Solving Channel Allocation Problem in Cellular Radio Networks Using Genetic Algorithm

Srinivas Pinagapany

Pune, India
psrinivas_11@yahoo.co.in

A V Kulkarni
Dept. of Electronics and Telecommunications,
D.Y.Patil College of Engineering,
Pune, India
anju_k64@yahoo.co.in

***Abstract*—With limited frequency spectrum and an increasing demand for cellular communication services, the problem of channel allocation becomes increasingly important. However, finding a conflict-free channel allocation with minimum interference and minimum channel span is NP hard. Genetic algorithm (GA) is one of the heuristic optimization tools that can be used to solve this problem efficiently. In this paper we investigate fixed and dynamic channel allocation problem for cellular networks and its optimum solution using GA.**

Keywords—Genetic algorithms, fixed channel allocation, dynamic channel allocation, adjacent channel interference, co-channel interference.

I. INTRODUCTION

As demand grows for wireless communication systems, the problem of frequency assignment becomes increasingly important. The available bandwidth for cellular communication is limited and the number of subscribers is increasing day by day, therefore it became necessary to find an optimal frequency assignment scheme that uses frequencies as efficiently as possible. This led to the development of cellular networks using space division multiplexing (SDM).

In SDM cellular networks, the service area or circle of operation is divided into a number of hexagonal cells. By using appropriate reuse distance criteria, large geographical area coverage with minimum interference with limited bandwidth is possible. This problem is generally referred as channel allocation problem (CAP) and has been widely investigated in [2]-[5], [7]-[15].

In CAP, each cellular network is considered to have n arbitrary cells. Without loss of generality, it is assumed that channels are evenly spaced in the radio frequency spectrum and the channels are mapped to consequent positive numbers. The separation between these channel numbers would reflect the separation in the frequencies in order to satisfy the interference constraints. While assigning channels to users in various cells, at least 3 constraints need to be satisfied namely:

- Co–site interference constraint (CSC)
- Adjacent channel interference constraint (ACC)
- Co–channel interference constraint (CCC)

The solution to the problem can be generally classified in to two types: 1) Fixed channel allocation (FCA), where channels are permanently allocated to each cell and 2) Dynamic channel allocation (DCA), where all channels are available for the entire network, and are allocated dynamically depending on the traffic requirement of each cell.

In either case the problem of allocating channels to users in all the cells satisfying the above three constraints is considered to be NP-complete [1], signifying that as the size of the problem grows, the calculation time grows not in a polynomial way but rather in an exponential way. Various heuristic methods such as GA [2], Artificial Neural-Networks [3], Simulated Annealing [4], and Tabu Search [5] have been used. In our approach we have used a modified GA to solve CAP with both FCA and DCA. The efficiency of our approach is demonstrated through simulation and application to a benchmark problem.

In section II we give a brief introduction on simple GA with appropriate modifications to solve CAP. In section III we address FCA and later apply it to DCA in section IV. In section V we perform some simulations and compare the results. Finally we conclude in section VI.

II. GA FOR SOLVING CAP

In our paper we suggest a modified GA to solve the channel allocation problem because the implicit parallelism coupled with an ability to effectively exploit accumulating information about the search space offers a significant advantage over many other approaches. According to Goldberg [6] ("How are GA different from traditional methods?"), GA is different from normal optimization and search methods in many ways. For a detailed study about GA refer [6].

The simple GA mentioned in [6], can have many variants. Each one derived by changing one or more of its operators to improve the convergence of the Objective function (also called the Cost). For instance the selection scheme could be a proportionate selection method (roulette wheel selection), rank based selection, tournament selection etc. We have used a variant of proportionate scheme called the *sigma truncation* method of selection. For crossover and mutation, there could be single point or multipoint crossover operations. Each channel allocation scheme is called an *Individual*. The individuals could be binary encoded or decimal (real value) encoded. The results of simulation for these variants are studied.

III. FCA USING GA

To start with, we consider a 9-Cell network with all three interference constraints and later extend it to a 21-Cell network. The traffic demand requirement for each cell is denoted by the vector *cell[i]*. The elements of this vector represent the number of users in Cell-*i*. The traffic demand for all generations and runs is assumed fixed in the case of FCA. The channel span in the network is referred to as BW. Various combinations of BW and total number of users in the network, and their impact on Objective function have been studied.

For each N-Cell network there is an N x N compatibility matrix C_N that would specify the ideally required spacing among channels in the cells. There would also be an allocation matrix A for each individual. The extent by which matrix A differs from matrix C_N specifies the value of Objective function. The population size on which GA is applied is referred as *pop* and is chosen after a number of trial and errors between 20 and 40. Large population (100) usually renders the selection inefficient and very small population (5 or 10) does not explore the solution space much.

The channels allocated to users in each cell are represented in a binary form. Consider an individual of 4 cells where a maximum of 4 users per cell is permitted and the BW is also 4. An example individual would look like this:

Channels allocated	Encoded individuals
CELL 1: 0	1 0 0 0
CELL 2: 1 2 3	0 1 1 1
CELL 3: 0 3	1 0 0 1
CELL 4: 0	1 0 0 0

Encoded string for such an individual would be, 1000011110011000; where 1 indicates channel assigned while a 0 indicates absence of channel. After encoding, using genetic operators like crossover and mutation, the algorithm creates the subsequent generation from the individuals of the current generation. This generation cycle is repeated until some pre-defined termination criterion like number of generations or satisfactory value of Objective function is reached. A variant of SGA with decimal encoding was also simulated and the results are compared in section V.

The compatibility matrix for a 9-Cell network with square shaped cells is shown in Fig. 1. CSC = 4, ACC = 3, CCC = 1, channel distance between users in diagonal cells = 2 (only in case of 9-Cell, square shaped).

	1	2	3	4	5	6	7	8	9
1	4	3	1	3	2	1	1	1	1
2	3	4	3	2	3	2	1	1	1
3	1	3	4	1	2	3	1	1	1
4	3	2	1	4	3	1	3	2	1
5	2	3	2	3	4	3	2	3	2
6	1	2	3	1	3	4	1	2	3
7	1	1	1	3	2	1	4	3	1
8	1	1	1	2	3	2	3	4	3
9	1	1	1	1	2	3	1	3	4

Figure 1. Compatibility Matrix for 9-Cell Network.

The allocation matrix for a typical allocation scheme is shown in Fig. 2.

4	2	3	1	3	1	0	1	2
2	4	1	3	5	1	2	1	4
3	1	4	1	1	1	1	2	2
1	3	1	4	2	2	5	2	1
3	5	1	2	4	0	7	4	1
1	1	1	2	0	4	1	0	1
0	2	1	5	7	1	4	1	6
1	1	2	2	4	0	1	4	3
2	4	2	1	1	1	6	3	4

Figure 2. Allocation Matrix for a typical allocation scheme.

We subtract the two matrices to get the difference matrix and then count the number of violations to the three constraints, which gives us the value of the Objective function of an individual that in the above case is 26. Our primary objective while solving both FCA and DCA would be to minimize this value to the best. The BW used was 30 channels i.e. channel 0 to channel 29. The demand vector is initialized with valid number of users before populating the individuals.

A. *Input parameters for FCA*

There are certain parameters we assumed for our simulation for FCA namely 'CELL' – Number of cells in the network and channel span or BW. We also assume the number of users in a cell in accordance with benchmark problems stated in [7].

B. *FCA for 21-Cell Network*

This problem also known as the Philadelphia benchmark problem [7] is having a network structure as shown in Fig. 3.

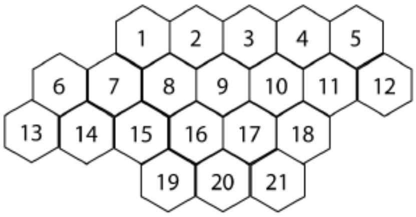

Figure 3. Philadelphia benchmark 21-Cell network structure.

The number inside each cell in Fig. 4 specifies the number of users or the traffic demand in that cell.

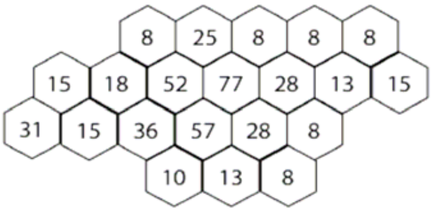

Figure 4. Philadelphia problem instance 1.

Since the cell shape is considered to be hexagonal now the compatibility matrix for this network would change accordingly and would now be a 21 x 21 matrix as given by [8]. The CSC is considered to be 5 and ACC 1. The demand vector has also changed and the total number of users in the network for FCA is 481 for problem instance 1. The upper bound on the number of users in any cell should be less than BW/CSC. Since CSC has a value of 5 and the maximum number of users in any cell is 77, the minimum BW to start the algorithm with could be 385. Further this demand is reduced to reach the most optimum BW for a specified demand in the network and the results are compared.

The following demand vectors describe two separate problem instances:

$Cell_1$ [i] = {8, 25, 8, 8, 8, 15, 18, 52, 77, 28, 13, 15, 31, 15, 36, 57, 28, 8, 10, 13, 8}

$Cell_2$ [i] = {5, 5, 5, 8, 12, 25, 30, 25, 30, 40, 40, 45, 20, 30, 25, 15, 15, 30, 20, 20, 25}

Fixed resource allocation schemes use a predetermined assignment strategy aimed at improving average case performance. However, such schemes are not able to adapt to the varying nature of user traffic. In FCA once channels are allocated to users in the cells, they are not reallocated dynamically if there is a lot of interference caused due to frequent movement of users in the network or drastic change in the traffic pattern. FCA also lacks efficient channel utilization. However, FCA is most suited for a predictable traffic pattern apart from being the easiest scheme of the two. DCA attempts to optimize system performance by adapting to the traffic variations. It completely removes the requirement of a static and structured frequency reuse pattern and makes available all radio channels available for every call.

IV. DCA USING GA

With DCA there is no fixed association of channels to cells. Each of the channels available to a cluster of cells could be used in any cell or sector within the cluster as needed. To simulate dynamic traffic conditions in the network, randomly selected users move from one cell to another, also known as *inter-cell handoff* in case of mobile cellular communications. We also allow new call originations in randomly selected cells in the network, as a result of which the demand in the respective cells changes accordingly. Whenever the arrival of call causes the demand in each cell to exceed the maximum, the call is blocked. For each such call arrival we calculate the blocking probability. Initially the blocking probability remains within the assumed threshold of 2 %, but as more and more calls originate with large holding time, the blocking increases. Once this threshold is crossed for a prescribed number of times it is an indication that we need to optimize channel allocation. Now GA is applied. Initially channel reallocation is done only for the cells in which users have moved in or new calls have originated in, so as to minimize the number of frequency reallocations. If reallocation is done for all cells then the resulting Objective function value could be worse than before. If a satisfactory Objective function value is not reached then channel allocation is done considering all users in the network.

Fig. 5 is a snapshot of a typical traffic distribution pattern in a 21-Cell network. Users have moved in to cells pointed by the arrows and the colored cells are the ones in which new calls have originated. New users also move into the network from outside which are treated as new call originations.

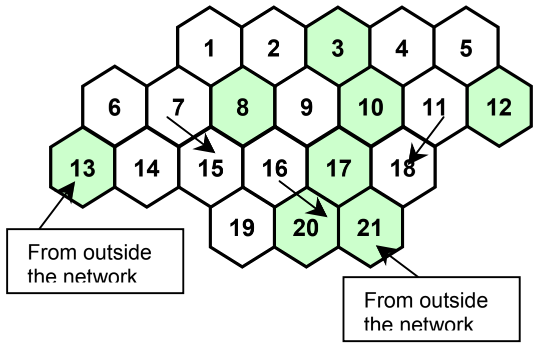

Figure 5. Typical traffic behavior.

Demand vector before movement:

cell[i] = {8, 25, 8, 8, 8, 15, 18, 52, 77, 28, 13, 15, 31, 15, 36, 57, 28, 8, 10, 13, 8}

Demand vector after the movement:

cell[i] = {8, 25, 9, 8, 8, 15, 17, 53, 77, 29, 12, 16, 33, 15, 37, 56, 29, 9, 10, 15, 10}

The mean call arrival rate was taken as 5 calls/sec and mean call holding time was assumed to be 100 seconds. Call arrival rates and holding time were exponentially distributed. Simulation results and inferences are shown in Figures 6 and 7 and Tables I-IV.

V. Simulation and Results

The modules in FCA are as follows:

Initialize: This module will randomly populate *pop* number of individuals with the number of channels dictated by the demand vector cell[i]. CSC is overcome while allocation is done initially so that the search space is reduced.

Evaluate: This module calculates the Objective function for each individual. We use another value called the *fitness* of an individual. The fitness is inversely proportional to the value of Objective function. Lower the Objective function value higher the fitness for an individual. There are several ways to calculate the fitness for every individual. The simplest method is to use the Objective function itself, but this doesn't always work very well as the values might be much closer in magnitude, so every one has an equal chance of being selected. Therefore to exaggerate the Objective function values to a more discerning value we use a statistical technique called *sigma truncation*. The routine calculates the average and the standard deviation of the Objective function and calculates a constant that is some number of standard deviations less than the average. This constant is subtracted from the Objective function for a more discerning fitness. We select a deviation of 1.5 (usually ranges between 1 and 3) standard deviations. It helps the selection module to select fitter individuals for the next generations.

Encode: This module is used to represent the channel allocation scheme for each individual into a bit string of total length CELL * BW. The scheme of representation was explained in the beginning of this section. Simulation was also done using decimal encoding or real value coded GA. In this scheme instead of representing the channel allocation scheme in binary format, for each individual, the channel numbers in each cell were concatenated together to form a decimal string. Thereafter the genetic operators were applied.

Selection: This module selects the individuals with higher fitness and makes a proportionate number of copies for the subsequent generations. Roulette wheel selection is used along with the *sigma truncation* method, so that maximum number of copies of fitter individuals is made in each generation.

Crossover: This module performs the genetic crossover operation on randomly selected pairs of individuals. In crossover a crossover site is randomly selected from the bit string and the bits after the crossover site are swapped among the individuals involved in crossover. Single point crossover is when crossover is done about one site. In Multipoint crossover two sites *m1* and *m2* are selected randomly and bits are swapped for the individuals starting from site *m1* till *m2*. The results for both techniques are compared. While performing crossover the integrity of demand vector should be preserved, i.e. the number of users in each cell should remain same after the genetic operators are applied on the individuals. For this reason a modified crossover technique called the 'genetic fix crossover' as suggested by [9] was used. For a detailed explanation on the principle of genetic fix algorithm please refer [9]. When decimal encoding was used for simulation no such issues are encountered, thereby reducing the calculation overhead. The value of probability of crossover (PCROSS) decides the number of individuals to be crossed in a generation. The results of the simulation with various values of PCROSS are compared in section V.

Mutation: This module performs random bit reversal in the individuals' bit string. As the demand vector should remain same after mutation, it is always done in pair. Mutation of individuals is also done based on the values of probability of mutation (PMUT). The results of the simulation with various values of PMUT are compared in section V.

Maximum: This module finds the best individual in a generation and writes it to a file. It also compares the best individuals in each generation and displays the final solution at the end of the program.

For simulation we had used a workstation with Windows XP OS on a Pentium 4 processor, 1.8 GHz speed and 256 MB RAM. The simulation was done using C. After many runs of simulation we observed that the Objective function converges to a minimum value in fewer generations for less intensive problems like the 9-Cell network, but takes a considerable number of generations for more difficult 21-Cell benchmark problems. Time taken for 1000 generations in each run ranges between 100-180 seconds. Time taken for each run plays a critical role in case of DCA. If a system is implemented that employs GA for DCA, it should converge to a minimum Objective function value before the network perceives any drastic change in the traffic pattern. With the advances in parallel and high speed computing such implementations would not pose any threat to the performance of the system.

TABLE I. GA using decimal encoding, Multipoint Crossover and Mutation

FCA for 9-Cell Network	**Problem instance 1**	**Problem instance 2**	**Problem instance 3**
Total users in the network	18	23	32
BW or Channel span	16	30	30
Population size	20	20	40
Maximum number of users in any cell	4	6	6
PCROSS	0.9	0.9	0.9
PMUT	0.0005	0.001	0.0009
Number of generations	1000	500	1000

TABLE II. PHILADELPHIA BENCHMARK PROBLEM

FCA for 21-Cell Network	Binary encoding		Decimal encoding	
	Problem instance 1	***Problem instance 2***	***Problem instance 1***	***Problem instance 2***
Total users in the network	481	470	481	470
BW or Channel span	385	225	385	225
Population size	40	40	40	40
Maximum number of users in any cell	77	45	77	45
PCROSS	0.9	0.9	0.9	0.9
PMUT	0.001	0.0009	0.001	0.0009
Generations	1000	1000	1000	1000

TABLE III. SUMMARY OF RESULTS FOR FCA

Problem Type	PMUT	Objective Function	Init. Value of Obj. Function	Pop Size	Generations
FCA for 9-Cell Network	0.004	16	30	20	1000
	0.001	10	34	20	1000
Problem Instance 3	0.0009	8	28	20	1000
	0.0009	9	33	40	1000
	0.0004	6	30	20	1000
FCA for 21-Cell Network	0.005	18	30	20	1000
	0.005	14	32	40	1000
Decimal encoding	0.001	14	32	40	1000
	0.0009	20	34	40	1000
	0.005	18	40	20	1000

TABLE IV. SUMMARY OF RESULTS FOR DCA

Problem Type	PMUT	Objective Function	Init. Value of Obj. Function	Pop Size	Generations
DCA 21-Cell					
Time instant 1	0.001	15	43	40	1000
Time instant 2	0.001	16	52	40	1000

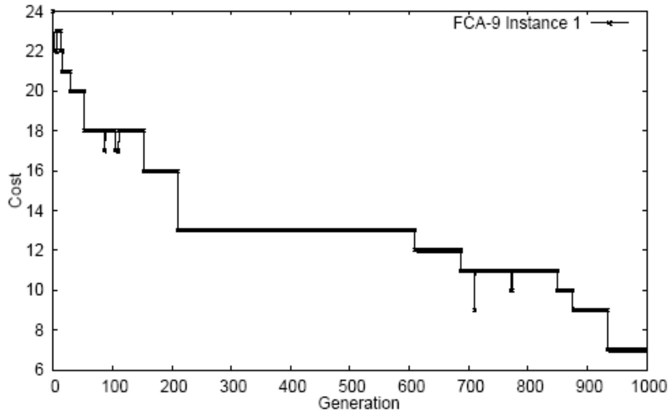

Figure 6. Plot for one of the problem instances summarized in Table I.

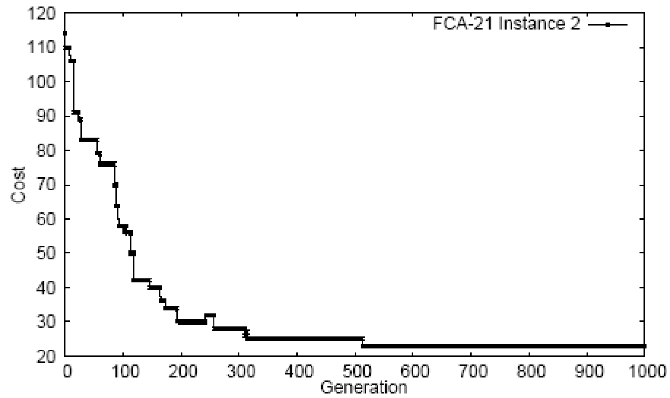

Figure 7. Plot for the problem instance 2 using decimal encoding summarized in Table II.

VI. CONCLUSIONS

- Table III summarizes the convergence behavior of GA for various PMUT, Population size, and number of generations.
- We observed that decimal encoding yields better values of objective function compared to binary encoding.
- The effect of PMUT on convergence behavior is indicated in Table III. Incase of FCA-9 cell, PMUT of 0.004 lead to premature convergence.
- It also indicates that for smaller problems large population sizes tend to slow down the convergence slightly.
- In case of FCA 21-Cell, lower PMUT still gave better Objective function values as mentioned before and large population sizes give better search results for such large problems.
- Also in case of a 21-Cell network, FCA with single-point crossover is found to give better Objective function values.
- Therefore for DCA in 21-Cell network, population size was chosen to be 40, PMUT 0.001, PCROSS 0.9, decimal encoding with single-point crossover, for each time instant.

Channel assignment was done for the whole network in both the time instances.

FUTURE RESEARCH DIRECTIONS

The application of GA to models where traffic load conditions are changed dynamically is important to examine the robustness of the method. Variants of the genetic operators could be further investigated for example, when the Objective function doesn't change for a number of generations, then increasing or decreasing the value of PMUT dynamically so as to germinate better Objective function values. The use of advanced heuristic techniques such as Multi Niche Crowding GA is worth investigating. The recent advancements in Evolutionary Algorithms (EA) has shown that combination of EA with other heuristic tools such as neural networks, and fuzzy logic is leading to more advanced optimization strategies and the application of such an approach to CAP is worth investigating.

There are several other models that could be used to test the implementation such as the benchmark problems in the COST 259 project, CALMA instances etc. In our future works we would also study the impact of various pseudo-random number generators on the performance of the GA. Moreover, the implementation of the GA based controllers for solving CAP in practical cellular environments would be discussed, particularly in systems that employ software defined radios or cognitive radios.

AKNOWLEDGMENTS

We would like to thank the three referees for their useful comments that helped us improve this paper. Our sincere thanks to Prof. G.S.Mani, Former Director & Dean, Institute of Armament Technology, Pune, whose guidance and support for this work was immensely helpful.

REFERENCES

[1] W. K. Hale, "Frequency assignment: theory and applications", *Proc. IEEE*, vol. 68, pp. 1497-1514, Dec. 1980.

[2] Lipo Wang, Sa Li, Sokwei Cindy Lay, Wen Hsin Yu, and Chunru Wan, "Genetic Algorithms for Optimal Channel Assignments in Mobile Communications", unpublished.

[3] D. Kunz, "Channel assignment for cellular radio using neural networks," *IEEE Trans. Veh. Technol.*, vol. 40, no. 1 part 2, pp. 188–193, Feb. 1991.

[4] Manuel Duque-Anton, Dietmar Kunz, and Bernhard Riiber, "Channel Assignment for Cellular Radio Using Simulated Annealing", *IEEE Trans. Veh. Technol., Vol. 42, No. 1, February 1993.*

[5] RobertoMontemanni, JimN.J.Moon and DerekH.Smith, "An Improved Tabu Search Algorithm for the Fixed-Spectrum Frequency-Assignment Problem", *IEEE Trans. Veh. Technol., Vol.52, No.3, May2003.*

[6] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[7] Karan I Aardal, Stan P.M. van Hoesel *et al.*, "Models and Solution Techniques for Frequency Assignment Problems", *Konrad-Zuse-Zentrum fÄur Informationstechnik*, Berlin, URL http://fap.zib.de.

[8] Goutam Chakraborty, "An Efficient Heuristic Algorithm for Channel Assignment Problem in Cellular Radio Networks", *IEEE Trans. Veh.Technol.*, vol. 50, no. 6, Nov. 2001.

[9] 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.

[10] K. N. Sivarajan, R. J. McEliece, and J. W. Ketchun, "Channel assignment in cellular radio," in *Proc. 39th IEEE Vehicular Technology Conf.*, May 1989, pp. 846–850.

[11] W. K. Lai and G. G. Coghill, "Channel assignment through evolutionary optimization," *IEEE Trans. Veh. Technol.*, vol. 45, no. 1, pp. 91–96, 1996.

[12] 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, no. 4, pp. 1261–1269, Jul. 1999.

[13] H. G. Sandalidis, P. Stavroulakis, and J. Rodriguez-Tellez, "An efficient evolutionary algorithm for channel resource management in cellular mobile systems," *IEEE Trans. Evol. Comput.*, vol. 2, no. 4, pp. 125–137, Nov.1998.

[14] Ken Murray and Dirk Pesch, "Adaptive Radio Resource Management for GSM using Neural Networks and Genetic Algorithms", IT & T Conference, Athlone, Ireland, Sep. 2001.

[15] Xiannong Fu, Yi Pan *et al.*, "Using a genetic algorithm approach to solve the dynamic channel-assignment problem", *Intl. Journal for Mobile Communications, Vol. 4, No. 3, 2006*.

[16] W.C.Y. Lee, Mobile Cellular Telecommunications: Analog and Digital Systems. New York: McGraw-Hill, 1995.