria can be described as follows (mostly adopted from [18]).

1. Compute $\alpha = f(s_{new}) - f(s_o)$
2. If $\alpha \leq 0$, accept s_{new}(update *objective function*, $f(s_{best}) = f(s_{new})$)
3. Else: Accept s_{new} with EMCq probability. If s_{new} is rejected and stopping condition=false, generate new neighbour.

The EMCq probability is computed by $e^{-\theta\psi}$

Where : $\theta = \alpha * t$ t = computation time
ψ = counter of consecutive non improvement iterations

The idea behind this acceptance criteria is that we only accept non-improving moves after all the neighbours of the current solution has been searched without any improvement in solution quality. When α is small and we have not found a better solution for a long time, non-improving moves are more likely to be accepted [18].

For the comparison, we also used random descent (RD) and steepest descent acceptance criteria (SD). RD will accept only first improving move meanwhile SD will only accept the best move in neighbour structure.

5. Testing and Results

We have implemented and tested the above algorithm on a Pentium III-700 MHz computer. We have compared our performance with [8], which proposed to generate a population of random valid solutions using a quadnary representation [0,+1,-1,+9] (these values represent the following, {assignable ,used ,unassignable ,unused}.

We implemented three different network sizes {21, 25, 55} with different compatibility matrices, C, and traffic demands, D. In this work we use a random constructive heuristic to generate the initial solution. For the purpose of comparison of our algorithm, we used three different initial solutions. The results as presented in table 6, 7 and 8.

Test	C_Matric(C_{ij})/ Demand(D_i) [8]	Trivial Lower Bound	Best Solution by [8]	Initial Span	Best Result	Time Taken to produce best result (s)
1	C1_21/D1_21	533	533	539/553/541	533/533/533	0.9/0.2/0.3
2	C1_21/D2_21	309	309	313/310/313	309/309/309	0.1/0.1/0.1
3	C2_21/D1_21	533	533	616/614/641	533/533/533	0.3/0.4/0.5
4	C2_21/D2_21	309	309	335/352/359	309/309/309	1.0/1.7/0.3
5	C3_21/D1_21	457	457	460/467/489	457/457/457	0.1/0.1/0.2
6	C3_21/D2_21	265	265	267/272/270	265/265/265	0.3/0.1/0.1
7	C4_21/D1_21	457	457	602/597/586	457/457/457	4.4/4.2/5.7
8	C4_21/D2_21	265	280	331/340/344	273/273/273	192/209/75
9	C5_21/D1_21	381	381	417/422/431	381/381/381	0.2/0.2/0.1
10	C5_21/D2_21	221	221	236/234/245	221/221/221	0.2/0.1/0.4
11	C6_21/D1_21	381	463	593/541/595	440/435/436	211/344/110
12	C6_21/D2_21	221	273	318/333/343	269/268/269	60/71/27
13	C7_21/D1_21	305	305	354/370/371	305/305/305	1.2/2.0/1.8
14	C7_21/D2_21	177	197	208/221/221	185/185/185	118/22/17
15	C8_21/D1_21	305	465	535/538/522	447/448/444	43/117/364
16	C8_21/D2_21	177	278	324/320/349	271/273/272	33/76/371
17	C1_25/D3_25	21	73	79/77/78	73/73/73	0.4/0.2/0.2
18	C1_25/D4_25	89	121	216/208/216	200/200/200	49/1.0/64
19	C1_55/D5_55	309	309	345/344/333	309/309/309	0.9/1.0/0.8
20	C1_55/D6_55	71	79	102/115/90	73/73/72	2.6/17/12

Table 6. Results over 3 different initial solutions using EMCq acceptance criteria

Test	C_Matric(C_{ij})/ Demand(D_i) [8]	Trivial Lower Bound	Best Solution by [8]	Initial Span	Best Result	Time Taken to produce best result (s)
1	C1_21/D1_21	533	533	539/553/541	533/533/533	0.2/0.2/0.3
2	C1_21/D2_21	309	309	313/310/313	309/309/309	0.1/0.1/0.1
3	C2_21/D1_21	533	533	616/614/641	533/533/533	0.5/0.4/0.4
4	C2_21/D2_21	309	309	335/352/359	309/309/309	0.3/0.4/0.7
5	C3_21/D1_21	457	457	460/467/489	457/457/457	0.1/0.1/0.1
6	C3_21/D2_21	265	265	267/272/270	265/265/265	0.1/0.1/0.1
7	C4_21/D1_21	457	457	602/597/586	457/457/457	2.7/2.6/5.2
8	C4_21/D2_21	265	280	331/340/344	280/277/278	3.7/2.8/3.4
9	C5_21/D1_21	381	381	417/422/431	381/381/381	0.2/0.1/0.3
10	C5_21/D2_21	221	221	236/234/245	221/221/221	0.2/0.1/0.1
11	C6_21/D1_21	381	463	593/541/595	451/456/455	8.2/3.4/8.1
12	C6_21/D2_21	221	273	318/333/343	278/274/275	30/3.8/2.7
13	C7_21/D1_21	305	305	354/370/371	305/305/305	1.1/2.0/1.6
14	C7_21/D2_21	177	197	208/221/221	187/186/188	6.2/3.2/2.2
15	C8_21/D1_21	305	465	535/538/522	447/448/444	43/117/364
16	C8_21/D2_21	177	278	324/320/349	277/282/282	7.7/4.5/5.1
17	C1_25/D3_25	21	73	79/77/78	73/73/73	11/16/30
18	C1_25/D4_25	89	121	216/208/216	201/201/201	7.1/2.0/6.6
19	C1_55/D5_55	309	309	345/344/333	309/309/309	1.1/1.0/0.7
20	C1_55/D6_55	71	79	102/115/90	75/74/75	1.4/0.7/1.8

Table 7. Results over 3 different initial solutions using RD acceptance criteria

Test	C_Matric(C_{ij})/ Demand(D_i) [8]	Trivial Lower Bound	Best Solution by [8]	Initial Span	Best Result	Time Taken to produce best result (s)
1	C1_21/D1_21	533	533	539/553/541	533/533/533	0.2/6/3.3
2	C1_21/D2_21	309	309	313/310/313	309/309/309	0.3/0.1/0.1
3	C2_21/D1_21	533	533	616/614/641	533/582/601	46/14/1.4
4	C2_21/D2_21	309	309	335/352/359	328/341/329	0.9/0.4/4.7
5	C3_21/D1_21	457	457	460/467/489	457/457/457	1/2.8/16
6	C3_21/D2_21	265	265	267/272/270	265/265/265	0.2/0.3/0.1
7	C4_21/D1_21	457	457	602/597/586	562/494/569	16/71/10
8	C4_21/D2_21	265	280	331/340/344	316/310/313	3.2/3.3/10
9	C5_21/D1_21	381	381	417/422/431	381/381/381	24/28/36
10	C5_21/D2_21	221	221	236/234/245	226/223/231	3.2/1.5/3.2
11	C6_21/D1_21	381	463	593/541/595	550/492/556	51/46/25
12	C6_21/D2_21	221	273	318/333/343	300/300/295	27/27/16
13	C7_21/D1_21	305	305	354/370/371	338/350/336	18/25/44
14	C7_21/D2_21	177	197	208/221/221	200/210/201	5.8/9.5/16
15	C8_21/D1_21	305	465	535/538/522	531/533/495	0.3/49/18.7
16	C8_21/D2_21	177	278	324/320/349	307/293/316	12/27/25
17	C1_25/D3_25	21	73	79/77/78	76/74/76	0.7/0.6/0.4
18	C1_25/D4_25	89	121	216/208/216	212/204/205	6.5/2.3/22
19	C1_55/D5_55	309	309	345/344/333	318/327/317	37/16/12
20	C1_55/D6_55	71	79	102/115/90	81/80/87	7.5/7.1/1.3

Table 8. Results over 3 different initial solutions using SD acceptance criteria

6. Discussion

In the above tables we have shown the 20 instances test which have been used in previous work. In the 4th column, the previous best result of previous work [8] is shown. We only make a comparison with [8] because it is the only previous work that used all 20 instances. In 5th column (initial span), we show the three different initial solutions, followed by our best achieved result and the time taken in column 6 and 7.

Table 6 shows the result of the greedy local search with an EMCq acceptance criteria. For the easier test instances (test 1 to 7, 9, 10 and 13), our algorithm is able to match the calculated lower bound and the performance is similar to the previous approach in [8]. We also can see that with different initial solutions, our algorithm manages to converge to the lower bound for all initial solutions. For the difficult benchmarks problem such as test 11 and 12, even though we cannot match the lower bound, but we can see that the percentage of improvement from initial span is around 25% and 18% respectively.

Table 7 shows the result of random decent local search. In this case, the algorithm only accepts a first improving. The performance of this acceptance criteria is similar to EMCq acceptance criteria for the easier problem, but for the difficult problems, it gets stuck in a local minima due to their being no diversification in the search strategy. For example, for test 11, the best span is 451 compare to 435 for EMCq.

Table 8 shows the result of steepest descent local search. The performance of SD was outperformed by EMCq and RD.

7. Conclusions and Future work

The channel assignment problem is one of the real world problems within the telecommunications industry. The number of available channels is fixed but demand is increasing due to additional mobile subscribers. In our study, we use simple local search with EMCq acceptance criteria and compare with random descent and steepest decent acceptance criteria. Based on the experimental results, we conclude that EMCq is superior when compared to random descent and a steepest decent acceptance criteria.

Our future work will refine our algorithm in order to further improve the results. We are considering the following areas.

- Using different strategies to generate initial solutions and study the effect of using good initial solution because currently, we are using randomly generated solutions.
- Using a different neighbour generating strategy such re-insert the call based on random selection.
- Implement additional acceptance criteria to compare against those (EMCq,RD,SD) reported in this paper.

8. References

[1] T.Fujii and N. Nakazima, Radio Channel Assignment System for Cellular Mobile Communications, *Electronics and Communications in Japan*, Part 1, Vol. 86, No. 5, 2003.

[2] X.N.Fernando and A.O. Fapojuwo, A Viterbi-Like Algorithm with Adaptive Clustering for Channel Assignment in Cellular Radio Networks, *IEEE Trans on Veh Technology*, vol 51, pp. 73-87, January 2002.

[3] I.Katzela and M.Naghshineh, Channel Assignment Schemes for Cellular Mobile Telecommunication Systems - A Comprehensive Survey, *IEEE Personal Communications Magazine*, June 1996.

[4] D.Corne, M.J. Oates and G.D Smith , Telecommunication Optimisation : Heuristic and Adaptive Techniques, *John Wiley & Sons*, 2000.

[5] J.A Zoellner and C.A Beall, A breakthrough in spectrum conserving frequency assignment technology, *IEEE Trans. Electromagnetic Comp.*,vol. EMC-19, pp. 313-319, Aug. 1977.

[6] F.Box, A Heuristic technique for assigning frequencies to mobile radio networks, *IEEE Trans. Veh. Technology*, vol. VT-27, pp. 57-64, 1978.

[7] K.N.Sivarajan, R.J.McEliece and J.W.Ketchum, Channel assignment in cellular radio, *in Proc. 39th IEEE Veh. Technology. Soc. Conf.*, pp. 846-850, May 1989.

[8] G.Chakraborty, An Efficient Heuristic Algorithm for Channel Assignment Problem in Cellular Radio Networks, *IEEE Trans on Veh Technology*, vol. 50, no. 6, pp. 1528-1539, Nov 2001.

[9] W.Wang and C.K Rushforth, An Adaptive Local-Search Algorithm for the Channel-Assignment Problem (CAP), *IEEE Trans on Veh Technology*, vol 45, pp. 459-446, August 1996.

[10] D.Kunz , Channel Assignment for Cellular Radio Using Neural Networks, *IEEE Trans on Veh Technology*, vol 40, pp. 118-193, Feb 1991.

[11] N.Funabiki and Y.Takefuji, A Neural Network Parallel Algorithm for Channel Assignment Problems in Cellular Radio Network, *IEEE Trans on Veh Technology*, vol. 41, no. 4, pp. 430-437, November 1986.

[12] J.S.Kim, S.H.Park, P.W.Dowd and N.M.Nasrabadi, Cellular radio channel assignment problems in cellular networks, *IEEE Trans on Veh Technology*, vol 46, pp. 957-967, Nov. 1996.

[13] M.Duque-Anton, D.Kunz and B.Ruber, Channel Assignment for Cellular Radio Using Simulated Annealing, *IEEE Trans on Veh Technology*, vol 42, pp. 14-21, Feb 1993.

[14] W.K.Lai, G.G.Coghill, Channel Assignment Through Evolutionary Optimisation, *IEEE Trans on Veh Technology*, vol 45, No.1, pp. 91-95, Feb 1996.

[15] B. Dirk and K. Ulrich, A New Strategy for Application of Genetic Algorithms to the Channel-Assignment Problem. *IEEE Trans on Veh Technology,* vol 48, no. 4, pp. 1261-1269, July 1999.

[16] K.Smith and M.Palaniswami, Static and Dynamic Channel Assignment Using Neural Networks, *IEEE Journal On Selected Areas In Communication*, Vol. 15, No.2, pp. 238-249, February 1997.

[17] Glover, F. and Laguna, M., Tabu search, chapter 3. In: Reeves, C. R.(ed) Modern heuristic techniques for combinatorial problems, *McGraw-Hill*, 1995, pp 70-150.

[18] M. Ayob and G.Kendall, A Monte Carlo Hyper-Heuristic To Optimise Component Placement Sequencing For Multi head Placement Machine, *InTech'03 Thailand*, pp. 132-141, ISBN 974-658-151-1.