University of Tulsa Technical Report UTULSA-MCS-94-1, January 1994.

Chromosome Reduction in Genetic Algorithms*

Arthur L. Corcoran
corcoran@penguin.mcs.utulsa.edu

Roger L. Wainwright
rogerw@penguin.mcs.utulsa.edu

Mathematical and Computer Sciences Dept.
The University of Tulsa
600 South College Avenue
Tulsa, OK 74104-3189, USA

1 Introduction

Genetic Algorithms [8, 9] have been very successful as adaptive search techniques on a wide variety of problems. However, the GA has difficulty with some problems, such as those with rugged fitness landscapes or those which require non-binary genome representations. Recent research has focused on new theories, models and techniques for improving the performance of GAs on difficult problems. Many new ideas have been proposed, including: deception, messy GAs, island and cellular parallel population models, analysis of fitness landscapes, "the royal road test", Markov chain analysis, "the simple GA", and "the GA fractal". Despite all of this research, there has yet to be any practical techniques proposed for these problems in general. In this paper, we introduce a new technique for improving the genetic algorithm's performance on a class of difficult, rugged fitness landscape, order-based problems. We call this technique *chromosome reduction*.

The paper is organized as follows: Section 2 provides the motivation for chromosome reduction. Section 3 describes chromosome reduction in detail. Section 4 gives a theoretical analysis of chromosome reduction. Section 5 summarizes experimental results comparing chromosome reduction with several standard techniques on a representative problem. Conclusions are given in Section 6.

2 Motivation for Chromosome Reduction

Some of the most difficult problems for a genetic algorithm include those which have a rugged fitness landscape, and problems which are order-based. These problems are discussed below. The schema theorem is re-examined to obtain clues for developing techniques to improve the GAs performance.

2.1 Rugged Fitness Landscapes

The idea of a *fitness landscape* was first introduced by Wright [20]. It is a mapping from the genomes in a population of individuals to their respective fitness values. It also refers to the visualization of that mapping. Recently, there has been a renewed interest in the analysis of fitness landscapes because of the theoretical and practical insights which may be gained into the difficulty of problems. In fact, the difficulty of a problem is often directly proportional to the *ruggedness* of its fitness landscape. Problems with a smooth fitness landscape, such as ones with a single large basin, are quite easy for the GA. Other problems, which have

*Research supported by OCAST Grant AR2-004 and Sun Microsystems, Inc.

rugged fitness landscapes marked by numerous local optima, degenerate the GAs adaptive search into a random walk.

The renewed interest in fitness landscapes was spawned by Kauffman [11, 12] with his ideas on *tunable ruggedness.* He defined a class of landscapes called NK landscapes where the genomes consist of binary strings with fixed length N. The contribution of each bit to the fitness of the chromosome is based on its own value as well as the value of K other bits. When K = 1, the landscape is smooth with a single large basin. As K approaches N, the landscape becomes progressively more rugged. When K = N, the landscape becomes completely rugged, causing the GAs performance to approach that of a random search.

Kauffman's work was extended by Weinberger [18, 19] in an analysis of several NK landscape features. Weinberger's work was applied to several combinatorial problems: by Stadler and Schnabl [17] on the Traveling Salesman Problem, by Stadler and Happel [16] on the Graph Bi-Partitioning Problem, and by Stadler [15] on the Matching Problem.

2.2 Order-based Problems

Order-based problems are another difficult area for the GA. The goal of these problems is to find an optimal ordering of a set of distinct items. Traditional binary representations and operators often cannot be used in these problems. Thus, one must use an integer permutation representation coupled with custom designed, order-based operators. Unfortunately, this approach offers little in the way of tools for theoretical analysis of the operation of the GA or of the ruggedness of the fitness landscape.

2.3 Clues from the Schema Theorem

The schema theorem has been used to describe the growth and decline of schemata from generation to generation. Goldberg [8] showed the growth and decline of schemata in a maximizing genetic algorithm using simple one-point crossover and simple mutation to be governed by the following equation:

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} \left[1 - p_c \frac{\delta(H)}{l-1} - p_m o(H) \right]. \tag{1}$$

In the above equation, $m(H, t)$ denotes the number of schema H in the population at time t, and $m(H, t+1)$ denotes the expected number of schema H in the population at time $t+1$. A measure of the relative fitness of schema H is given by the ratio $\frac{f(H)}{\bar{f}}$, where $f(H)$ denotes the average fitness of the chromosomes containing schema H and $\bar{f}$ denotes the average fitness of the entire population calculated using

$$\bar{f} = \frac{\sum_{i=1}^{n} f_i}{n}.$$

The remainder of Equation 1 reduces the expected growth of schema H by the net disruption caused by crossover $p_c \frac{\delta(H)}{l-1}$ and by mutation $p_m o(H)$. In general, Equation 1 can be simplified to the following form:

$$m(H, t+1) = m(H, t) f_r(H) p_s(H). \tag{2}$$

Here, $f_r(H)$ denotes the relative fitness of schema H and $p_s(H)$ denotes the survival probability of schema H through recombination.

Consider the beneficial schema H_b. Suppose that H_b is highly fit relative to the current population, that is $f_r(H_b) \geq 1$. Thus, one would expect H_b to grow in the population. The growth of H_b in this case is governed by Equation 2:

$$m(H_b, t+1) = m(H_b, t) f_r(H_b) p_s(H_b).$$

Since $f_r(H_b) \geq 1$, the expected number of schema H_b in the population will grow as long as recombination is not excessively disruptive. That is, when $f_r(H_b) p_s(H_b) < 1$, then recombination is sufficiently disruptive and will result in the negative growth of H_b.

Problems which are easy for the GA rarely require the use of disruptive recombination operators. Problems which do require disruptive recombination operators include those with rugged fitness landscapes and order-based problems. In particular, recombination in order-based problems is inherently disruptive because of the interrelationships between the genes similar to those described in Kauffman's NK-landscapes. That is, it is difficult, if not impossible, to characterize the distance in the fitness landscape between any two instances. Each instance actually has N adjacent neighbors.

Consequently, for the GA to be effective on order-based problems and problems with rugged fitness landscapes, it must ensure that beneficial genetic material survives a disruptive recombination process. The ultimate goal of the GA is for beneficial schemata to grow and harmful or nonbeneficial schemata to decline. Ideally, the GA should obey the following alternative form of the schema theorem:

$$m(H,t+1) = \begin{cases} m(H,t)f_r(H) & \text{if } H \text{ is beneficial} \\ m(H,t)f_r(H)p_s(H) & \text{otherwise} \end{cases} \tag{3}$$

This means that for each beneficial schema, H_b, the survival probability $p_s(H_b)$ must be one. That is, the beneficial schemata in each chromosome is preserved while the nonbeneficial schemata are disrupted.

3 Chromosome Reduction

Chromosome reduction is a heuristic technique developed to augment the GA in order to obey the variant of the schema theorem given by Equation 3. Chromosome reduction is a member of a class of heuristic techniques described by Corcoran and Wainwright [5]. The goal of each of these techniques is to identify sequences of genes in the chromosome which correspond to beneficial schemata and preserve these sequences through recombination. Chromosome reduction uses a *sliding window* to identify the sequences and *reduction* to preserve the sequences through recombination. Summarized below is the process as described by Corcoran and Wainwright [3, 5].

The first part of chromosome reduction is the identification of sequences of genes which correspond to beneficial schemata. This is done through the use of a sliding window. As shown in Figure 1, the sliding window is composed of left and right pointers which define a sequence within the chromosome. The sliding window is initialized with the left and right pointers positioned at the first gene in the chromosome. This defines a sequence with a single gene. This sequence is evaluated to determine if it corresponds to beneficial schemata. The simplest method for evaluating the sequence is to have a user defined function. A more general approach might be to use an intelligent rule base or statistics obtained over the run of the GA to evaluate the sequence. If the sequence is beneficial, then reduction is used to preserve it. Otherwise, there are two other possible cases: the sequence is too short to be beneficial or it is too long to be beneficial. When the sequence is too short, the right endpoint is incremented to make a longer sequence. When the sequence is too long, the left endpoint is incremented to make a shorter sequence. In this way, the window is slid across the chromosome in search of beneficial sequences.

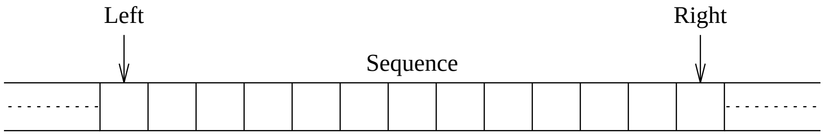

Figure 1: Sliding window

After beneficial sequences have been identified, the next step is to ensure that they are preserved through crossover. This is done through the use of reduction, or partitioning the chromosome into beneficial and nonbeneficial genetic material. Figure 2 illustrates reduction of the chromosome. The reduction boundary indicates the partition between the beneficial and nonbeneficial genetic material. The genetic material on the left is a collection of beneficial sequences which together make up an optimal subsolution. In Figure 2a, a beneficial sequence has been identified in the nonbeneficial partition by the sliding window. In Figure 2b, the beneficial sequence has been moved into the beneficial partition and the reduction boundary has been adjusted appropriately. Once a chromosome has been reduced, the GA ensures the material in the beneficial partition is not disturbed. The GA is free to recombine genetic material in the remainder of the chromosome without regard to the magnitude of the disruption, since the remaining genetic material is known to be nonbeneficial in its current order.

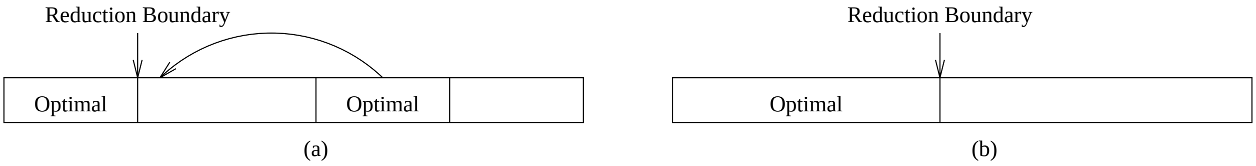

Figure 2: Chromosome reduction

As the reduction boundary approaches the right side of the chromosome, the quality of the solution increases and the amount of work necessary to recombine the remaining material decreases. When the boundary reaches the right side of the chromosome, the global optimal has been found. Even when the right side is never reached, the reduction boundary gives a measurable indication of the quality of the solution. Note, it is not always the case that beneficial sequences represent optimal subsolutions. In fact, some fuzziness can and perhaps should be introduced in the definition of a beneficial sequence. In many cases, it may be advantageous for the fuzziness to be adaptive. Whatever the case, reduction preserves the desired beneficial material through recombination and focuses the GA on the remainder of the chromosome. This not only reduces the time and effort required by the GA but also reduces or prunes the size of the remaining search space. That is, reduction can lead to a quicker search and better solutions. However, it is important to use common sense in the definition of beneficial sequences to avoid premature convergence.

4 Theoretical Analysis

In this section we introduce the Bin Packing Problem as a representative difficult problem for the GA. We survey the fitness landscape and analyse the expected and actual changes made by chromosome reduction. We use exhaustive enumeration of small problems to show the effectiveness of chromosome reduction.

4.1 Bin Packing: A Representative Problem

The Bin Packing Problem is a representative problem which is both order-based and has a rugged fitness landscape. The bin packing problem can be briefly stated as follows:

> Given a finite collection $U = \{u_1, u_2, ..., u_n\}$ of items, a positive integer size $s(u)$ for each $u \in U$, a positive integer bin capacity B, and a positive integer K, is there a partition of U into disjoint sets $U_1, U_2, ..., U_K$ such that the sum of the sizes of the items in each U_i is B or less?

The bin packing problem has been shown to be NP-complete in the strong sense [6]. Consequently, approximation methods have been developed which provide near-optimal solutions.

The three best known approximation methods for the bin packing problem are Next Fit, First Fit, and Best Fit. Next Fit starts with a single empty bin and places each item into the bin in turn from the list of items. When the placement of an item into the bin exceeds the capacity of the bin, the item is placed into a new bin. Packing resumes with this new bin and the remaining unpacked items are never placed into any of the old bins. First Fit begins by placing items from the list into a single empty bin. When an item exceeds the capacity of the bin, it is placed into a new bin. The placement for each of the remaining items is determined by scanning each of the bins from the first bin to the last bin, until a bin is found with enough space for the item. That is, each item is placed in the first bin into which it will fit. A new bin is created only when an item cannot be placed in any bin. Best Fit is nearly identical to First Fit except it places each item into the bin with the amount of empty space closest to the item's size. For one dimensional bin packing, these methods have been shown to have improved results if the list of items are first sorted by nonincreasing value.

Approximation methods for the bin packing problem have been shown to have impressive performance bounds. Garey and Johnson [7] and Johnson *et al.* [10] show bounds for the one dimensional problem to be as good as 1.2 times optimal. Coffman *et al.* [1] show bounds for the two dimensional problem to be as good as 1.7 times optimal using a simple technique, and 1.5 times optimal for a more complex method.

Li and Cheng [13] present a complex heuristic for the three dimensional problem with bounds of 3.25 times optimal.

Genetic algorithms have been used to obtain improvements over the approximation methods. Smith [14] used a GA on a two dimensional problem using a bottom left packing strategy. Corcoran and Wainwright [4] used a GA on a two dimensional problem using a level oriented strategy. Corcoran and Wainwright [2] also applied a GA to a three dimensional problem. However, as Corcoran and Wainwright [4] report, the GA fails to find the optimal on a contrived two dimensional problem with as few as 75 packages. The GAs inability to find the optimal on small and medium sized problems leads one to expect poor performance on larger problems in which the sampling efficiency is less.

4.2 Analysis of the Fitness Landscape

It is difficult to characterize the landscape of the bin packing problem. The bin packing problem is order-based and uses an integer permutation encoding scheme. Given a list of n packages, each package has $n-1$ possible packages with which it can be swapped. This corresponds to $n-1$ adjacent permuted lists. This is very similar to Kauffman's NK landscape with K close to N. Thus, this landscape is very rugged.

Consider for example a randomly generated collection of 6 packages with respective sizes of $\{6, 2, 4, 6, 10, 2\}$, and a bin capacity of 10. If this list were packed using next fit, 4 bins would be required. New permutations can be generated by swapping any package with any of the 5 other packages. One can quite easily rearrange the packages, for example $\{6, 4, 6, 2, 2, 10\}$, to find an optimal packing requiring only 3 bins. It is also quite simple in this small case to find a packing, such as $\{2, 10, 2, 4, 6, 6\}$, which represents the worst possible packing, requiring 5 bins. Thus, the set of possible packings with this set of packages range from 3 to 5 bins. With 6 packages, there is 6! or 720 possible orderings of the packages, which is small enough to enumerate exhaustively.

Figure 3 illustrates the fitness landscape for the 6 package data set. For each of the 720 permutations, the fitness (or number of bins required to pack each permutation) is plotted. Figure 3 clearly shows a majority of the possible solutions to require 4 bins. There are also several optimal 3 bin solutions and relatively few 5 bin solutions. Considering there are only three possible fitness values, this fitness landscape is clearly very rugged and the GA would have a hard time outperforming random search.

The bin packing problem is especially well suited for chromosome reduction. Packages in a bin with no wasted space can be rearranged without affecting the result. For example, the sequence $\{6, 2, 2\}$ fills a single bin with no wasted space. This sequence can be packed in any order with the same result: $\{6, 2, 2\}$, $\{2, 6, 2\}$, $\{2, 2, 6\}$, etc. In addition, optimal sequences can be combined in any order and the combined subsolution is also optimal. For example, $\{\{6, 4\}, \{6, 2, 2\}\}$, $\{\{6, 2, 2\}, \{6, 4\}\}$, etc. The advantage of reduction in bin packing is the possibility of discovering optimal sequences which without reduction fall across a bin boundary. For example, the list $\{6, 2, 4, 6, 10, 2\}$ has an optimal sequence $\{4, 6\}$ resulting in $\{\{4, 6\}, 6, 2, 10, 2\}$. A second optimal sequence $\{10\}$ can also be preserved resulting in $\{\{4, 6\}, \{10\}, 6, 2, 2\}$. The remainder of the chromosome is also reducible, resulting in $\{\{4, 6\}, \{10\}, \{6, 2, 2\}\}$, which is an optimal packing. Thus, in this case chromosome reduction has found an optimal solution where one might never have been found before.

Figure 4 illustrates the fitness landscape for the 6 package data set after reduction. This is a striking difference from the landscape before reduction. There are no longer any packings requiring 5 bins and the overwhelming majority of solutions are optimal. It is clear that reduction has smoothed out the fitness landscape in this case and has shifted the average number of bins required closer to the optimum.

4.3 Enumerative Analysis

Landscape analysis provides an intuitive picture of the power of chromosome reduction. However, an enumerative analysis offers more insights.

As stated above, the list $\{6, 2, 4, 6, 10, 2\}$ has 720 possible permutations. When packed using next fit and a bin size of 10, the number of required bins ranges from 3 to 5 bins. The number of solutions in each class was found to obey the following cost distribution:

```
[3]:###########.......................................: 144 (20%)
[4]:#######################################...........: 544 (75.5556%)
[5]:###...............................................: 32 (4.44444%)
```

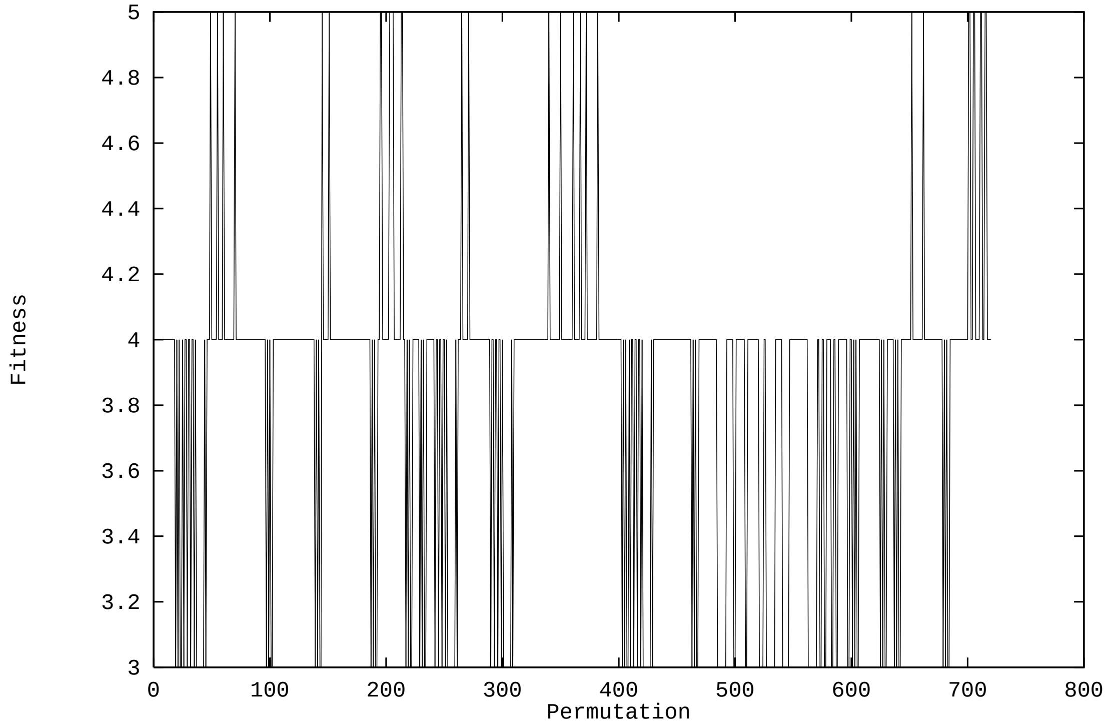

Figure 3: Fitness Landscape without Reduction

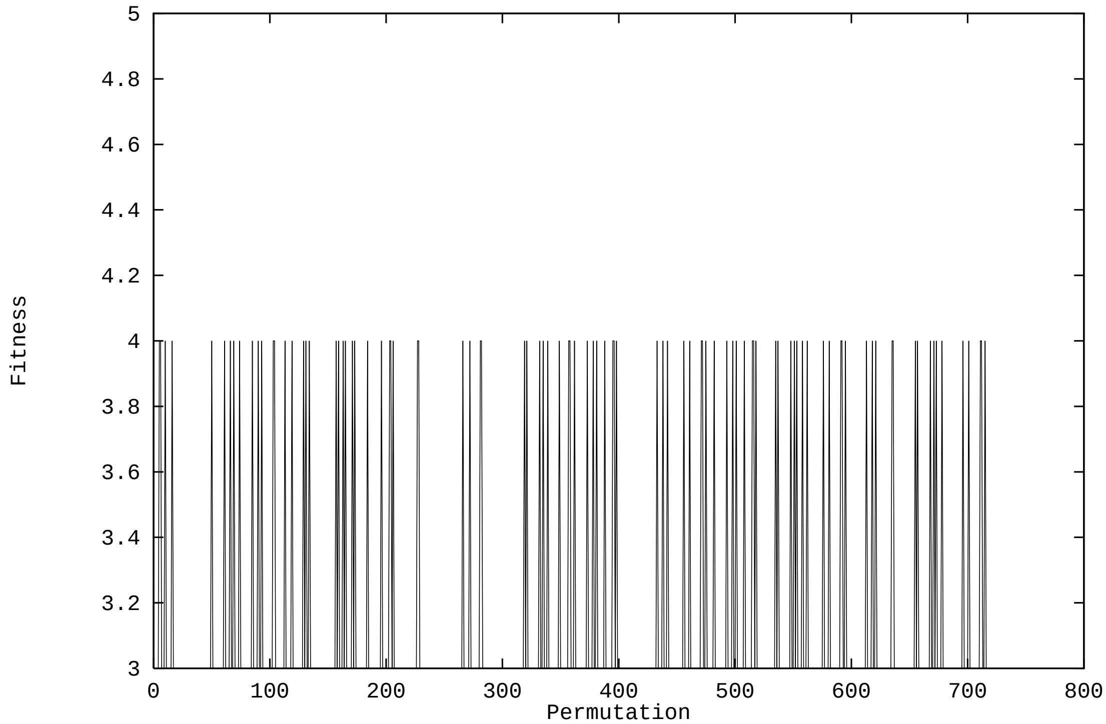

Figure 4: Fitness Landscape with Reduction

The majority of the solutions, 544 out of 720 or 76%, require 4 bins. There are 144 optimal solutions, representing 20% of the total number. Only 32 solutions or 4% require 5 bins. The average cost for this data set is 3.84 bins and the standard deviation is 0.469.

Using reduction, the solutions ranged from 3 to 4 bins with the following cost distribution:

```
[3]:###############################################......: 624 (86.6667%)
[4]:#######.........................................: 96 (13.3333%)
```

The overwhelming majority of the cases, 624 out of 720 or 87%, are optimal with 3 bins. There are 96 solutions or 13% which require 4 bins. This distribution has an average cost of 3.13 bins and a standard deviation of 0.340.

Number of	Average Cost	
Packages	Without Reduction	With Reduction
6	3.84	3.13
7	4.82	4.21
8	5.30	5.04
9	6.01	5.48
10	6.22	5.51
11	6.35	5.61

Table 1: Average cost for various one dimensional data sets

Table 1 shows the average costs with and without reduction for various sized data sets using enumeration. Problem sizes greater than 11 are too time consuming to enumerate. Table 1 clearly shows how chromosome reduction remains effective as the problem size is increased. Each of the data sets has the same pattern of cost distribution behavior as the original 6 package data set. For example, the 11 package data set has the following distribution without reduction:

```
[5]:##..............................................: 836352 (2.09524%)
[6]:##################################..................: 24843744 (62.2388%)
[7]:###################...............................: 13825440 (34.6356%)
[8]:#...............................................: 411264 (1.0303%)
```

After reduction, the cost distribution was as follows:

```
[5]:######################..........................: 17319648 (43.3894%)
[6]:############################......................: 20775864 (52.0479%)
[7]:###.............................................: 1818120 (4.55477%)
[8]:#...............................................: 3168 (0.00793651%)
```

Note the dramatic shift towards the optimal solution.

Number of	Average Cost	
Packages	Without Reduction	With Reduction
6	15.15	14.69
7	17.11	16.56
8	23.61	22.56
9	30.64	29.79
10	38.69	37.99
11	40.99	40.41

Table 2: Average cost for various two dimensional data sets

Table 2 shows the average costs with and without reduction for a two dimensional bin packing problem. In this case, the fitness is obtained by using a level-oriented next fit on a single open-ended bin. Table 2 clearly indicates that chromosome reduction is just as effective on the two dimensional problem as it is for the one dimensional problem. For example, the two dimensional 11 package data set has the following distribution without reduction:

```
[38]:#######..........................................: 4959360 (12.4242%)
[39]:##...............................................: 1209600 (3.0303%)
[40]:##################...............................: 13685760 (34.2857%)
[41]:####.............................................: 2452320 (6.14358%)
[42]:###############..................................: 10939680 (27.4062%)
[43]:##...............................................: 1123200 (2.81385%)
[44]:#######..........................................: 5114880 (12.8139%)
[46]:#................................................: 432000 (1.08225%)
```

After reduction, the cost distribution is as follows:

```
[38]:##########.......................................: 7233600 (18.1217%)
[39]:###..............................................: 1689600 (4.2328%)
[40]:#####################............................: 16072320 (40.2646%)
[41]:####.............................................: 2745600 (6.87831%)
[42]:############.....................................: 9039360 (22.6455%)
[43]:#................................................: 681120 (1.70635%)
[44]:####.............................................: 2455200 (6.15079%)
```

Note, the two dimensional problem has a non-Gaussian cost distribution which should make it harder for the GA to find the optimal. Despite this fact, chromosome reduction is still successful in improving the average cost and shifting the cost closer to the optimum.

5 Experimental Analysis

Theoretical analysis shows how chromosome reduction is effective on small problems and how it is expected to perform on larger ones. To illustrate the effectiveness of chromosome reduction, we ran experimental tests with the contrived 75 package data set of Corcoran and Wainwright [4]. This data set is composed of 75 two-dimensional packages which can be optimally packed in a single open-ended bin using a level-oriented packing strategy. The optimal packing height for this data set is 60.

Test runs revealed that the contrived 75 package data set has at least 46 distinct fitness solution values in the cost distribution. These range from the optimal value of 60 to the worst observed value of 105.

	Fitness				Average
Crossover Method	Best	Average	Variance	Std. Dev.	Generations
Asexual with Reduction	60	60.00	0.000	0.000	107.00
Asexual	64	66.90	2.322	1.524	145.60
PMX	65	67.00	2.000	1.414	118.50
Cycle	65	67.40	4.267	2.066	120.60
UOX	66	67.60	3.378	1.838	199.00
Order1	67	67.90	1.211	1.101	138.10

Table 3: Results for contrived 75 package data set (Optimal = 60)

Table 3 summarizes the results of running chromosome reduction on the contrived 75 package data set. The columns in the table indicate the best, average, variance and standard deviation of 10 independent runs of the GA and the average number of generations when the best solution was found in each run. The first row shows the results of the GA augmented with chromosome reduction using asexual crossover. Asexual crossover is simply a swap of two randomly selected genes in the chromosome. Note, with chromosome reduction the optimal was found in all 10 runs. The remaining rows indicate the results using various crossover operators *without* chromosome reduction. The best performing operator in this case was asexual crossover, with an average packing height of 66.90 and a best packing height of 64. Partially Mapped

Crossover (PMX), Cycle Crossover, Uniform Order Crossover (UOX) and Order1 Crossover had averages of 67.00, 67.40, 67.60, and 67.90, respectively. Chromosome reduction clearly is the winner as far as fitness is concerned, and it also is a winner with respect to the average number of generations required to find the result. Chromosome reduction used an average of 107.00 generations while its closest competitor, PMX, needed 118.50. Note, the best value present in the initial pool was 83 for the standard crossover methods and was 81 for asexual crossover with reduction.

	Fitness				Average
Crossover Method	Best	Average	Variance	Std. Dev.	Generations
Asexual with Reduction	71	73.50	2.500	1.581	130.50
Asexual	75	76.20	1.067	1.033	174.10
Cycle	75	76.40	1.600	1.265	121.10
Order1	74	76.50	1.833	1.354	150.80
PMX	75	77.00	2.222	1.491	107.10
UOX	76	77.20	1.511	1.229	202.00

Table 4: Results for random 75 package data set (Optimal $\geq$ 69)

Table 4 summarizes the results of running chromosome reduction on a random 75 package data set. This data set consists of 75 randomly generated two-dimensional packages. The optimal packing for this data set is unknown, however, by the area of the packages the optimal can be no better than 69. Table 4 again shows asexual crossover with chromosome reduction to be the best technique with average performance over ten independent runs of 73.50 and a best case of 71. Asexual crossover was the best of the operators without using chromosome reduction with average performance of 76.20. The average performance for cycle, order1, PMX, and UOX was 76.40, 76.50, 77.00, and 77.20, respectively. While chromosome reduction showed clear superiority to the other methods with respect to fitness, the average number of generations required was more competitive. Chromosome reduction required an average of 130.50 generations. PMX and cycle outperformed reduction with averages of 107.10 and 121.10, respectively. However, considering the quality of the fitness results, the extra number of generations used by chromosome reduction was well worth the effort. These results clearly show chromosome reduction to be a highly effective technique for improving the GAs performance on the bin packing and related problems.

6 Conclusions

In this paper, we have described how problems with rugged fitness landscapes and problems which are order-based are among the most difficult for the GA. We have demonstrated that the recombination operators that are needed for these types of problems often are sufficiently disruptive to hinder the growth of beneficial schemata. We offer chromosome reduction as one of a class of heuristic techniques that can ensure the survival of beneficial schemata through a disruptive recombination. We presented a theoretical analysis of chromosome reduction using the bin packing problem as a representative problem for illustration. Analysis of the fitness landscapes showed chromosome reduction to have a smoothing effect with a shift toward better solutions. Enumerative analysis on small cases quantified this dramatic shift and verified the continuing effectiveness with each combinatorial increase in problem size up to 11 packages. Experimental results showed continued effectiveness on much larger data sets. In conclusion, chromosome reduction shows great promise as a general heuristic technique for improving the GAs performance on a class of difficult problems. Chromosome reduction enables the GA to obtain better solutions while at the same time often requiring fewer function evaluations.

Acknowledgements

This research has been supported by OCAST Grant AR2-004. The authors also wish to acknowledge the support of Sun Microsystems, Inc.

References

[1] E. G. Coffman, M. R. Garey, D. S. Johnson, and R. E. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal of Computing*, 9(4):808–826, Nov. 1980.

[2] A. L. Corcoran and R. L. Wainwright. A genetic algorithm for packing in three dimensions. In H. Berghel, E. Deaton, G. Hedrick, D. Roach, and R. Wainwright, editors, *Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing*, pages 1021–1030, New York, 1992. ACM Press.

[3] A. L. Corcoran and R. L. Wainwright. A heuristic for improved genetic bin packing. Technical Report UTULSA-MCS-93-8, The University of Tulsa, May 1993.

[4] A. L. Corcoran and R. L. Wainwright. LibGA: A user-friendly workbench for order-based genetic algorithm research. In E. Deaton, K. M. George, H. Berghel, and G. Hedrick, editors, *Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing*, pages 111–118, New York, 1993. ACM Press.

[5] A. L. Corcoran and R. L. Wainwright. Reducing disruption of superior building blocks in genetic algorithms. Technical Report UTULSA-MCS-93-1-1, The University of Tulsa, Jan. 1993.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, San Francisco, 1979.

[7] M. R. Garey and D. S. Johnson. Approximation algorithms for bin packing problems: A survey. In G. Ausiello and M. Lucertini, editors, *Analysis and Design of Algorithms in Combinatorial Optimization*, pages 147–172. Springer-Verlag, New York, 1981.

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

[9] J. H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, Michigan, 1975.

[10] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal of Computing*, 3(4):299–325, Dec. 1974.

[11] S. A. Kauffman. Adaptation on rugged fitness landscapes. In *Lectures in the Sciences of Complexity: Volume I in the Santa Fe Institute Studies in the Sciences of Complexity.* Addison-Wesley, Redwood City, California, 1989.

[12] S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, New York, 1993.

[13] K. Li and K. H. Cheng. On three-dimensional packing. *SIAM Journal of Computing*, 19(5):847–867, Oct. 1990.

[14] D. Smith. Bin packing with adaptive search. In J. J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 202–207, Hillsdale, New Jersey, 1985. Lawrence Erlbaum.

[15] P. F. Stadler. Correlation in landscapes of combinatorial optimization problems. *Europhys. Lett.*, 20(6):479–482, 1992.

[16] P. F. Stadler and R. Happel. Correlation structure of the landscape of the graph-bipartitioning problem. *J. Phys. A: Math. Gen.*, 25:3103–3110, 1992.

[17] P. F. Stadler and W. Schnabl. The landscape of the travelling salesman problem. *Physics Letters A*, 161:337–344, 1992.

[18] E. D. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biol. Cybern.*, 63:325–336, 1990.

[19] E. D. Weinberger. Local properties of Kauffman's N-k model: A tunably rugged energy landscape. *Pysical Rev. A*, 44(6399), 1991.

[20] S. Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proceedings of the Sixth International Congress on Genetics*, volume 1, pages 356–366, 1932.