

Identifying Common Structures in Solving 3D Bin Packing with Genetic Algorithm

Jiradej Ponsawat¹, Chatchawit Aporntewan², and Prabhas Chongstitvatana³

Department of Computer Engineering, Faculty of Engineering, Khon Kaen University, jiradej@kku.ac.th¹

Department of Mathematics, Faculty of Science, Chulalongkorn University, chatchawit.a@chula.ac.th²

Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, prabhas@chula.ac.th³

ABSTRACT

Recently, it is shown that genetic algorithms perform optimization by identifying and composing common structures of above-average solutions. The quantity of building blocks is important for the success of GAs. This paper aims to identify the quantity of building blocks in 3D bin packing problem. A new method for measuring the quantity of building blocks is proposed. A solution can be encoded into a binary string by many different coding. Choosing a good coding is very problematic for users who are not GA experts. The results show that the quantities of building blocks are significantly different according to how the solutions are encoded. The coding that gives high quantity of building blocks yields better average fitness of solutions. As a result, we can spend a little time to predict the efficiency of a large number of coding by measuring the quantity of building blocks. If the quantity of building blocks is high, executing GAs, otherwise, turning to another optimized algorithm.

Keywords: Building Block Identification, Bin Packing, Genetic Algorithm

1. INTRODUCTION

Genetic algorithm (GA) is a problem solving method that is robust for a wide range of applications. One of the explanations for the success of GA relies on the building-block hypothesis [1,2]. It states that GA discovers common structures of solutions or building blocks (BBs). Modern GAs explicitly identify the BBs [3,4]. The solution recombination is performed by a genetic operator in order to mix the BBs. It is strongly believed that the solution quality can be improved by mixing the building blocks.

The quantity of building blocks is important for the success of GAs. This paper aims to identify the quantity of building blocks in real-world applications. Three-dimensional bin packing is chosen to represent a difficult problem. A new method for measuring the quantity of BBs is proposed. A solution can be encoded into a binary string by many different coding. Choosing a good coding is very problematic for users who are not GA experts. The encoding of the candidate determines, to a large extent, whether building blocks exist in this problem. An intuitive idea behind solving a bin packing problem is that grouping or pairing suitable items together, subject to

the constraint of the available space, will generally achieve a good packing.

2. EXPERIMENT SETTING

The experiment is separated into two parts. First, the measurement of quantity of building blocks without executing GA since the quantity of BBs might be affected by many GA parameters. Therefore, the GA parameters are discarded except the coding of solutions. We want to sort the coding by their quantities of BBs. In the second part, we execute GA with each coding in order to see the effectiveness of GA according to the coding. In this case, we cannot avoid the interference of GA parameters which sometimes perfectly exploit the BBs and sometimes disrupt the BBs. Unfortunately, it is still not clear how those parameters interact with each other.

2.1 Building Block Measurement without GA

The 3D bin packing is defined as follows. We have to pack 10 boxes in a container of which the cross-section area is 100x100. The container length is unlimited. The shorter packing length is the better solution. A solution defines a permutation of 10 boxes. A packing algorithm starts with an empty container. A box is fed to the algorithm. Next, the algorithm places the box in the container and so on. We fix the packing algorithm [5]. Coding represents the permutation of 0 to 9. The first coding concatenates each 4-bit binary number together. The permutation, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, is encoded as 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001. The second coding represents a permutation by a 10x10 matrix. The matrix elements are either 0 or 1. If the matrix element in row i and column j is one, box j is fed to the packing algorithm immediately after box i . For example, the permutation, 7, 8, 2, 4, 6, 5, 1, 3, 0, 9 is represented by the following matrix.

```
0123456789
0 X000000001
1 0X01000000
2 00X0100000
3 100X000000
4 0000X01000
5 01000X0000
6 000001X000
7 0000000X10
8 00100000X0
9 000000000X
```

There are $10! = 3,628,800$ possible solutions. A set of 10,000 solutions is randomized from the solutions of which the length of packing is less than a constant L . Smaller L results in better solutions. For each L , the quantity of BBs is measured by *counting the number of edges* in Bayesian network. To construct the acyclic Bayesian network, we use the Win Mine Toolkit [6] version 2.0. The number of nodes in the network is equal to the number of bits of a solution. The edges indicate the dependence between a pair of nodes. The common structures are some bits that repeat more often than that of a random data. As a result, the dependence between those bits can be recognized, and the algorithm that constructs the Bayesian network put a number of edges to connect those bits (or nodes).

2.2 Building Block Measurement on GA Result

Instead of randomize a set of solutions we execute GA and make a snapshot every generation. The GA parameters are set as follows. Population size is set at 2000. The crossover operator is Partial Matched Crossover (PMX) [1] which always produces valid solutions. The crossover and mutation rates are set at 0.9 and 0.0 respectively. PMX is the operator that combines two permutations, and results in two new permutations that partially inherit from the former permutations. For example, the PMX operates on the permutations $P1 = 2, 4, 6, 5, 1, 0$ and $P2 = 0, 6, 4, 1, 2, 5$. Two cut points are randomly selected every time performing PMX. Supposed the cut points are 2 and 4. In both $P1$ and $P2$, 6 is swapped with 4 and 5 is swapped with 1.

P1	2	4	6	5	1	0
P1	0	6	4	1	2	5
C1	2	6	4	1	5	0
C2	0	4	6	5	2	1

To perform PMX with the first encoding, we change each chunk of four bits to a decimal number. For example, 0000 0001 0010 0011 0100 is changed to 0, 1, 2, 3, 4. For the second encoding, the matrix PMX, exchanging the sequence of order is achieved by, first, swapping the rows and then the columns. An example will illustrate the operation. Fig. 1 shows the crossover region to be at row 4 to row 6. To cross row 4, $P1$ has a 1 in column 6, $P2$ has a 1 in column 1, so the swapping will be between 6 and 1. First, row 6 is swapped with row 1 in $P1$, then column 6 is swapped with column 1. The same operation is done in $P2$.

P1: 7 8 2 4 6 5 1 3 0 9
P2: 5 6 0 3 9 8 2 4 1 7

P1		P2	
X000000001		X001000000	
0X01000000		0X00000100	
00X0100000		00X0100000	
100X000000		000X000001	
0000X01000	6	0100X00000	1
01000X0000	1	00000X1000	6
000001X000	5	100000X000	0
00000000X0		00000000X0	
00100000X0		00100000X0	
000000000X		000000001X	

C1: 7 8 2 4 6 0 1 3 5 9
C2: 0 6 5 3 9 8 2 4 1 7

Fig.1: Partial Matched Crossover for Matrix Encoding. Parents are $P1$ and $P2$. The offspring are $C1$ and $C2$.

2.3 Continuity Measurement

We have presented a method for measuring the quantity of BBs. We developed another measurement for predicting the success of GA. It is named "the continuity of BBs." Roughly speaking, the continuity is the similarity of BBs between two set of solutions, $S1$ and $S2$, defined by two thresholds $L1$ and $L2$, $L1 > L2$. Usually, $|L1 - L2|$ is small. Let $G1$ and $G2$ are Bayesian networks according to $S1$ and $S2$. To measure the continuity, we count the number of edges that appear in both $G1$ and $G2$. The continuity plots are shown in Fig. 6. If the continuity is smooth, it is very likely that GA is able to evolve from $S1$ to $S2$. Otherwise, it is difficulty to carry out $S2$ by identifying and mixing BBs in $S1$.

3. RESULTS

The experiment is carried out to solve 3D bin packing problem. The results are presented for measuring the quantity of BBs and the continuity for two cases, with and without GA.

3.1 Quantity of Building Block without GA

Fig.2 shows the plot of the quantity of BBs for the population with two encodings: Encoding1 and Encoding2. Encoding1 is the first encoding which encodes a solution as a permutation of the box numbers. Encoding2 is the second encoding which encodes a solution as a matrix as described in section 2.1. The horizontal axis represents the *quality* of the solution (the best is at the right most). The vertical axis represents the *quantity* of BBs as the normalized number of links. The quantity of BBs in the better quality population with Encoding1 tends to increase as the quality of the solution is increased. For Encoding2, the trend is in the opposite direction. The quantity of BBs is decreasing as the quality of the solution is increasing.

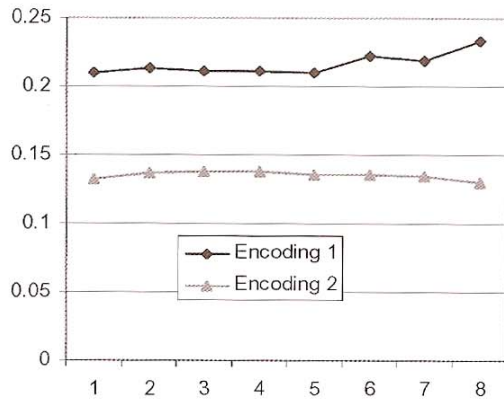


Fig.2: Normalized Number of Link without GA

The difference between the quantity of BBs at the beginning (quality 1) and at the end (quality 8) indicates the change. At the beginning the measurement is taken from the whole population, this signifies the quantity of BBs when the population is random. At the end, the sampling consists of mostly the high quality solution hence the quantity of BBs is at its maximum.

This trend shows that in terms of the quantity of BBs, the Encoding1 is better than the Encoding2. This difference explains the difference in the ability to find solutions when both encodings are used in running the Genetic Algorithm in the next experiment.

3.2 Quantity of Building Block on GA Result

The measurement of the quantity of BBs is taken while running Genetic Algorithm to solve the problem. The quality of a solution is measured by its packing length; the shorter packing length is the better solution. Using the population size of 2000, both encodings find very good solutions very early (Fig.3). However, the average length of Encoding1 decreases rapidly while the average length of Encoding2 fluctuates around a constant and is much higher than the average length of Encoding1.

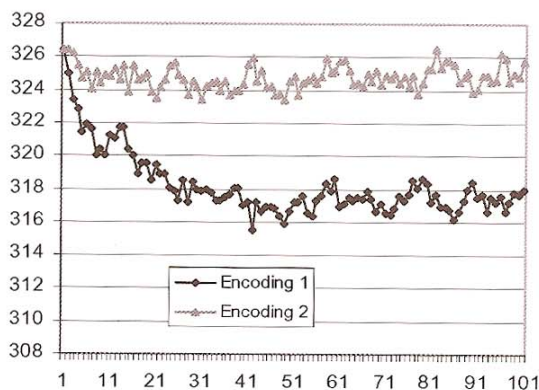


Fig.3: Average Packing Length from Each Generation

Fig.4 shows the plot of the quantity of BBs versus the generation. The quantity of BBs for Encoding1 tends to increase while the quantity of BBs for Encoding2 is constant and tends toward decreasing. The observation is consistent with the result in Section 3.1 (Fig. 2).

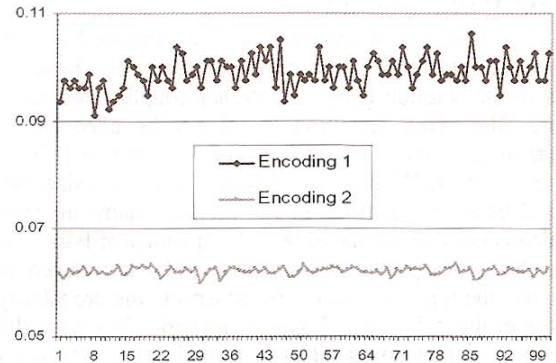


Fig.4: Normalized Number of Link on GA Result

3.3 Quantity of Continuity from GA result

The continuity of BBs is measured by the similarity of Bayesian network between the solutions of the adjacent generations as explained in Section 2.3. Fig.5 shows Encoding2 has a much higher fluctuation of continuity of BBs more than Encoding1. This can be interpreted as BBs in Encoding1 is more related between generations than BBs in Encoding2. It is interesting to note that although the measure of quantity of BBs of Encoding2 is rather smooth (Fig. 4), its continuity is much fluctuated. This indicates that mixing BBs in Encoding2 is not effective in improving solutions.

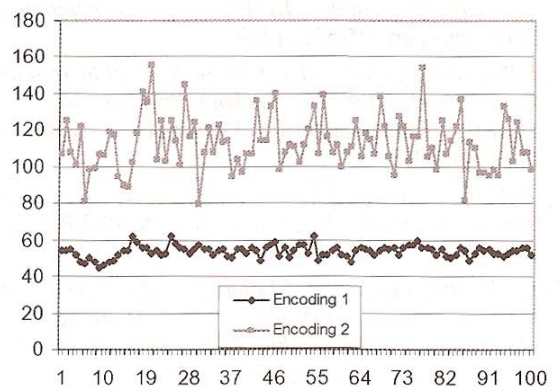


Fig.5: Continuity of Both Encoding

4. CONCLUSIONS

This work presented two measurements of Building Blocks based on the number of edge of Bayesian network building from set of solutions. The proposed measurements are used in running Genetic Algorithm to

solve a real-world problem: 3D bin packing. The difference in encoding can be observed from the measurement of BBs and it can explain the difference of the success in solving the problem. The Encoding1 has higher quantity of BBs and also has more success in solving the problem.

5. RECOMENDATION

The measure of quantity of BBs proposed in this work is the measure of the degree of “coupling” between bits in the solution [7,8]. The higher coupling results in more BBs. This definition might not be adequate in explaining many phenomena. For example, Trap functions or ADF might have BBs of a certain size, but the degree of coupling might include many artifacts which is not the true measure of the quantity of BBs. It is possible that another measure should be introduced to classify the type of coupling by observing the probability value of the link in the Bayesian network. This work did not discuss how to use BBs in Genetic Algorithms, for example, the crossover operator can be more effective by avoiding disruption of BBs.

5. REFERENCES

- [1] D.E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison Wesley, Reading, MA, 1989.
- [2] D.E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer Academic Publisher, Boston, MA, 2002.
- [3] M. Pelikan and D.E. Goldberg, “Hierarchical BOA solves Ising Spin Glasses and MAXSAT,” *Proceeding of Genetic and Evolutionary Computation Conference*, pp.1271-1282, 2003.
- [4] M. Pelikan, D.E. Goldberg and E. Cantú-Paz, “BOA: The Bayesian Optimization Algorithm,” *Proceeding of Genetic and Evolutionary Computation Conference*, pp.525-532, 1999.
- [5] J. Ponsawat, *Genetic Algorithm with Local Search for Three Dimensional Bin Packing Problem*, M. Eng. Thesis, Chulalongkorn University, Bangkok, Thailand, 2003.
- [6] D.M. Chickering, *The WinMine Toolkit*. Technical Report, Microsoft, Redmond, WA, MSR-TR-2002-103, 2002.
- [7] C. Apornwewan and P. Chongstitvatana, “Chi-square Matrix: an Approach for Building-block Identification,” *Proceeding of 9th Asian Computing Science Conference*, pp.63-77, 2004.
- [8] C. Apornwewan and P. Chongstitvatana, “A Quantitative Approach for Validating the Building-block Hypothesis,” *Proceeding of IEEE Congress on Evolutionary Computation*, pp.1403-1409, 2005.