Fault Spectrum Analysis for Fast Spare Allocation in Reconfigurable Arrays *

Wengang Che and Israel Koren

Department of Electrical and Computer Engineering
University of Massachusetts at Amherst
Amherst, MA 01003

Abstract

Repairing a reconfigurable array by row and column replacement using SR rows and SC columns was shown to be an NP-complete problem. In order to reduce the search time, we propose to apply a three phase procedure. In the first phase, we suggest to use a heuristic to find good, but not necessarily optimal, feasible cover for the faulty array. Only if the heuristic method fails to generate a feasible cover, the array is examined to find out whether it is repairable at all. If deemed economical, repairable chips will undergo an exhaustive analysis. This three phase strategy can considerably reduce the average time for repair analysis. Searching for a good heuristic to be applied in phase 1, we investigated the fault distribution pattern on the faulty array and considered the effect of a row or column replacement on this fault distribution. Accordingly, a k degree fault spectrum for a bipartite graph is defined and a maximum spectrum is introduced as a heuristic for selecting vertices. We prove that the vertices which are most likely to be included in the feasible cover will be selected by our heuristic. Consequently, a fast method to generate a feasible cover is proposed and a suitable algorithm is developed.

1 Introduction

Many strategies have been proposed for reconfiguring a faulty memory or processor array [2]. One method which has been extensively studied is the use of spare rows and columns to replace the faulty rows and columns of the array. This strategy has been widely adopted since it needs few additional connections to reconfigure the faulty chip. Furthermore, the repair is easily realized, especially in memory integrated circuits [1] [9] [11] [12] . The problem of repairing a faulty array of size $M \times N$ using at most SR rows and SC columns was proved to be $NP-hard$ [9].

An efficient spare allocation method was proposed by Kuo and Fuchs [9] using a cost function derived from laser repair processes [11]. This branch and bound algorithm reaches an optimal result without examining all other feasible covers, and consequently, the search time is reduced considerably. In order to speed up the search for a feasible cover, different heuristics were investigated in [6] [7] [9] and others.

As pointed out in [10], most existing reconfiguration heuristics follow one of the following two strategies: *early-abort* and *partial solution* . An example for the first is the *maximum matching* method [7] [9] that tests whether the array is repairable or not. If not, an *early-abort* is performed. For the *partial solution* method, a *must-repair* step was proposed [4] [9] [12], so that the whole search space is reduced. Another ***mandatory-repair*** method proposed the use of a ***critical set*** which must be included in every minimum cover [7]. Recently, Hadas and Liu [6] presented an effective method to reduce the partial solutions generated in the exhaustive search using ***excess-k*** critical sets.

*This work was supported in part by NSF under contract MIP-9013013.

0-8186-2837-5/92 $03.00 © 1992 IEEE

However, the problem is still $NP-hard$ whatever method is adopted. In industrial environment, a good heuristic method running in polynomial time is more practical for larger arrays. In [9], a heuristic approximation was proposed selecting the vertex with highest degree. Based on a probabilistic model, an average-case polynomial time algorithm is also given in [10].

In fact, the previous methods have rarely considered the fault distribution pattern on the faulty array and the changes in this pattern after replacing rows or columns by spares. Actually, the existence of a feasible cover for a faulty array depends mainly on the fault distribution pattern on the array and the effects on this pattern due to elimination of rows or columns. Recent investigations have shown that for a certain pattern of distribution, the repair analysis can be accelerated [1] [5] [8] [10]. Unfortunately, in practice, defects don't always occur according to a certain presumed distribution. Therefore, a direct method to analyze the fault distribution on a device is more practical.

In this paper, we introduce a heuristic approximation with good performance by analyzing the fault distribution on a faulty array. We call this method the ***k degree spectrum*** analysis. According to this method, the effects on the fault pattern of the array due to row and column replacements are estimated. We show that the lines with maximum spectrum at the lowest degree are most likely to be included in a feasible cover. As a result, the most likely vertices for a feasible solution will be selected step by step in polynomial time $O(M^3)$, where M is the size of the bipartite graph for the faulty array.

As an overall algorithm for repair analysis, we propose a three phase strategy. In the first step of the strategy, a ***heuristic approximation*** is applied and a good feasible cover is found in most cases. When the heuristic approximation fails to generate a feasible cover, a ***maximum matching*** is applied to find out whether there exists a minimum cover for the faulty array. If there exists one, an exhaustive analysis is performed. Therefore, the overall time for repair analysis is estimated as $pT_a + (1-p)(T_m + T_e)$, where T_a, T_m and T_e are the average execution time for the three phases, respectively, and p is the probability of successful repair by the heuristic approximation. Consequently, this strategy depends largely on the performance of the heuristic approximation because a larger p will reduce the average time considerably.

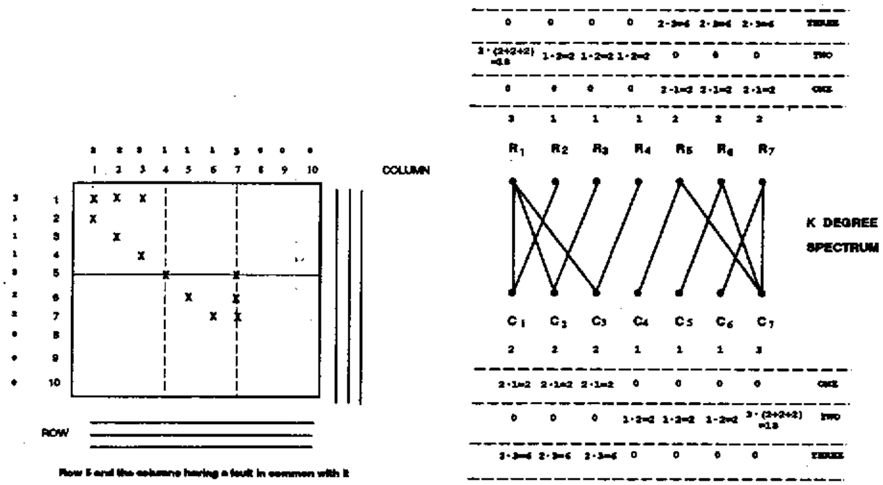
2 Fault Spectrum Analysis

For simplicity, the reconfigurable array is assumed to be a rectangle having $M \times N$ cells with SR spare rows and SC spare columns. The problem can be described as a bipartite graph $\{A \bigcup B, E\}$ [9], where the lines of the array are represented by vertices in the graph. Two vertices, corresponding to a row and a column in the array, are connected by an edge if there is a fault in common to the row and the column. The degree of a vertex in $\{A \bigcup B\}$ is defined as the number of edges connected to this vertex, or the number of faults on the corresponding line.

To find a set of vertices which is most likely to be included in a feasible cover, we have to analyze the fault distribution of the array and the effect of the row and column replacements on this fault pattern.

2.1 Repair Analysis with the k Degree Spectrum

Obviously, to repair the faulty array, the involved basic element is a row or column. When replacing a row (or column) by a spare row (or column), the faults on the row (or column) are erased. After deleting the row (or the column), the fault distribution on the array has been changed. In this case, the affected lines are those columns or rows which had faults in common with the replaced row or column.


COLUMN
ROW
Row 5 and the columns having a fault in common with it
0 0 0 0 2·3=6 2·3=6 2·3=6 THREE
3·(2+2+2)=18 1·2=2 1·2=2 1·2=2 0 0 0 TWO
0 0 0 0 2·1=2 2·1=2 2·1=2 ONE
3 1 1 1 2 2 2
R1 R2 R3 R4 R5 R6 R7
K DEGREE SPECTRUM
C1 C2 C3 C4 C5 C6 C7
2 2 2 1 1 1 3
2·1=2 2·1=2 2·1=2 0 0 0 0 ONE
0 0 0 1·2=2 1·2=2 1·2=2 3·(2+2+2)=18 TWO
2·3=6 2·3=6 2·3=6 0 0 0 0 THREE

(a) A faulty array

(b) The bipartite graph and its k degree spectrum for (a)

Figure 1: Repair analysis

To be more explicit, we investigate a faulty row. If we replace this row by a spare row, we will never use another spare row to repair the faults on this row. As for columns which have faults in common with the row, the situation is different. If there is a single fault on such a column, i.e, it is a one degree column vertex, after the row has been replaced by a spare row, the column becomes fault-free. Consequently, no more columns or rows are necessary to repair this column. However, if the column had more than one fault in it, i.e., the vertex's degree was $k \geq 2$, at least one spare column or at most $k-1$ additional spare rows are necessary to repair the remaining faults in the column. For a faulty column, the analysis is similar.

For instance, if we want to repair row 5 in Figure 1 (a), there are two columns having a fault in common with this row. Denote by D_{r_i} and D_{c_j} the degree of row i and column j, respectively. The fault pattern of row 5 and the effect on the fault pattern of the array due to the replacement of row 5 can be described through $\{D_{r_5} = 2, D_{c_4} = 1, D_{c_7} = 3\}$. After this row has been replaced by a spare row, no more columns or rows are necessary to repair the faults on column 4. This is because the fault degree of this column is one. However, one needs at least one column or at most two additional rows to repair the remaining faults in column 7 because its degree is three.

Therefore, when the degree of a line in the faulty array is D_l, the effect of replacing the line on the fault distribution pattern of the array is determined by the line's degree and that of the orthogonal lines having faults in common with the line. There are exactly D_l orthogonal lines affected. Different orthogonal lines with different degrees will be affected differently, but orthogonal lines having the same degree should be classified as one group. In mathematical terms, the fault pattern of the line and the effect on the fault pattern due to the line's replacement are defined by the so called fault spectrum.

DEFINITION 2.1: Fault Spectrum of a Line (or Vertex) l:

The fault distribution pattern on a line l with degree D_l can be represented by the degrees of the orthogonal lines O_i which have a fault in common with the line l. To each of the orthogonal lines, we assign a weight which equals the product of D_l and D_{o_i}. The sum of the weights of the orthogonal lines with degree k is called the k degree spectrum of the line l and is denoted by $S_{k,l}$.

$$S_{k,l} = D_l \cdot \sum_i (D_{o_i} \cdot \delta_{k,D_{o_i}}), \tag{1}$$

where $\delta_{k,D_{o_i}} = 0$ when $k \neq D_{o_i}$, and $\delta_{k,D_{o_i}} = 1$ when $k = D_{o_i}$.

For all lines of a given array, we can obtain their k degree spectrum using the above definition. In Figure 1 (b), the bipartite graph for the array in Figure 1 (a) and its k degree spectrums are illustrated. For example, the degree of the vertex R_5 is 2, only C_4 out of the column vertices has a common edge with R_5 and a degree of 1. Thus, the 1 degree spectrum for R_5 is $2 \cdot 1 = 2$. There is no column vertex of degree 2 having edges in common with R_5, so the 2 degree spectrum for R_5 is $2 \cdot 0 = 0$. Obviously, the 3 degree spectrum for R_5 is $2 \cdot 3 = 6$. No other higher degree spectrums exist for R_5. All k degree spectrums of the vertices for the bipartite graph are shown in Figure 1 (b).

2.2 Properties of the k Degree Spectrum

Property 1: For a faulty array $\{A \bigcup B, E\}$, if we select the line with the maximum spectrum value at the lowest degree from 1 to $max\{M, N\}$, the number of spare lines used to repair the faults on the selected line and the faults on the orthogonal lines which have a fault in common with this line will be minimized in most cases.

(1). For a row with degree D_r, if we use a spare row to replace it, no other rows are necessary to repair the faults on the row. Therefore, selecting a row with maximum D_r will cover the maximum number of faults.

(2). For a k degree column having a fault in common with the row, when $k = 1$, no other columns or rows are necessary to repair the column. When $k \geq 2$, at least one column or at most $k-1$ additional rows are necessary to repair the remaining faults on the column. So a column with lower degree will use less additional spare lines to cover the faults on it. Therefore, in order to reduce the number of additional spare lines to repair the faults on the corresponding columns, the row with the maximum number of orthogonal columns at the lowest degree will minimize the possibility of requiring extra spare lines to repair the faults on the related columns.

(3). To consider both situations in (1) and (2), it is necessary to use a weighting function that takes into account the number of faults on the row and the number of columns having the same degree from 1 to $max\{M, N\}$. At degree k, the weighting function that we suggest is $\frac{(D_r + N_k \cdot k)}{2}$ which satisfies:

$\frac{(D_r+N_k \cdot k)}{2} \geq \sqrt{D_r \cdot N_k \cdot k}$

where N_k is the number of the corresponding columns having degree k.

Note that the term $D_r \cdot N_k \cdot k$ is identical to the spectrum in Definition 1. Consequently, we are able to use the spectrum of the faulty array to choose a row having maximum spectrum at the lowest degree k where k ranges from 1 to $max\{M, N\}$. Replacing such a row by a spare row will increase the chances that we use as few spares as possible. As a result, the selected row is very likely to be included in a feasible cover.

(4). The case of columns is identical. □

Based on this property, a line with the largest spectrum at the lowest degree will take care of the largest number of faults on the line and fewer additional spare lines will be required to repair the faults on its intersecting lines. Because we didn't investigate further relationships between the degree of the row and its orthogonal lines' degrees, the obtained result is only an approximation. However, from our experience, this approximation produces excellent results. In Figure 1 (b), $\{(C_1, C_2, C_3), (R_5, R_6, R_7)\}$ have maximum spectrum at degree 1, so they are selected first. This selection immediately leads to a feasible cover for repairing the faults on the array.

Property 2: In a bipartite graph $\{A \bigcup B, E\}$, if there is a subset $\{A_e \bigcup B_e, E_e\}$ containing more than one vertex whose k degree spectrums from 1 to $max\{M, N\}$ have the same value at every degree k, selecting any of them is equivalent in the repair process for the given array. We call them equivalence subsets.

Proof: Suppose there are two vertices in $\{A_e \bigcup B_e\}$. Because their spectrums from 1 to $max\{M, N\}$ are the same, their degrees are equal to each other and the fault pattern on the orthogonal lines which have faults in common with them are the same too. Therefore, in terms of repairing by a spare vertex, they are equivalent for the current bipartite graph. □

Based on this property, it is clear that, no matter which one of the equivalent vertices is selected first, the effect on the fault distribution of the array will remain the same. For instance, in Figure 1 (b), anyone of the vertices $\{(C_1, C_2, C_3), (R_5, R_6, R_7)\}$ has the same k degree spectrums from 1 to 3. Therefore, it doesn't matter which one of them is selected first while repairing the faulty array.

Property 3: In an equivalence subset $\{A_e \bigcup B_e, E_e\}$, if any row in A_e is replaced by a spare row, the spectrum of the remaining rows of the subset will increase or stay unchanged, the spectrums at the lowest degree of the remaining columns in B_e will decrease or stay unchanged. If a column has been replaced, the spectrum at the lowest degree of the columns will increase or stay unchanged, and that of the rows will decrease or stay unchanged.

Proof: Omitted for the sake of brevity.

This property indicates that if we select a row in an equivalence subset, the next selection would probably be another row in the subset with maximum spectrum at degree k. If the first selection is a column, the next would probably be a column too.

On the basis of the spectrum analysis, a feasible cover for the faulty array is found quickly in most cases. This is because we select the line with maximum spectrum among all rows and columns. This selection can be done in $O(M^2)$ time. Therefore, the time complexity for selecting a feasible cover is polynomial and is given by $O(M^3)$. For the example in Figure 1 (b), the unique cover for the faulty array is found in polynomial time rather than through an exhaustive search. Here we select the vertices with maximum spectrum from degree 1 to 4, and the feasible cover is $\{(C_1, C_2, C_3), (R_5, R_6, R_7)\}$.

3 Fast Algorithm for Feasible Cover

3.1 Heuristic Approximation

Based on the above discussion of k degree spectrum, it is reasonable to claim that a set of vertices with maximum spectrum at the lowest degree is included in a good feasible cover for a given bipartite graph. The algorithm for finding a set of vertices with maximum k degree spectrum at the lowest degree is as follows. Let $\{A_t \bigcup B_t\}$ denote a temporary bipartite graph in which a set of vertices with maximal k degree spectrum will be stored. At step k, different vertices with maximal spectrum at degree k will be selected and stored in $\{A_t \bigcup B_t\}$. Finally, when a vertex with maximum spectrum at k is found, or k becomes larger than the maximum degree of the faulty array, the algorithm has found the desired group of vertices. In the former, a vertex will be selected. In the latter, there are three kinds of vertices in $\{A_t \bigcup B_t\}$. The first kind consists of only rows, the second kind consists of only columns, the last kind consists of both rows and columns. Because the algorithm finds the maximal k degree spectrum by increasing k by one at each step, the vertices with the maximum spectrum at the lowest degree will be selected. The above algorithm can be summarized as follows.

```
/* k Degree Spectrum Analysis for $\{A \bigcup B, E\}$ */
  load $(A_t \bigcup B_t)$ with $(A \bigcup B)$;
  $k$ = minimal degree of $(A_t \bigcup B_t)$;
  while ( number of vertices in $(A_t \bigcup B_t) \geq 2$
  and $k \leq$ maximal degree )
  {
    find the vertices with maximum spectrum among $(A_t \bigcup B_t)$ at $k$ degree;
    replace $A_t$ by the new row vertices with maximum $k$ spectrum;
    replace $B_t$ by the new column vertices with maximum $k$ spectrum;
    $k = k + 1$;
  }
/* the vertices with the maximum spectrum are now in $\{A_t \bigcup B_t\}$ */.
```

Because the algorithm must examine $O(M)$ lines at every k, and k ranges from the lowest degree to the highest degree, the time complexity of this maximum spectrum algorithm is $O(M^2)$, where M is the size of the bipartite graph.

Using this k degree spectrum analysis, our heuristic approximation is as follows. First of all, a vertex or a set of vertices with maximum spectrum is obtained by k degree spectrum analysis. These vertices are stored in a partial solution record, then these vertices and their corresponding edges are deleted from the original bipartite graph. This procedures is repeated on the reduced bipartite graph until no edges are left. The final result is a feasible cover for the bipartite graph. Otherwise, the approximation fails to return a cover.

According to the properties of k degree spectrum, the search process will generate a unique solution if there are no equivalence subsets consisting of both rows and columns. This is because in every step of the selection, only one vertex is a candidate for selection. However, when equivalence subsets exist, it is uncertain how to select among these rows or columns. Different selections will lead to either a feasible cover or an infeasible cover under the restriction of the number of available spares. Therefore, a simple cost function is set up to count the number of spare rows and columns left at each partial solution. Thus, at a step where equivalence subsets exist, the rows of the subset are selected if the number

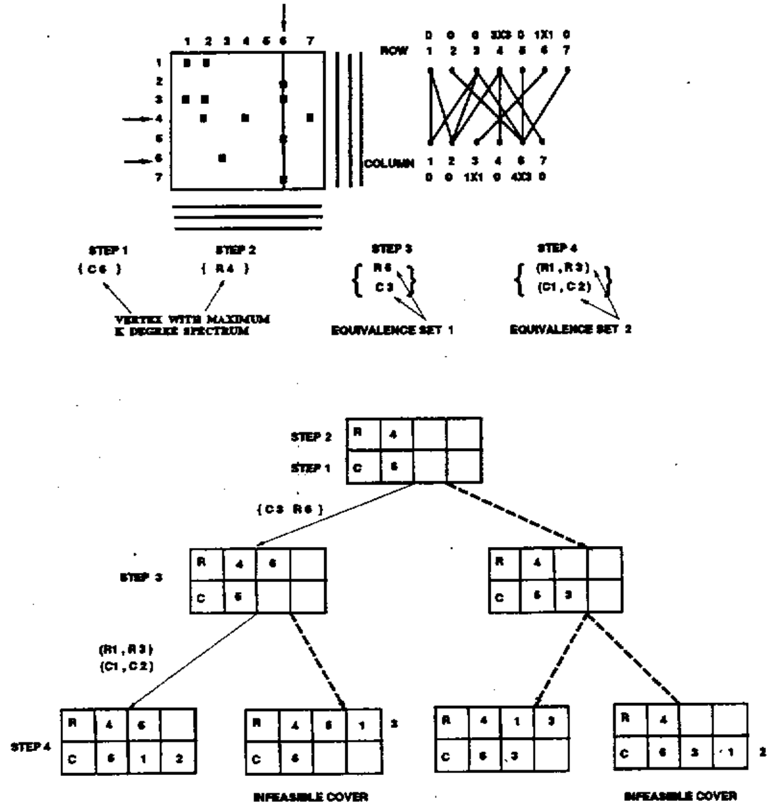


Figure 2: An example of repairing a faulty array

of spare rows left is larger than that of spare columns left. The columns of the subset are selected if the number of spare columns left is larger than that of spare rows left. In order to search for all possible covers, two partial solution records are generated, where one includes the rows in the equivalence subset, and the other includes the columns in the equivalence subset. This partial branch and bound method can be used to find another cover when the first selection is unsuccessful.

An example with 3 spare rows and 3 spare columns illustrating the generation of a feasible cover by the k degree maximum heuristic approximation is depicted in Figure 2. First of all, the 1 degree spectrum of the bipartite graph is calculated. In this 1 degree spectrum, C_6 has the maximum spectrum, therefore C_6 is selected first and no further k degree spectrum analysis is necessary. C_6 and the edges incident to C_6 are deleted from the bipartite graph. In the next step, the k degree spectrum for the reduced bipartite graph is recalculated and R_4 is identified as the vertex with maximum spectrum at degree 1. The deletion process is then repeated. Next, $\{R_6, C_3\}$ is found to have maximal spectrums. Since the remaining spares of both row and column for this record are identical, anyone of $\{R_6, C_3\}$ can be selected. R_6 is selected arbitrarily. The final bipartite graph contains an equivalence subset $\{(R_1, R_3), (C_1, C_2)\}$. The remaining spare rows and columns for the bipartite graph are examined. Thus, (C_1, C_2) is selected and a feasible cover is obtained.

Table 1: Results of repairing faulty arrays.

Array#	Size	# of Defects	Spare rows	Spare columns	Repairable ?	Spares used	Time(sec)
kuo1	128 × 128	5	4	4	yes	5	0.1
kuo2	128 × 128	15	4	4	no	-	0.1
kuo3	256 × 256	10	5	5	yes	9	0.1
kuo4	256 × 256	30	5	5	no	-	0.1
kuo5	512 × 512	15	5	5	yes	5	0.1
kuo6	512 × 512	19	10	10	yes	18	0.1
kuo7	512 × 512	45	10	10	no	-	0.1
kuo8	512 × 512	45	20	20	yes	38	0.1
kuo9	1024 × 1024	40	20	20	yes	36	0.1
kuo10	1024 × 1024	60	20	20	no	-	0.1
kuo11	1024 × 1024	200	20	20	no	-	0.1
kuo12	1024 × 1024	400	20	20	no	-	0.1
kuo13	1024 × 1024	400	20	20	yes	37	0.1

The average execution time is 0.1 sec for this group of data.

If no feasible cover is found, a backward search is conducted to the last partial solution recorded by another equivalence subset until a feasible cover is found or no further equivalence sets are left.

Thus, a set of most likely vertices to be included in a feasible cover is selected. The algorithm finds a feasible cover quickly. We have tested all the examples presented in [6] and [9], and the results demonstrate the good performance of our method. In Tables 1 and 2, the algorithm found the feasible cover for the given array in less than 0.3 second system time, on the average.

3.2 Three Phase Algorithm

As stated in the introduction, we propose a three phase algorithm. The first phase is the heuristic approximation to find a good feasible cover. If a suitable feasible cover is found, the repair analysis for this faulty array is terminated. If the approximation could not find a feasible cover, a maximum matching algorithm is employed [7][9]. If there is no minimum cover for the problem, the search is ended concluding that the faulty array can not be repaired. Otherwise, an exhaustive search is performed if deemed economical.

In Tables 1 and 2, we present the tested arrays and the results. In the first group of arrays, no exhaustive search was necessary. In the second group of arrays, only one example required an exhaustive search. The number of spares used in our algorithm is slightly larger than that used by the "excess-k" procedure in [6]. However, the average time used for repair analysis is smaller.

A precise comparison between the execution time of both algorithms is difficult since our algorithm was executed on a DECstation 5000/120 and the time reported for *excess-k* method is the execution time on an Encore Multimax.

As indicated in the introduction, the average overall execution time in our method is

Table 2: Comparison between the *excess* − *k* method [6] and the *k* degree spectrum method.

Array Sise	# of Faults	SR	SC	*k* Degree Spectrum			Excess-*k*		
				Repairable	Spares used	Time (sec)	Repairable used	Spares used	Time (sec)
512 × 512	80	10	10	yes	19	0.1	yes	18	0.4
512 × 512	99	17	16	yes	30	0.1	yes	30	0.5
512 × 512	107	12	12	yes	19	0.1	yes	19	0.3
512 × 512	114	8	14	yes	22	0.1	yes	22	0.8
512 × 512	196	22	22	yes	44	0.1	yes	43	5.5
512 × 512	216	25	25	yes	35	0.1	yes	35	0.7
512 × 512	311	32	32	yes	66	0.1	yes	64	2.5
512 × 512	397	15	35	yes	46	0.1	yes	45	1.9
512 × 512	574	35	45	yes	77	0.1	yes	77	10.8
512 × 512	630	50	50	yes	78	0.1	yes	78	2.5
512 × 512	1069	60	60	yes	113	0.3	yes	107	6.1
1024 × 1024	127	10	15	no	–	0.1	no	–	2.0
1024 × 1024	218	15	20	yes	35	0.1	yes	35	5.6
1024 × 1024	927	50	60	yes	107	6.4(*)	yes	107	4.5
1024 × 1024	1267	100	100	yes	120	0.1	yes	120	6.8
1024 × 1024	1538	75	68	yes	137	0.2	yes	137	19.2
1024 × 1024	532	20	30	yes	50	0.1	yes	46	0.5
1024 × 1024	1084	25	20	no	–	0.1	no	–	2.9
1024 × 1024	642	12	25	no	–	0.1	no	–	1.7
1024 × 1024	3221	50	36	no	–	0.5	no	–	23.9

The average execution time is 0.43 sec for this group of data by our method.
(*) exhaustive search applied

estimated by $pT_a + (1-p)(T_m + T_e)$. For the tested results in Table 1, the average execution time is 0.1 second. For that of Table 2, the average execution time is 0.43 second. For the arrays that we tested, p is approximatively 0.975. This means that for most of the faulty arrays, it is possible to find a good feasible cover by the *k degree* spectrum method in a short time.

4 Concluding Remarks

We proposed a new three phase strategy to solve the problem of spare allocation for reconfigurable arrays. The first phase uses a new heuristic approximation.

Using the maximum spectrum as a heuristic cost function, we are able to select a set of suitable lines which are most likely to be included in a feasible cover. From the *k* degree spectrum analysis, equivalence subsets with maximum spectrum are found. In an equivalence subset, each vertex has the same property for the elimination process. Therefore, selecting any one of them first will have the same effect on the final result. In order to obtain a feasible cover, a simple cost function which counts the number of spares left is used to guide the search process when the equivalence subset consists of both rows and columns.

The second step of the strategy is based on the maximum matching method. If the heuristic was unable to find a feasible cover, this phase is used to examine whether there exists a cover for the faulty array.

If not, the repair analysis is terminated concluding that there is no feasible cover for the faulty array. Otherwise, a exhaustive analysis is used to find out a feasible cover for the faulty array.

As indicated in Tables 1 and 2, this three step strategy returned a feasible cover quickly in most cases. Therefore, the average execution time for the repair analysis decreases considerably.

Acknowledgements

We would like to thank Professor W. K. Fuchs, Professor C.L. Liu, Mr. R. Libeskind-Hadas and Mr. Weiping Shi for providing the data in Tables 1 and 2.

References

[1] D. M. Blough, "On the Reconfiguration of Memory Arrays Containing Clustered Faults," *Proc. of IEEE International Symposium on Fault-Tolerant Computing*, 1991, pp.444-451.

[2] M. Chean and J. A. B. Fortes, "A Taxonomy of Reconfiguration Techniques for Fault Tolerant Processor Arrays," *IEEE Computer*, pp.55-69, Jan. 1990.

[3] K. L. Clarkson, "A Modification of Greedy Algorithm for Vertex Cover," *Information Processing Letters*, vol. 16, pp.23-25, 1983.

[4] J. Day, "A Fault Driven Comprehensive Redundancy Algorithm," *IEEE Design and Test*, vol. 2, pp.35-44, June 1985.

[5] R. W. Hadad, A. T. Dahbura and Anup B. Sharma, "Increased Throughput for the Testing and Repair of RAM's with Redundancy," *IEEE Trans. on Computers*, vol. 40, pp.154-166, Feb. 1991.

[6] R. L. Hadas and C. L. Liu, "Fast Search Algorithms for Reconfiguration Problem," *Proc. of the IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems*, 1991, pp.260-273.

[7] N. Hasan and C. L. Liu, "Minimum Fault Coverage in Reconfigurable Arrays," *Proc. of IEEE International Symposium on Fault-Tolerant Computing*, 1988, pp.348-353.

[8] I. Koren, Z. Koren and C. H. Stapper, "A Unified Negative Binomial Distribution for Yield Analysis of Defect Tolerant Circuits," *to appear, IEEE Trans. on Computers*, 1992.

[9] S. Y. Kuo and W. K. Fuchs, "Efficient Spare Allocation for Reconfigurable Arrays," *IEEE Design and Test*, pp.24-31, Feb. 1987.

[10] W. Shi and W. K. Fuchs, "Probabilistic Analysis and Algorithms for Reconfiguration of Memory Arrays," *to appear, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 1992.

[11] R. T. Smith, J. D. Chlipala, J. F. Bindels, R. G. Nelson, F. H. Fischer and T. F. Mantz, "Laser Programmable Redundancy and Yield Improvement in 64K DRAM," *IEEE Journal of Solid-State Circuits*, vol. SC-16, pp.506-513, Oct. 1981.

[12] M. Tarr, D. Boudreau and R. Murphy, "Defect Analysis System Speeds Test and Repair of Redundant Memories," *Electronics*, pp.175-179, Jan. 1984.