A memetic algorithm for channel assignment in wireless FDMA systems

Sung-Soo Kim[a], Alice E. Smith[b,*], Jong-Hyun Lee[c]

[a]*Systems Optimization Laboratory, Department of Industrial Engineering, Kangwon National University, Chunchon, 200-701, Korea*
[b]*Industrial and Systems Engineering, Auburn University, AL 36849-5346, USA*
[c]*Nespot Service Division/Nespot Project Team Service Development Laboratory, KT 17 Woomyun-Dong, Seocho-Gu, Seoul, 137-792, Korea*

Abstract

A new problem encoding is devised for the minimum span frequency assignment problem in wireless communications networks which is compact and general. Using the new encoding, which reduces search space dramatically over previous problem encodings, an optimization algorithm is developed which combines a genetic algorithm global search with a computationally efficient local search method from the literature. This memetic algorithm is shown to be more effective than six previous approaches in the literature on a suite of established test problems. Further, it shown that the integration of the global search with the local search is important; neither component by itself is nearly as effective.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Channel assignment; Memetic algorithm; Genetic algorithm; Wireless communications systems

1. Introduction

There is a rapidly growing demand for wireless telecommunications; however, the restricted number of channels is a significant bottleneck of mobile cellular systems capacity. Consequently, when assigning channels to base stations, it is desirable to reuse the same channel as much as possible in frequency division multiple access (FDMA) wireless communications systems.

---

* Corresponding author. Tel.: +1 334 844 1400; fax: +1 334 844 1381.
  E-mail address: aesmith@eng.auburn.edu (A.E. Smith).

The channel assignment problem (assigning communications channels to wireless cells) is known to be NP-hard. There are three types of constraints that affect the problem. The co-channel constraint (CCC) restricts using the same channel in different cells within a certain distance from each other. The co-site constraint (CSC) defines the minimum channel distance that must separate channels assigned to the same cell. The adjacent channel constraint (ACC) specifies the minimum cell distance that adjacent channels must have. These constraints can be represented with a compatibility matrix, $\mathbf{C}$, that shows the minimum separation in channels required between all calls in all cells.

The basic channel assignment problem model consists of the following components [1]:

(1) N: a set of distinct wireless cells.
(2) d_i $(1 \leqslant i \leqslant N)$: number of channels required for cell i.
(3) $|N| \times |N|$ compatibility matrix $C_{ij} (1 \leqslant i, j \leqslant N)$: minimum required frequency (channel) separation between a call in cell i and cell j.
(4) Frequency (channel) f_{ik} $(1 \leqslant i \leqslant N, 1 \leqslant k \leqslant d_i)$, is assigned to a_{ik}, the kth call in the ith cell, which is an index. Each frequency (channel) is represented by a positive integer.
(5) Set of frequency-separation constraints defined by the compatibility matrix

$$|f_{ik} - f_{jl}| \geqslant C_{ij} \quad \text{for all } i, j, k, l \ (i \neq j \text{ and } k \neq l) \ 1 \leqslant i, j \leqslant N, \ 1 \leqslant k \leqslant d_i, \ 1 \leqslant l \leqslant d_j.$$

There are a number of versions of channel assignment problem—we focus on the minimum span frequency assignment problem (MS-FAP) (see [2] for classification). In this problem, the total bandwidth required by the system is to be minimized

$$\text{Min} \left[Z = \underset{i,k}{\text{Max}} \ f_{ik} \right] \quad where \ 1 \leqslant i \leqslant N, \ 1 \leqslant k \leqslant d_i.$$

The objective of this paper is to develop a hybrid genetic algorithm and local search method (termed a memetic algorithm) for the MS-FAP. To solve it, we actually minimize blocked calls, b, for a given value of Z, maximum allowable channels. Blocked calls are those that require channels outside the allowable range considering both the $\mathbf{C}$ matrix and the maximum allowable Z. Therefore, a blocked call is one for which there is no available channel to be assigned without violating the compatibility matrix. As an example consider Fig. 1. At least 13 channels are required for no blocked calls based on the following $\mathbf{C}$ when using the first ordering $(a_{11}, a_{21}, a_{31}, a_{41}, a_{42}, a_{43})$ channel assignments in Fig. 1.

$$\mathbf{C} = \begin{bmatrix} 5 & 4 & 0 & 0 \\ 4 & 5 & 0 & 1 \\ 0 & 0 & 5 & 2 \\ 0 & 1 & 2 & 5 \end{bmatrix}.$$

Only 11 channels are needed for no blocked calls when using the second ordering $(a_{41}, a_{42}, a_{43}, a_{21}, a_{31}, a_{11})$ of channel assignments in Fig. 1. Using 11 channels with the first ordering results in one blocked call (a_{43}). Thus, the second ordering is a better solution than the first.

Z is set to a lower bound value (from [3,4] for example) for an optimization run, then iteratively increased until a solution is found where $b = 0$ and Z is at its lowest value. A new encoding of the problem, which is much more compact than any used previously, is devised. Computational results using problems and results from the literature show that the proposed method is effective, reliable and efficient.
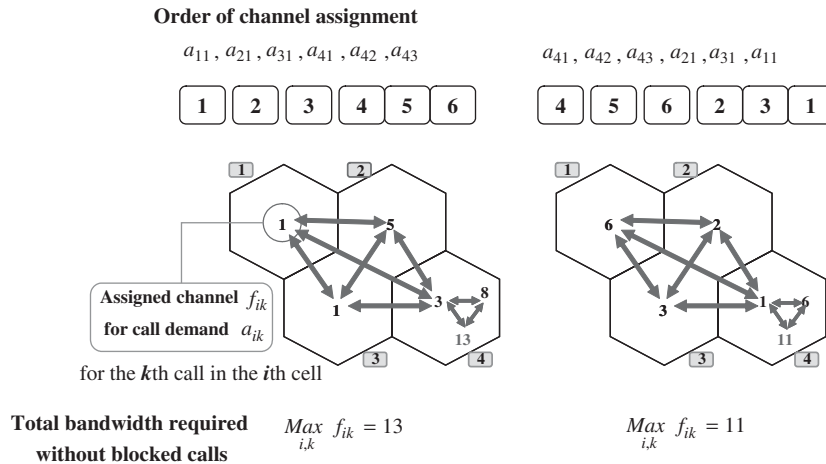
Fig. 1. Encoding and evaluation method of a solution.

While we show our method on problems where b must equal 0, the method could also be used (without alteration) to minimize b for a given Z, that is, minimize blocked calls for a set of available frequencies.

2. Previous approaches

Both [1,5] provide comprehensive overviews of the frequency assignment problem. There have been previous papers addressing the MS-FAP [e.g., 3,4,6–13] including those that minimize the maximum frequency assigned.

Channel assignment problem 3 (CAP3) [1] is hybrid deterministic/stochastic neighborhood local search algorithm for channel assignment and is very efficient in finding a local optimum. On the other hand, genetic algorithms (GAs), ant algorithms, tabu search, and simulated annealing are global optimization metaheuristics. In many applications, GAs effectively locate the neighborhood of the global optimum, but have problems converging to the optimum itself, in other words, they are not particularly efficient local search mechanisms. Song and Irving [14] have suggested that hybridization is a way to develop more powerful algorithms. Radcliffe [15] first formally defined a memetic algorithm as one that integrates local search as part of the reproductive mechanism in GAs. The term memetic algorithm comes from the notion of a meme, a unit that can be genetically modified by thought or experience. This is different from a gene, which is unaltered by experience. Using a newly developed compact encoding, we integrate a GA for diversified search and the CAP3 local search—a memetic algorithm approach—for the MS-FAP in FDMA wireless systems.

3. The memetic algorithm for channel assignment

The scheme of the memetic algorithm for channel assignment is shown in Fig. 2. It is composed of three modules, GA, CAP3 [1], and the no interference channel assignment (NICA) algorithm. The GA module
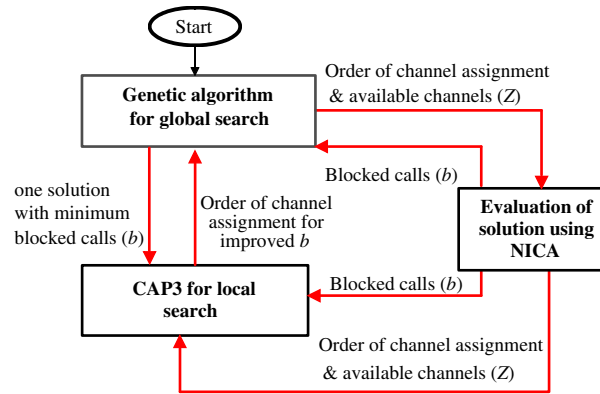
Fig. 2. Scheme of memetic algorithm for the MS-FAP.

is used to find promising regions of the search space. The CAP3 module is used to find locally optimal solutions within the regions of the GA solutions. The NICA module is an adaptation of the well-known frequency exhaustive assignment (FEA) heuristic [16] that uses a cell ordering rather than a call ordering. NICA is used for evaluating solutions by assigning frequencies (channels) given a cell ordering.

The memetic algorithm starts from the GA module. The available number of channels (Z) and the GA parameters are set, then the initial population is generated. The GA (or the CAP3 module) sends the available channels (Z) and the order of cells for channel assignment to NICA for frequency assignment. This produces the channels used (up to Z) or the number of blocked calls, b (if the number of channels required $> Z$). The memetic algorithm terminates if the blocked call value, b, is equal to 0 or if the pre-specified computation time is reached. If there are no significant improvements in the difference between blocked calls, b, within a pre-specified number of generations in the GA (or the CAP3 module) then the other module is invoked. We use 100 generations of GA to trigger the CAP3 module (a single solution is selected to be sent to CAP3 using the standard GA roulette wheel approach which probabilistically favors better solutions) and 2000 iterations of CAP3 to trigger the GA.

3.1. The NICA module

We developed NICA based on the FEA proposed by Sivarajan et al. [16]. NICA assigns the lowest possible frequency to each call which is consistent with previous assignments based on the specified order of cells. In step 1 of NICA, assign channel 1 to the first cell in the ordering. In step 2 of NICA, assign to the next cell in the ordering the lowest possible channel consistent with the previous assignment without violating the constraints using the compatibility matrix. Continue this procedure until channels are assigned for all call demands. NICA is a heuristic that produces a frequency assignment for any cell ordering, and like FEA, usually produces optimal or near optimal assignments.

Fig. 3 shows an example of channel assignment for a encoding using NICA with **C** as

$$\mathbf{C} = \begin{bmatrix} 5 & 2 & 0 \\ 2 & 5 & 2 \\ 0 & 2 & 5 \end{bmatrix}.$$
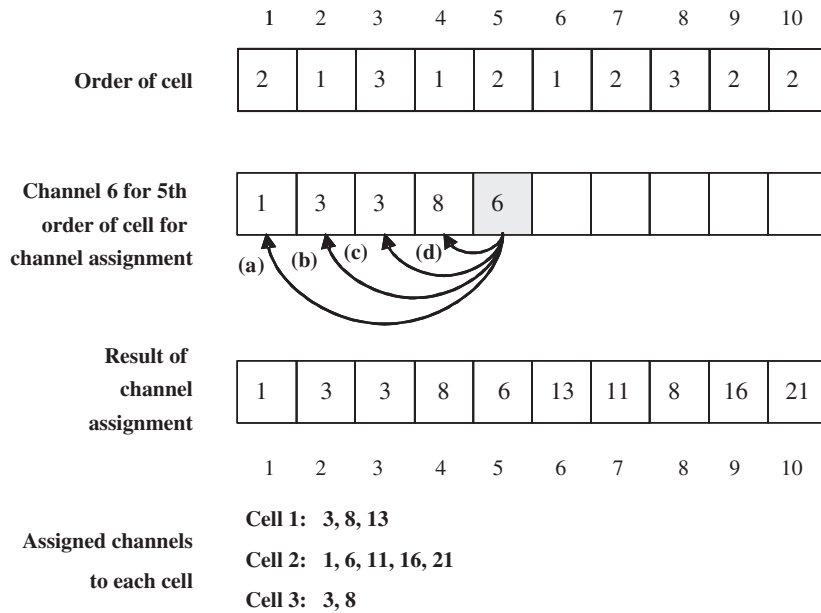
Fig. 3. An example of channel assignment for a solution using NICA.

There are call demands 3, 5, and 2 in cells 1, 2, and 3. Channels 1, 3, 3, and 8 are already assigned by NICA based on the order of each cell (shown in the encoding). NICA now decides which channel to assign to the 5th gene (portion of the encoding) of the encoding (2nd call demand in cell 2). The channel of the 5th gene must be greater than or equal to 6 (5+1). This is because there must be at least 5 channel differences from channel 1 of the 1st gene ((a) in Fig. 3), which is already assigned to the same cell. Also, the channel of the 5th gene must be greater than or equal to 5 (3+2). This is because there must be at least two channel differences from channel 3 of the 2nd gene ((b) in Fig. 3), which is already assigned to cell 1. In this way we choose channel 6 for the 5th gene of the encoding (ordering of cells) based on the (a), (b), (c), and (d) constraints. Continuing in this manner, NICA assigns channels 3, 8, and 13 to cell 1, channels 1, 6, 11, 16, and 21 to cell 2, and channels 3 and 8 to cell 3.

3.2. The genetic algorithm module

3.2.1. Representation of a solution

A representation of the solution (or encoding) of a GA gives the order of cells for assigning channels. Fig. 4(a) shows an encoding from the literature [8,16]. The first representation of the call list shows that the 2nd, 4th, and 6th channels are assigned to cell 1, the 1st, 7th, 10th, 9th, and 5th channels are assigned to cell 2, and the 8th and 3rd channels are assigned to cell 3. Different sequences (2, 4, 6), (4, 2, 6), (6, 4, 2), (2, 6, 4), (6, 2, 4), and (4, 6, 2) in cell 1 have the same channel assignment since the sequence of assignment within a cell is unimportant. For this specific example, there are 1440 (3! × 5! × 2!) combinations of solutions. We developed a different encoding as shown in Fig. 4(b). As before, the 2nd, 4th, and 6th channels are assigned to cell 1. The sequence within a cell, since it is unimportant, is not
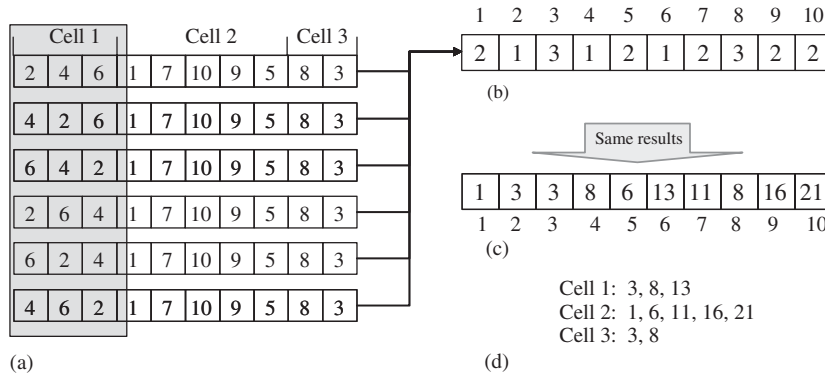
Fig. 4. New compact encoding: (a) representation of call list from the previous literature; (b) our representation of order of cell; (c) list of assigned channels using NICA; and (d) assigned channels to each cell.
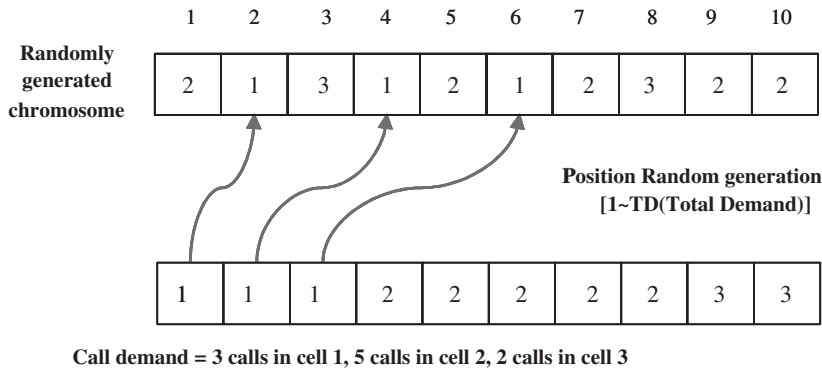

Fig. 5. An example of a randomly generated solution.

encoded. This is a much more efficient encoding that loses no information for the solution. The list of assigned channels and assigned channels to each cell are shown in Fig. 4(c) and (d).

There are 3, 5, and 2 call demands in cells 1, 2, and 3 in Fig. 5. This is an example of a randomly generated solution. The GA module generates a solution that satisfies the required number of channels needed by each cell to fulfill traffic demand requirements. The length of all representation of solutions is fixed and identical.

3.2.2. Evaluation of a solution

NICA assigns the frequencies/channels to the cell ordering using the call demand and the compatibility matrix. If the number of channels used exceeds Z, then $b > 0$. Fig. 6 shows an example solution evaluation. The number of channels used is 21. If $Z \geqslant 21$, then the blocked call value is 0 as shown in Fig. 6(a). If there are 12 available channels, then the blocked call value is equal to 3. There is one blocked call in cell 1 and two blocked calls in cell 2 as shown in Fig. 6(b). The evaluation value is b, where lower is better.
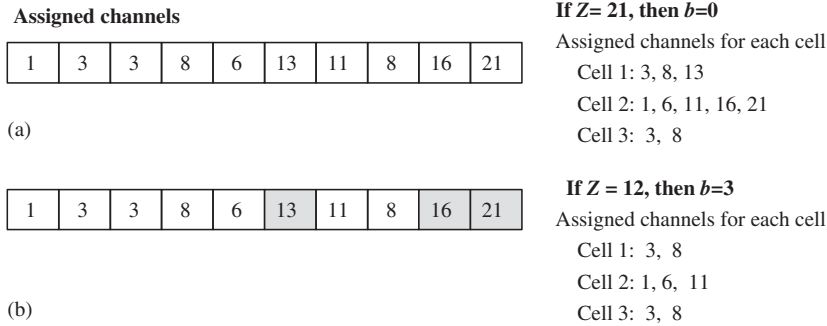
Assigned channels

1	3	3	8	6	13	11	8	16	21

(a)

If Z= 21, then b=0

Assigned channels for each cell

   Cell 1: 3, 8, 13

   Cell 2: 1, 6, 11, 16, 21

   Cell 3: 3, 8

1	3	3	8	6	13	11	8	16	21

(b)

If Z = 12, then b=3

Assigned channels for each cell

   Cell 1: 3, 8

   Cell 2: 1, 6, 11

   Cell 3: 3, 8

Fig. 6. An example of solution evaluation.

Parent 1 | 2 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 2 | 2 |

Binary Mask | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

Parent 2 | 1 | 2 | 2 | 3 | 1 | 2 | 1 | 3 | 2 | 2 |

Children 1 & 2 after order based crossover

Child 1 | 1 | 1 | 3 | 2 | 2 | 2 | 1 | 3 | 2 | 2 |

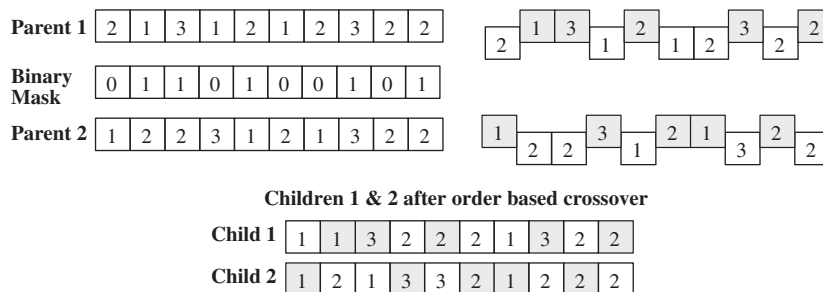**Child 2** | 1 | 2 | 1 | 3 | 3 | 2 | 1 | 2 | 2 | 2 |

Fig. 7. Order-based crossover that creates two children solutions from two parent solutions.

3.2.3. Crossover and mutation

The solutions are selected for order-based crossover using the crossover rate, which varies over evolution. If the number of selected solutions is even, then we can pair them easily. If the number of selected solutions is odd, then we can either add one extra solution or remove one selected solution. This choice is made randomly. Once selected, two solutions are mated as shown in Fig. 7. We randomly generate a binary number mask where the length equals that of the solution. Child 1 keeps genes 1, 3, 2, 3, and 2 of parent 1 (these have a binary digit of 1 at the 2nd, 3rd, 5th, 8th, and 10th position). Child 2 keeps genes 1, 3, 2, 1, and 2 of parent 2 (these have a binary digit of 0 at the 1st, 4th, 6th, 7th, and 9th positions). Child 1 rearranges the remaining genes (2, 1, 1, 2, and 2) of parent 1 (these have a binary digit of 0) based on the order of the cell number in parent 2. Child 2 rearranges the remaining genes (2, 2, 1, 3, and 2) of parent 2 (these have a binary digit of 1) based on the order of the cell number in parent 1.

Children are then mutated. Fig. 8 shows the mutation procedure. If a random number from 0 to 1 for each gene is less than the mutation rate, then the gene is selected for mutation. The selected genes (1, 4, 7, and 8) are randomly rearranged during mutation.

Once the mutated children are generated, they are pooled with the existing population, and all are ranked according to fitness (that is, by b, with lower being better). A set of the best solutions (termed the elitist set) of size F_E (see Eq. (1)) is retained for the next generation. The remaining solutions are selected for retention to the next generation using the standard roulette wheel approach, which is probabilistic, favoring better solutions. To move from a more random, more diverse search in early generations to a more
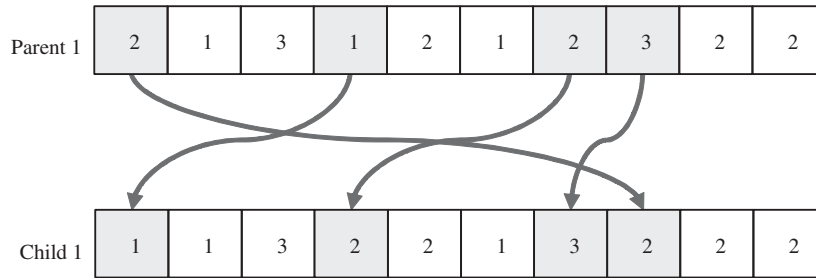
Fig. 8. Mutation of a solution.

focused search in later generations we adapt F_E, the crossover rate and the mutation rate over evolution. F_E increases with generation while the mutation rate and the crossover rate decrease, all according to the following equations:

$$F_E = E - INTEGER(E \times \alpha^g), \tag{1}$$

$$F_m = m \times \beta^g, \tag{2}$$

$$F_c = c \times \beta^g, \tag{3}$$

where F_E is the number of solutions used for the elitist pool in generation g, E the maximum number of solutions used for the elitist pool, F_m the mutation rate in generation g, m the mutation rate in generation 0, F_c the crossover rate in generation g, c the crossover rate in generation 0, α, β the weighting factors, $0 < \alpha < 1$, $0 < \beta < 1$ and g the generation 0, 1, 2, 3, ..., G.

3.3. The CAP3 local search module

There were several precursors to CAP3: CAP1 randomly generates neighbors (probabilistic) and CAP2 generates neighbors systematically in sequential order (deterministic) [1]. CAP3 combines both probabilistic and deterministic aspects [1]. CAP3 exchanges the gene with the highest channel number with a randomly selected gene, as shown in Fig. 9. (If there are multiple genes with the highest channel number, one is chosen randomly.) We take an exchanged solution to be the new solution if its evaluation value is better than or equal to that of the current best solution. This continues until no more improvement is realized.

4. Experiments and analysis

The memetic algorithm was coded in JAVA and a Pentium III 933 MHz CPU was used for eight benchmarking problems in a 21 cell wireless FDMA system (Fig. 10), the so called Philadelphia problem (see [2] for complete description). Table 1 describes the two sets of call demands. Table 2 shows eight problems that have been used in many papers [1,8,16–19]. We use a population size of 50 for all problems. Other GA parameters are set as follows (the parenthesized values for each are those considered during preliminary experimentation): $\alpha = \beta = 0.999$ (0.99–0.999); $E = 10$ (10–15) which moves the number
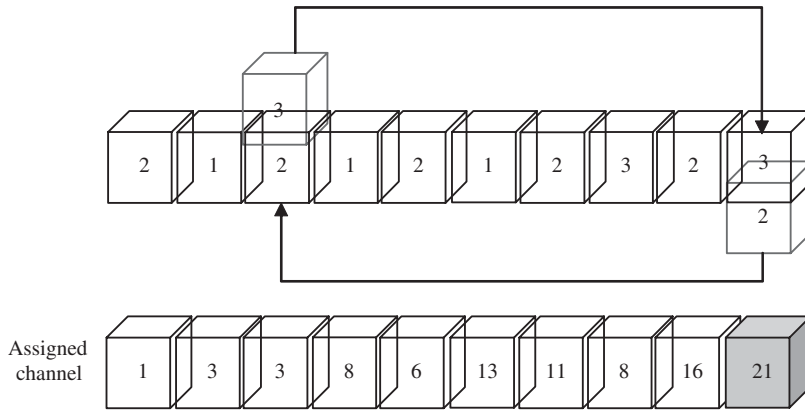
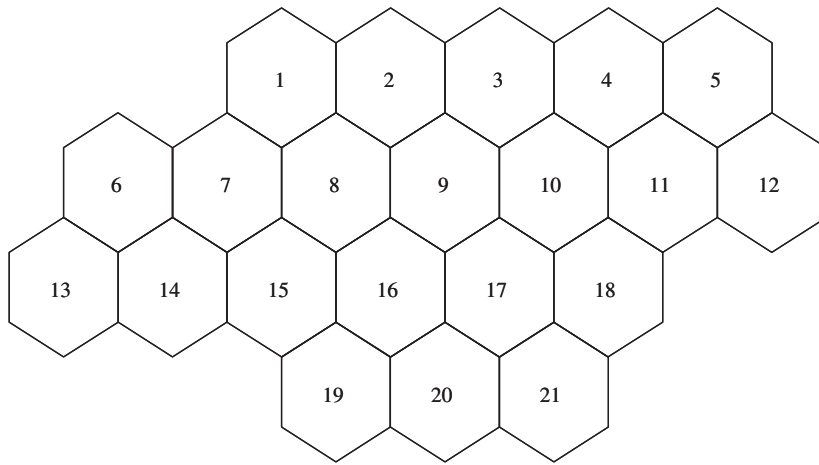Fig. 9. CAP3 [13] local search algorithm using our encoding (top).

Fig. 10. The 21 cell wireless FDMA "Philadelphia" system [2].

Table 1
Call demands *D1* and *D2* for the 21 cell "Philadelphia" system [2]

Cell no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Total demand
D1	8	25	8	8	8	15	18	52	77	28	13	15	31	15	36	57	28	8	10	13	8	481
D2	5	5	5	8	12	25	30	25	30	40	40	45	20	30	25	15	15	30	20	20	25	470

of elitist solutions from 0 early on to 10 ($\frac{1}{5}$ of the population) later on; $G = 100$, $m = 0.2$ (0.1–0.5) and $c = 0.3$ (0.1–0.5).

Table 2
Problems with different constraints, call demands, and allowable Z [2]

Problem number	1	2	3	4	5	6	7	8
Adjacent channel constraint	1	2	1	2	1	2	1	2
Co-site constraint	5	5	7	7	5	5	7	7
Demand	D1	D1	D1	D1	D2	D2	D2	D2
Allowable Z	381	427	533	533	221	253	309	309

Table 3
Performance comparison of channel assignment methods from the literature with our memetic algorithm

Problems	1	2	3	4	5	6	7	8
Lower bound [8,19]	381	427	533	533	221	253	309	309
This paper	381	427	533	533	221	253	309	309
(2003)[19]	381	427	533	533	221	253	309	309
(2001)[17]	381	427	533	533	221	254	309	309
(2001)[18]	381	463	533	533	221	273	309	309
(1999)[8]	381	427	533	533	221	253	309	309
(1996)[1]	381	433	533	533	221	263	309	309
(1989)[16]	381	447	533	533	–	270	–	310

4.1. Comparisons with the literature

The results of our memetic algorithm and approaches in the literature on the eight test problems are given in Table 3. Most of the algorithms cited in Table 3 can find assignments ($b = 0$) for only the six easier problems (1, 3, 4, 5, 7, and 8). It is much more difficult to find optimal solutions for problems 2 and 6 (with 427 and 253 channels, respectively). Only we, Ghosh et al. [19] and Beckmann and Killat [8] found optimal solutions ($b = 0$) for these problems. Battiti et al. [17] found the optimal solution with 247 channels for problem 2, but found a solution for problem 6 with 254, instead of 253, channels.

Ghosh et al. [19] gave computation times for the six easier problems. They mentioned that their computation times for the optimal assignments of these problems varied between 12 and 80 h for the different runs on a DEC Alpha station. Beckmann and Killat [8] achieved 21% and 2% optimal rates in 100 simulation experiments for problems 2 and 6, respectively. They mentioned that these two difficult problems required average computing times of about 8 and 10 CPU minutes, respectively, using a HP Apollo 9000/700 workstation. We find optimal solutions with 70% and 48% rates, respectively, in 100 simulation experiments within several CPU minutes depending on the different parameter settings, as shown in Table 4. The new compact encoding and the memetic algorithm combine to be both more efficient and more reliably effective for difficult channel assignment problems than previous approaches reported in the literature.

Table 4
Reliability of the memetic algorithm

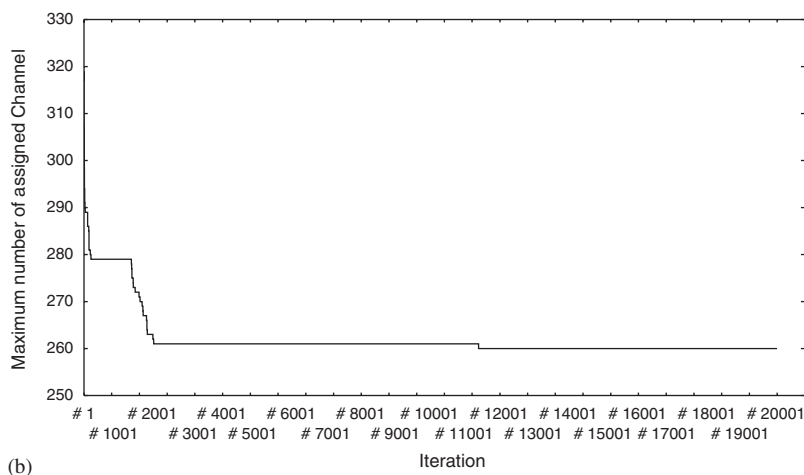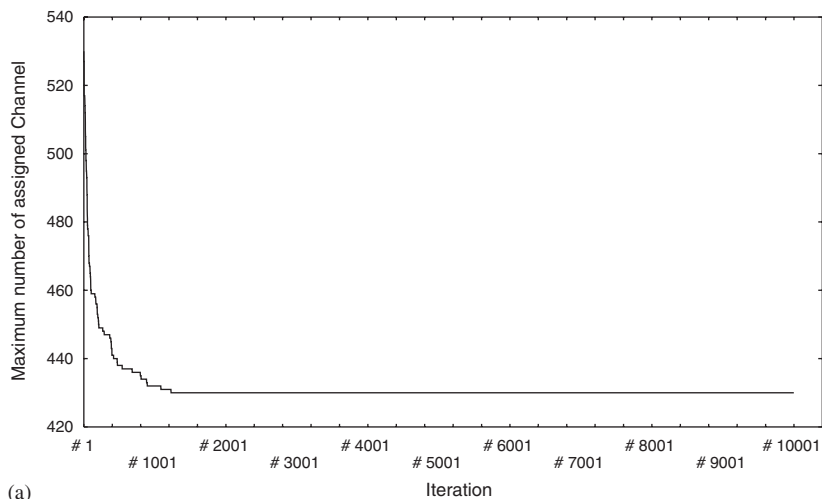| Problem	1	2	3	4	5	6	7	8
Times optimal solution found	100/100	70/100	100/100	100/100	100/100	48/100	100/100	100/100

Fig. 11. Actuals of the channels used for problems 2 and 6 using only CAP3 local search: (a) problem 2; and (b) problem 6.

4.2. Analysis of performance of the memetic algorithm on the difficult problems

Problems 2 and 6 are difficult and need further analysis. One question is whether the GA or CAP3 is effective alone, or if there a synergism between the GA and the local search (as there should be in a proper memetic algorithm). Shown in Fig. 11(a) and (b) are the number of assigned channels used per
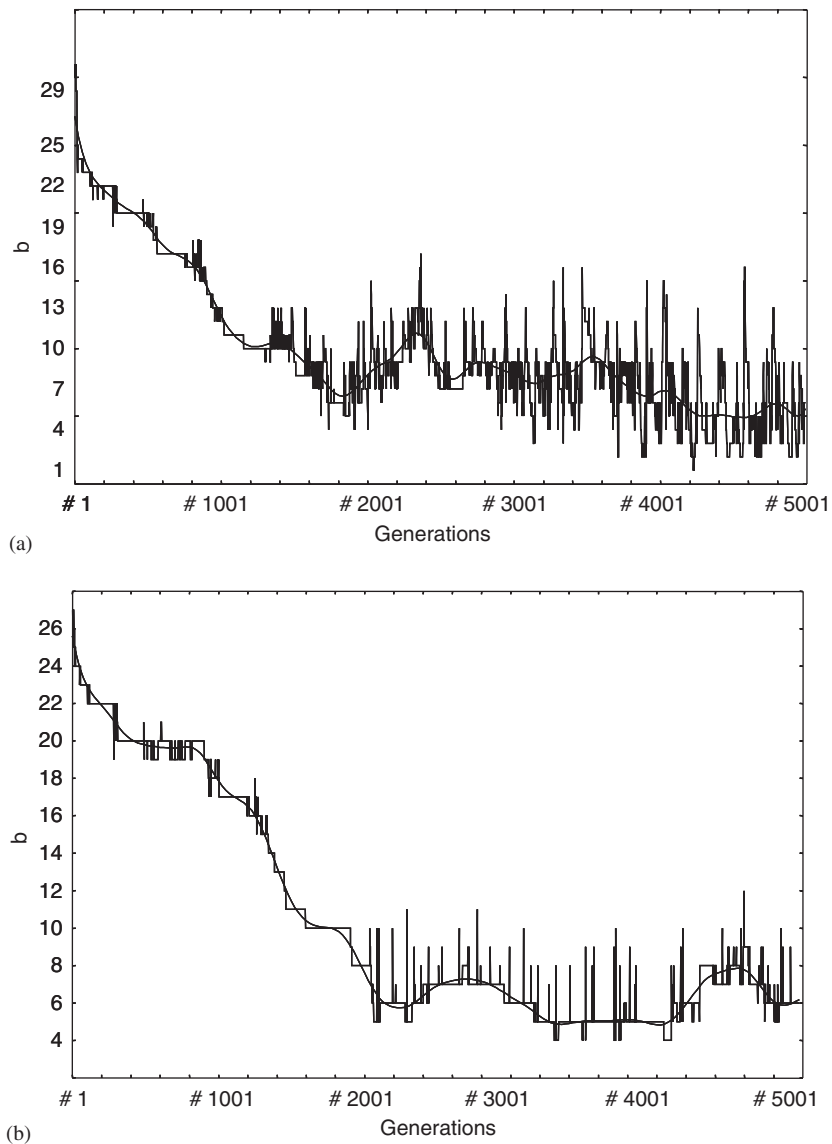
Fig. 12. Actuals and trend (smoothed line) of the blocked call value, b, for (a) problems 2 and (b) 6 using only the GA: (a) problem 2; and (b) problem 6.

search iteration using only CAP3. CAP3 converged quickly, as expected from a local search algorithm, and found solutions with assigned channels of 430 for problem 2 and 260 for problem 6, both higher than the minimum Z found by the memetic algorithm of 427 and 253, respectively.

Fig. 12 shows the trend of the blocked call value, b, for problems 2 and 6 using only a GA. The GA is relatively slow to converge compared to CAP3, and while it improves upon CAP3 in solution quality, it does not find the optimal solution where $b = 0$ to either problem.
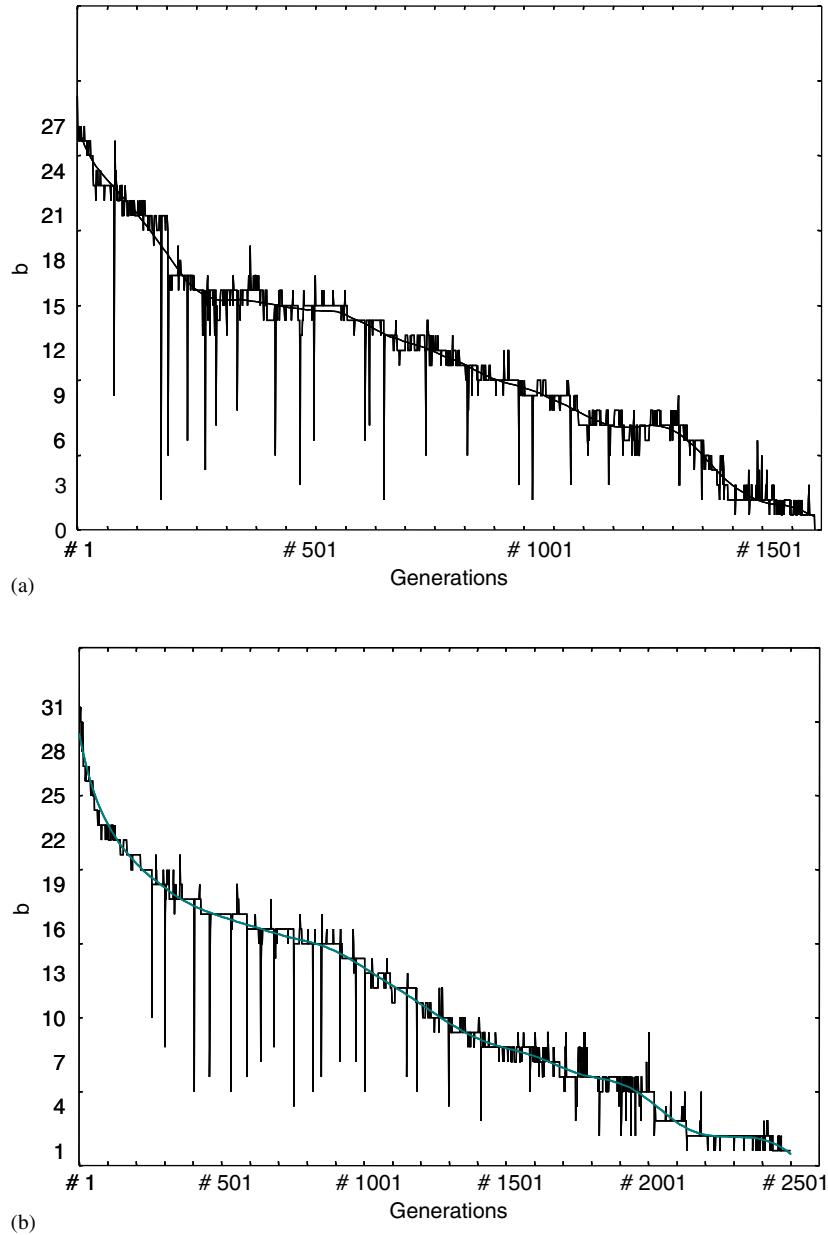
Fig. 13. Actuals and trend (smoothed line) of blocked call value, b, for problems (a) 2 and (b) 6 using the memetic algorithm: (a) problem 2; and (b) problem 6.

Fig. 13 shows the trend of the blocked call value, b, for problems 2 and 6 using the memetic algorithm. We find an optimal channel assignment where the blocked call value $b = 0$. The solutions that came from the CAP3 module are sporadically improved upon through the GA mechanisms. One must be careful in comparing iterations (x-axis) directly for these three optimization methods (CAP3, GA, and memetic

algorithm). For each iteration, there is increased computational effort of the GA over CAP3 and of the memetic algorithm over the GA. However, all are within a reasonable computational time frame (seconds to minutes per problem) and exact computational effort is not a primary issue in design optimization, which is done infrequently and offline.

5. Conclusions

This paper has addressed a difficult combinatorial optimization problem that has growing importance in wireless communications systems. Using a new problem representation that more compactly encodes the salient features of channel assignment allows for more efficient and reliable optimization. This is because the encoding greatly reduces the search space of possible solutions.

Although the new encoding could be used with a variety of optimization approaches, we chose to integrate a GA with a known local search method. GAs are powerful global searchers but are inefficient local searchers. Local search methods, like the CAP3 one used herein, are fast but can only identify locally optimum assignments. Because suboptimal assignments can results in either blocked calls or in higher system costs (by specifying more channels than absolutely necessary), finding the global optimum to these problems is important. The memetic algorithm is effective (finding optimal solutions) and computationally reasonable. It outperforms the six methods from the literature to which we compare on an established set of test problems.

References

[1] Wang W, Rushforth CK. An adaptive local-search algorithm for the channel-assignment problem (CAP). IEEE Transactions on Vehicular Technology 1996;45(3):459–66.

[2] ⟨http://fap.zib.de/⟩.

[3] Gamst A. Some lower bounds for a class of frequency assignment problems. IEEE Transactions on Vehicular Technology 1986;35:8–14.

[4] Sung CW, Wong WS. Sequential Packing algorithm for channel assignment under cochannel and adjacent channel interference constraint. IEEE Transactions on Vehicular Technology 1997;46:676–85.

[5] Aardal KI, van Hoesel CPM, Koster AMCA, Mannino C, Sassano A. Models and solution techniques for frequency assignment problems. 4OR 2003;1:261–317.

[6] Allen SM, Hurley S, Smith DH, Thiel SU. Using lower bounds in minimum span frequency assignment. Meta-heuristics: advances and trends in local search paradigms for optimization. Dordrecht: Kluwer; 1999. p. 191–204.

[7] Avenali A, Mannino C, Sassano A. Minimizing the span of d-walks to compute optimum frequency assignments. Mathematical Programming 2002;91(2):357–74.

[8] Beckmann D, Killat U. A new strategy for the application of genetic algorithms to the channel-assignment problem. IEEE Transactions on Vehicular Technology 1999;48(4):1261–9.

[9] Hao J-K, Dorne R, Galinier P. Tabu search for frequency assignment in mobile radio networks. Journal of Heuristics 1998;4:47–62.

[10] Hurley S, Smith DH, Thiel SU. FASoft: a system for discrete channel frequency assignment. Radio Science 1997;32: 1921–39.

[11] Janssen J, Kilakos K. An optimal solution to the "Philadelphia" channel assignment problem. IEEE Transactions on Vehicular Technology 1999;48(3):1012–4.

[12] Janssen J, Kilakos K, Marcotte O. Fixed preference channel assignment for cellular telephone systems. IEEE Transactions on Vehicular Technology 1999;48:533–41.

[13] Tcha D, Chung Y, Choi T. A new lower bound for the frequency assignment problem. IEEE/ACM Transactions on Networking 1997;5:34–9.

[14] Song YH, Irving MR. Optimization techniques for electrical power systems: part 2 heuristic optimization methods. IEE Power Engineering Journal 2001;15(3):151–60.

[15] Radcliffe NJ. Formal memetic algorithms. In: Fogarty T, editor. Evolutionary computing. Springer lecture notes in computer science, vol. 865. Berlin: Springer; 1994. p. 250–263.

[16] Sivarajan KN, McEliece RJ, Ketchum JW. Channel assignment in cellular radio. Proceedings of the IEEE vehicular technology conference, 1989. p. 846–50.

[17] Battiti R, Bertossi A, Cavallaro D. A randomized saturation degree heuristic for channel assignment in cellular radio networks. IEEE Transactions on Vehicular Technology 2001;50(2):364–74.

[18] Chakraborty G. An efficient heuristic algorithm for channel assignment problem in cellular radio networks. IEEE Transactions on Vehicular Technology 2001;50(6):1528–39.

[19] Ghosh SC, Sinha BP, Das N. Channel assignment using genetic algorithm based on geometric symmetry. IEEE Transactions on Vehicular Technology 2003;52(4):860–75.