Genetic Search for Fixed Channel Assignment Problem with Limited Bandwidth

Eun-Jong Park, Yong-Hyuk Kim, and Byung-Ro Moon
School of Computer Science & Engineering, Seoul National University
Shillim-dong, Kwanak-gu, Seoul, 151-742 Korea
{silver,yhdfly,moon}@soar.snu.ac.kr

Abstract

We propose a hybrid genetic algorithm for the fixed channel assignment problem with limited bandwidth. A local optimization algorithm was devised to enhance the genetic algorithm's fine-tunning. In a matrix representation, the local optimization algorithm improves on a chromosome by horizontal and vertical moves of channels. The crossover is directly performed on two-dimensional matrices. Experimental results showed dramatic improvement against previous works.

1 Introduction

As the demand of mobile telecommunication sharply grows, the effective use of limited resources becomes more important. In a mobile system, the channel assignment problem is to assign channels optimally for the requests of cells. In this paper, a channel corresponds to a frequency band.

When we assign the channels to the cells, EMC (Electro-Magnetic Compatibility) constraints [14] must be satisfied. There are three types of constraints, namely, i) the cochannel constraint (CCC): the same channel cannot be assigned to certain pairs of radio cells simultaneously, ii) the adjacent channel constraint (ACC): channels adjacent in the frequency spectrum cannot be assigned to adjacent radio cells simultaneously, and iii) the cosite constraint (CSC): the channels assigned to the same radio cell need minimal frequency separation between cells. If these EMC constraints are not satisfied, interference occurs. Under these constraints, we can define the channel assignment problem as follows [4]:

$$\text{Minimize } Z$$

subject to

$$\sum_{p=1}^{Z} f_{ip} = d_i \qquad \text{for } 1 \le i \le N,$$

$$|p-q| \ge C_{ij} \qquad \text{for } 1 \le p,q \le Z \text{ and } 1 \le i,j \le N \text{ such that } f_{ip} = f_{jq} = 1, \text{ and}$$

$$f_{ip} = \begin{cases} 0 \\ 1 \end{cases} \quad \text{if channel } p \text{ is } \left\{ \begin{array}{l} \text{not assigned} \\ \text{assigned} \end{array} \right\} \text{to cell } i$$

where

Z: the number of available channels,
N: the number of cells,
D: demand vector $D = (d_1, d_2, ..., d_N)$, where d_i is the demand of channels for cell i, and
C: compatibility matrix $C_{N \times N}$, where each element C_{ij} represents the required minimum distance of separations between two the channels assigned to cell i and cell j.

The status of channels allocated to each cell can be represented with an $N \times Z$ binary matrix F (see Figure 4). Each element of F has value 0 or 1; if the j^{th} channel is assigned to the i^{th} cell, f_{ij} has value 1, otherwise f_{ij} has value 0.

Consider a simple channel assignment problem proposed in [14]; there are 4 cells, and the compatibility matrix and the demand vector are given as

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

and $D = (1, 1, 1, 3)$. The diagonal elements $C_{ii} = 5$ mean that if any two channels are assigned to the same cell, interference occurs unless the frequencies are apart by at least 5. With the above compatibility matrix and the demand vector, an optimal assignment is shown in Figure 1.

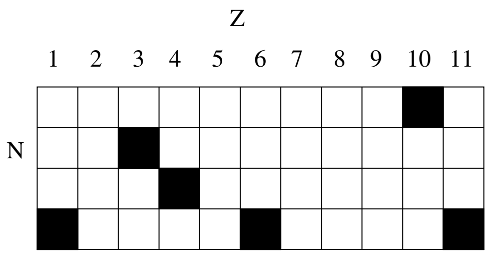

Figure 1: An interference-free assignment for the network with 4-cell and 11-channel.

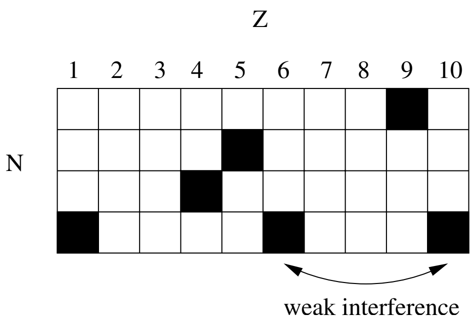

Figure 2: A weak-interference assignment for the network with 4-cell and 10-channel.

However, if there are only ten channels available, all the demands cannot be satisfied; we need to minimize the severity of interference. Figure 2 shows a solution with relatively weak cosite interference in cell 4 [2].

Generally, the fixed channel assignment problem (FCAP) is equivalent to the generalized graph-coloring problem [13][14][15]. We can obtain a graph with vertices and edges, where each vertex corresponds to a request of cells and the weight of the edge between two vertices is the required minimum separation for assigning two cells [15]. If all C_{ij}'s are 0 or 1, then there exists only the cochannel constraint and the FCAP reduces to the classical graph-coloring problem. Since the graph-coloring problem (either classical or generalized) is NP-complete [8], so is FCAP [14]. Hitherto a number of approximation algorithms have been proposed for FCAP. These include graph theoretic ordering approaches [14], neural network [6][15], simulated annealing [5], and genetic algorithms (GAs) [4][13].

The studies of FCAP can be classified into two approaches. One focuses on minimizing the total number of channels assigned to the cells with all constraints satisfied [14][16]. The other proposes appropriate cost functions and attempts to minimize the costs [4][13][6]. The cost should be zero when an assignment is conflict-free.

There have been a number of studies for FCAP using GAs. Ngo and Li [13] determined the number of available channels as lower bound using a graph theoretic method, and considered the interference cost as the fitness value. As Ngo and Li [13] suggested, Dirk and Ulrich [4] assumed the total number of available channels as lower bound according to the lower-bound rule of Gamst [7], used the blocking rate as the evaluation cost, and searched for the assignments with cost value zero. They searched for the conflict-free assignments with minimum channel span.

However, in practice, the available channels are limited, and the requests for channels often overflow beyond the capacity. In this situation, minimizing the total channel span is meaningless [9] and it is more useful to attempt to obtain the best assignments possible, given the number of channels [15]. Jin *et al.* [10] suggested a new formulation for a limited channel bandwidth below the lower bound. With the limited channel environment, all constraints cannot always be satisfied and we should allow blocked calls and/or the interference [10][9][15]. They consider the non-assigned requests and the violation of EMC constraints as blocking rate and interference cost, respectively. An optimal assignment maximally satisfies the demands of cells and minimizes the interference between cells. We use this model in this study.

There are two types of channel assignment problems: the fixed channel assignment and the dynamic channel assignment. In fixed channel assignment (FCA), one-time assignment is performed; in dynamic channel assignment (DCA), assignments are repeated with changing requests. FCA is more important in situation with heavy traffic loads [13]. Even in the situation in which DCA is used, the initial solution is usually provided by FCA, and DCA keeps modifying on it [15]. The running time is thus not very critical in FCA.

In this paper, we propose a hybrid genetic algorithm for FCA with limited channel bandwidth. Basically, we follow the formulation proposed in [10]. We designed a GA with two-dimensional chromosomes and an effective local optimization heuristic with horizontal/vertical searches.

The rest of the paper is organized as follows. In Section 2, we describe the formulation used in this paper. The proposed hybrid GA and local optimization algorithm are presented in Section 3. The experimental results are provided in Section 4. Finally, Section 5 summarizes the study.

2 Fixed Channel Assignment Problem

If a call cannot be assigned a channel, we say it is *blocked*. The *blocking rate* (the damage cost by blocked calls) is defined as,

$$\sum_{i=1}^{N} \sum_{j=DF_i+1}^{n_i} P(X_i = j)(j - DF_i)$$

where

F_i: the amount of channels assigned to cell i,
X_i: a random variable of required channels in cell i,
n_i: the expected number of requests in cell i, and
D: the number of channels provided by each frequency.

In assignments with the fixed channel condition, EMC constraints may be violated and the violation brings *interference*. EMC constraints are provided by the compatibility matrix C. In the matrix, each diagonal element C_{ii} represents the cosite constraint, and each non-diagonal element C_{ij} represents the minimum separation distance between any two frequencies assigned to cells i and j [13]. If $C_{ij} = 1$, it represents the cochannel constraint, and if $C_{ij} = 2$, it represents the adjacent channel constraint. Interference cost is defined as follows:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{p=1}^{Z} \sum_{q=1}^{Z} f(i, j, p, q)$$

where
N: the number of cells,
Z: the number of available frequencies,

$$f(i,j,p,q) = \begin{cases} 0, & \text{if } |p-q| \geq C_{ij} \\ f_{ip} f_{jq} \psi_C(C_{ij} - |p-q|), & \text{if } |p-q| < C_{ij} \text{ and } i = j \\ f_{ip} f_{jq} \psi_A(C_{ij} - |p-q|), & \text{otherwise.} \end{cases}$$

ψ_C and ψ_A are some strictly increasing functions.

Since the number of available channels is limited, it is hard to satisfy both of the constraints. The total damage due to an assignment is formulated as

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{p=1}^{Z} \sum_{q=1}^{Z} f(i,j,p,q) + \alpha \sum_{i=1}^{N} \sum_{j=DF_i+1}^{n_i} P(X_i = j)(j - DF_i)$$

where α is a weighting factor. In the formula, the first term represents the interference cost and the second term represents the blocking cost.

3 A Hybrid GA for the FCAP

In this section, we describe the proposed GA for FCAP. Figure 3 shows the template of a hybrid steady-state GA.

```
create initial population of a fixed size;
do {
        choose parent1 and parent2 from population;
        offspring ← crossover(parent1, parent2);
        mutation(offspring);
        local-optimization(offspring);
        replace(population, offspring);
} until (stopping condition);
return the best individual;
```

Figure 3: A typical hybrid steady-state GA

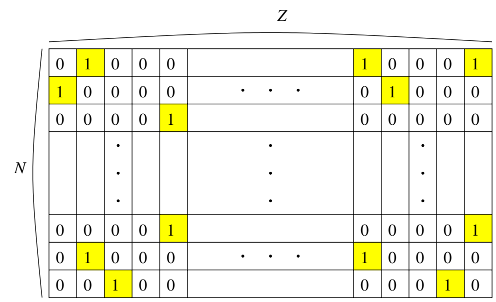

Figure 4: Two-dimensional chromosome for a solution

3.1 2D Representation

We represent a solution by a binary $N \times Z$ matrix. N is the number of cells and Z is the number of channels. If a gene $f_{ip} = 1$, then the p^{th} channel is assigned to the i^{th} cell (e.g., see Figure 4).

3.2 2D Crossover

A two-dimensional encoding/crossover pair can reflect more geographical linkages of genes than one-dimensional encoding/crossover pairs [12]. Cohoon and Paris [3] proposed a two-dimensional crossover which chooses a small rectangle from one parent and copies the genes in the rectangle into the offspring, with the rest of the genes copied from the other parent. Anderson *et al.* [1] suggested the block-uniform crossover on $n \times n$ grid. It divides the grid into $i \times j$ blocks at random; each block of one parent is interchanged randomly with the corresponding block of the other parent based on a pre-assigned percentage.

Although two-dimensional encoding can preserve more geographical relationships among the genes, when traditional straight-line-based cutting strategies are used, the power of new-schema creation is far below that of the crossovers on linear encodings [11].

Geographic crossover was suggested to resolve this problem [11] [12]. In the case of a two-dimensional encoding, it chooses a number of monotonic lines,

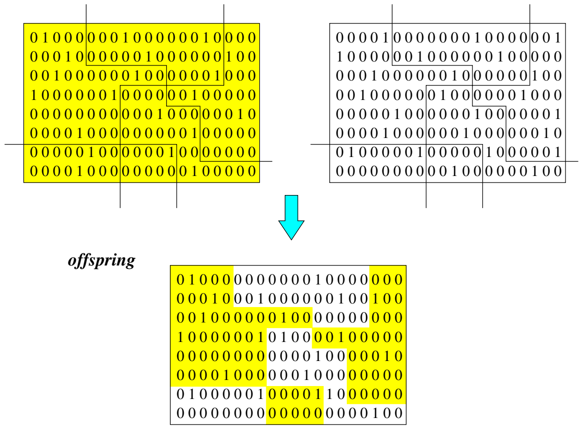

Figure 5: An example of geographic crossover

divides the chromosomal domain into two equivalence classes, and alternately copies the genes from the two parent chromosomes. We used geographic crossover in this work. Figure 5 shows an example geographic crossover operator for this problem. By combining two-dimensional representation and geographic crossover, we are pursuing both reduced information loss in the stage of encoding and the power of new-schema creation.

3.3 Mutation

On each cell i, we generate two random numbers. One is to get a channel number p and the other is a binary random number. We assign 0 or 1 to f_{ip} depending on the number. We control the total number of 1's in row i not to exceed d_i of the demand vector in any case.

3.4 HV-Move: the Local Optimization

We devised a local search heuristic to fine-tune around local optima. First, we fix the blocking rate and reduce the interference cost by a row-based search. Next, we reduce the interference and blocking cost simultaneously by a column-based search. The process of local search is performed by moving 1's to more attractive chromosomal positions.

Horizontal Search (Row-Based Search)

In the horizontal search, we redistribute the gene value 1's in each row so that the interference cost in the corresponding cell is minimized. In this search, the number of assigned channels in each cell does not change.

Vertical Search (Column-Based Search)

The vertical search is performed after the horizontal search and reduces the blocking rate and interference cost simultaneously. We redistribute the gene value 1's in each column so that the sum of blocking rate and interference cost is minimized. This search changes the number of channels in the cells.

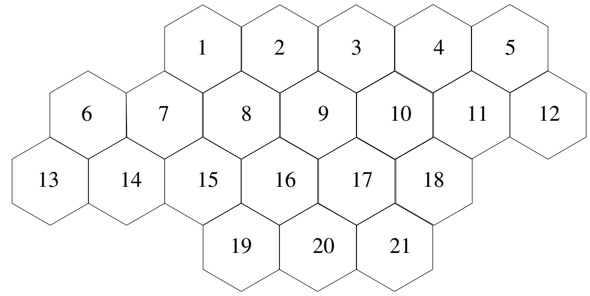

Figure 7: The 21-cell system

Figure 6 shows the outline of the local optimization algorithm. For each gene with value 1, we tries to move 1 to another position in the same row. If the move gives some gain, the value 1 moves. Then we move the value 1 to another position in the same column if the move gives some gain. Mark the position to which the value 1 finally moved. After performing the above process for all genes with value 1, we repeat the above process as far as there is at least one marked position.

4 Experimental Results

4.1 Benchmark Problems

The 21-cell system is a useful benchmark for the channel assignment problem (see Figure 7). The compatibility matrix C_3, C_4, C_5, and the demand vector D of each cell are based on [10], [9], and [6]. Figure 9 shows the compatibility matrices and Figure 10 represents the demand vectors. The benchmark set is composed of 8 problems, originated from [10] and [9]. Table 1 shows the specification of the problems. In our experiments, we used the function $\psi_C(X) = 5^{x-1}$ and $\psi_A(X) = 5^{2x-1}$ to evaluate chromosomes and set the weight of blocking cost $\alpha = 1000$[1], as in [10] and [9].

4.2 Experimental Parameters

In our experiments, the population size was set to 50. In selection, the probability to select the best chromosome was given four times higher than the worst. Mutation rate was 0.01%. Our GA is a steady-state GA; each generation produces one offspring and re-

[1]The weight of blocking cost was mistakenly written as 10,000 in [10] and [9]. We corrected it to 1,000 by personal communication with them.

```
HV–Move
{
    count ← 0;
    mark 1 on all the positions of value 1;
    repeat
    {
        count ← count + 1;
        for each row
            for each gene with mark count in the row
            {
                if there exists gain by moving the value 1 to another position in the same row
                    then  move the value 1;
                if there exists gain by moving the value 1 to another position in the same column
                    then  move the value 1;
                if there was any move in the above two trials
                    then  mark the finally moved position with count;
            }
    } until (there is nothing marked with count);
}
```

Figure 6: The outline of the local optimization algorithm

places with it one of the chromosomes in the population. The stopping condition is a fixed number of generations (we set the number to be 10,000). We performed the experiments on Pentium III 750 MHz.

4.3 Results

The experimental results are summarized in Table 2. The costs were dramatically reduced compared with [10] and [9]. In an extreme case (Problem 1), we found a solution of cost 0.385 while Horng *et al.* [9] reported 203.266 as the best solution cost.

Figure 8 shows the numbers of horizontal moves and vertical moves over the generations. The numbers were summed up every 200 generations. In the figure, one can observe that the horizontal moves occur more often than the vertical moves. Although not very often, the vertical moves steadily occurred over the generations.

5 Conclusions

A hybrid GA was proposed to solve the fixed channel assignment problem that assign limited channels to the requests of cells. Solutions were represented by two-dimensional chromosomes and the geographic crossover was applied. To help the GA's fine-tunning, we devised a local optimization heuristic that performs row-based search and column-based search.

We may consider the parallelization of our method. Although the suggested GA showed dramatic improvement over the previous studies, we believe that there still remains room for further improvement, particularly in the local optimization part. We plan to apply our algorithm to other benchmark problems [15] and larger scale problems.

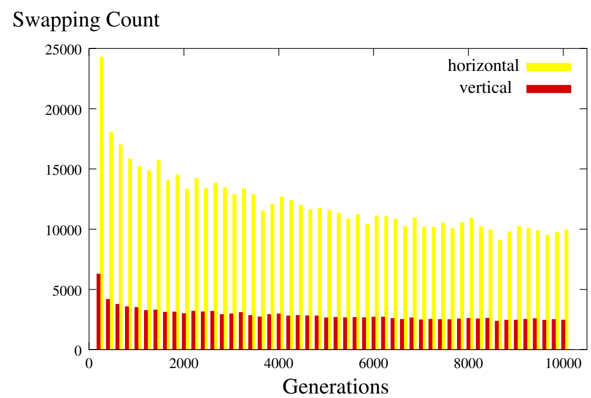

Figure 8: The swapping count in the local optimization

Acknowledgments

We thank Mr. Ming-Hui Jin for providing us the evaluation function procedure. This work was supported by KOSEF through Statistical Research Center for Complex Systems at Seoul National University and Brain Korea 21 Project. The RIACT at Seoul National University provided research facilities for this study.

Table 1: The Specification of the Problems

Problem	No. of cells(N)	No. of available channels(Z)	Compatibility matrix(C)	Communication load table(μ, δ)
1	21	60	C_3	$D_1^\dagger$
2	21	60	C_4	D_1
3	21	60	C_5	D_1
4	21	60	C_4	D_2
5	21	60	C_5	D_2
6	21	40	C_5	D_1
7	21	40	C_5	D_2
8	21	64	C_5	D_3

† The demand vector was mistakenly written as D_3 in [10] and [9]. We corrected it to D_1 by personal communication with them.

Table 2: Experimental Results

Problem	Previous works			Our GA results	
	Best [10]	CPU[1][10]	Best [9]	Best(Average[3])	CPU[2]
1	217.947	34976	203.266	0.385(0.510)	65504
2	276.623	42807	271.366	27.945(30.881)	88692
3	2013.751	39226	1957.366	63.089(79.346)	89918
4	950.995	31465	906.299	675.849(684.134)	95585
5	4495.609	45712	4302.298	1064.090(1092.484)	87905
6	4857.711	27412	4835.366	1149.755(1227.302)	35790
7	21700.624	54426	20854.300	5636.684(5831.756)	37323
8	58089.148	42248	53151.570	41883.012(41967.549)	135224

1. CPU seconds on Pentium II 400 MHz.
2. CPU seconds on Pentium III 750 MHz.
3. Average over 30 runs.

References

[1] C. A. Anderson, K. F. Jones, and J. Ryan. A two-dimensional genetic algorithm for the Ising problem. *Complex Systems*, 5:327–333, 1991.

[2] G. Chakraborty. An efficient heuristic algorithm for channel assignment problem in cellular radio network. *IEEE Trans. on Vehicular Technology*, 50:1528–1539, November 2001.

[3] J. P. Cohoon and W. Paris. Genetic placement. In *IEEE International Conference on Computer-Aided Design*, pages 422–425, 1986.

[4] B. Dirk and K. Ulrich. A new strategy for the application of genetic algorithms to the channel-assignment problem. *IEEE Trans. on Vehicular Technology*, 48(4):1261–1269, 1999.

[5] M. Duque, D. Kunz, and B. Rueber. Channel assignment for cellular radio using simulated anealing. *IEEE Trans. on Vehicular Technology*, 42(1):14–21, 1993.

[6] N. Funabiki and Y. Takefuji. A neural network parallel algorithm for channel assignment problems in cellular radio networks. *IEEE Trans. on Vehicular Technology*, 41(4):430–437, 1992.

[7] A. Gamst. Some lower bounds for a class if frequency assignment problems. *IEEE Trans. on Vehicular Technology*, 35:9–14, 1986.

[8] M. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completen ess.* Freeman, San Francisco, 1979.

[9] J.Z. Horng, M.H. Jin, and C.Y. Kao. Solving fixed channel assignment problems by evolutionary approach. In *Genetic and Evolutionary Computation Conference*, pages 351–358, 2001.

[10] M.H. Jin, H.K. Wu, J.Z. Horng, and C.H. Tsai. An evolutionary approach to fixed channel assignment problems with limited bandwidth constraint. In *IEEE International Conference on Communications*, pages 398–412, 2001.

[11] A. B. Kahng and B. R. Moon. Toward more powerful recombinations. In *Sixth International Conference on genetic Algorithms*, pages 96–103, 1995.

[12] B. R. Moon, Y. S. Lee, and C. K. Kim. GEORG: VLSI circuit partitioner with a new genetic algorithm framework. *Journal of Intelligent Manufacturing*, 9(5):401–412, 1998.

[13] C.Y. Ngo and V.O.K. Li. Fixed channel assignment in celluar radio networks using a modified genetic algorithm. *IEEE Trans. on Vehicular Technology*, 47(1):163–171, 1998.

[14] K.N. Sivarjan, R.J. McEliece, and J.W. Ketchun. Channel assignment in celluar radio. In *IEEE Vehicular Technology Conference*, pages 846–850, 1989.

[15] K. Smith and M. Palaniswami. Static and dynamic channel assignment using neural networks. *IEEE Journal on Selected Areas in Communications*, 15(2):238–249, 1997.

[16] W. Wang and C.K. Rushforth. A adaptive local-search algorithm for the channel-assignment problem(CAP). *IEEE Trans. on Vehicular Technology*, 45(3):459–466, 1996.

5	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0
1	5	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0
1	1	5	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0
0	1	1	5	1	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	5	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	5	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
1	1	0	0	0	1	5	1	1	0	0	0	1	1	1	1	0	0	1	0	0
1	1	1	0	0	1	1	5	1	1	0	0	0	1	1	1	1	0	1	1	0
1	1	1	1	0	0	1	1	5	1	1	0	0	0	1	1	1	1	1	1	1
0	1	1	1	1	0	0	1	1	5	1	1	0	0	0	1	1	1	0	1	1
0	0	1	1	1	0	0	0	1	1	5	1	0	0	0	0	1	1	0	0	1
0	0	0	1	1	0	0	0	0	1	1	5	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	5	1	1	0	0	0	0	0	0
1	0	0	0	0	1	1	1	0	0	0	0	1	5	1	1	0	0	1	0	0
1	1	0	0	0	1	1	1	1	0	0	0	1	1	5	1	1	0	1	1	0
1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	5	1	1	1	1	1
0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	5	1	1	1	1
0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	5	0	1	1
0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	0	5	1	1
0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	5	1
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	5

(a) The compatibility matrix C3

7	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0
1	7	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0
1	1	7	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0
0	1	1	7	1	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	7	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	7	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
1	1	0	0	0	1	7	1	1	0	0	0	1	1	1	1	0	0	1	0	0
1	1	1	0	0	1	1	7	1	1	0	0	0	1	1	1	1	0	1	1	0
1	1	1	1	0	0	1	1	7	1	1	0	0	0	1	1	1	1	1	1	1
0	1	1	1	1	0	0	1	1	7	1	1	0	0	0	1	1	1	0	1	1
0	0	1	1	1	0	0	0	1	1	7	1	0	0	0	0	1	1	0	0	1
0	0	0	1	1	0	0	0	0	1	1	7	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	7	1	1	0	0	0	0	0	0
1	0	0	0	0	1	1	1	0	0	0	0	1	7	1	1	0	0	1	0	0
1	1	0	0	0	1	1	1	1	0	0	0	1	1	7	1	1	0	1	1	0
1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	7	1	1	1	1	1
0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	7	1	1	1	1
0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	7	0	1	1
0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	0	7	1	1
0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	7	1
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	7

(b) The compatibility matrix C4

7	2	1	0	0	1	2	2	1	0	0	0	0	1	1	1	0	0	0	0	0
2	7	2	1	0	0	1	2	2	1	0	0	0	0	1	1	1	0	0	0	0
1	2	7	2	1	0	0	1	2	2	1	0	0	0	0	1	1	1	0	0	0
0	1	2	7	2	0	0	0	1	2	2	1	0	0	0	0	1	1	0	0	0
0	0	1	2	7	0	0	0	0	1	2	2	0	0	0	0	0	1	0	0	0
1	0	0	0	0	7	2	1	0	0	0	0	2	2	1	0	0	0	0	0	0
2	1	0	0	0	2	7	2	1	0	0	0	1	2	2	1	0	0	1	0	0
2	2	1	0	0	1	2	7	2	1	0	0	0	1	2	2	1	0	1	1	0
1	2	2	1	0	0	1	2	7	2	1	0	0	0	1	2	2	1	1	1	1
0	1	2	2	1	0	0	1	2	7	2	1	0	0	0	1	2	2	0	1	1
0	0	1	2	2	0	0	0	1	2	7	2	0	0	0	0	1	2	0	0	1
0	0	0	1	2	0	0	0	0	1	2	7	0	0	0	0	0	1	0	0	0
0	0	0	0	0	2	1	0	0	0	0	0	7	2	1	0	0	0	0	0	0
1	0	0	0	0	2	2	1	0	0	0	0	2	7	2	1	0	0	1	0	0
1	1	0	0	0	1	2	2	1	0	0	0	1	2	7	2	1	0	2	1	0
1	1	1	0	0	0	1	2	2	1	0	0	0	1	2	7	2	1	2	2	1
0	1	1	1	0	0	0	1	2	2	1	0	0	0	1	2	7	2	1	2	2
0	0	1	1	1	0	0	0	1	2	2	1	0	0	0	1	2	7	0	1	2
0	0	0	0	0	0	1	1	1	0	0	0	0	1	2	2	1	0	7	2	1
0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	2	2	1	2	7	2
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	2	2	1	2	7

(c) The compatibility matrix C5

Figure 9: The compatibility matrices

cell	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
μ	5	5	5	8	12	25	30	25	30	40	40	45	20	30	25	15	15	30	20	20	25
σ	1.2	1.1	1.15	1.6	2.24	5.0	5.72	5.0	6.1	8.1	8.02	9.11	4.07	5.99	5.61	3.14	3.07	5.86	4.12	3.98	5.18

D_1

cell	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
μ	8	25	8	8	8	15	18	52	77	28	13	15	31	15	36	57	28	8	10	13	8
σ	1.61	4.88	1.52	1.49	1.61	3.11	3.52	9.73	11.62	5.1	2.15	2.66	4.72	2.77	4.93	8.64	3.92	1.25	1.72	2.14	1.27

D_2

cell	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
μ	59.38	105.56	95.79	92.83	61.35	191.81	39.53	62.84	42.58	34.73	36.02	41.59	80.7	35.67	61.91	42.22	65.61	52.45	85.78	33.9	80.62
σ	6.23	8.98	14.07	11.08	5.61	21.62	6.78	8.72	10.3	6.63	7.92	6.29	11.34	6.02	5.61	5.96	7.39	5.86	9.92	3.52	6.18

D_3

Figure 10: Communication load tables