Iranian Journal of Science & Technology, Transaction B, Vol. 26, No. B4
Printed in Islamic Republic of Iran, 2002

"Research Note"

A GENETIC-NEURO ALGORITHM FOR TILING PROBLEMS WITH ROTATION AND/OR REFLECTION OF FIGURES*

R. MONSEFI[1**] AND M. A. TOOSI[2]

[1]Dept. of Computer Science, Ferdowsi University, Mashhad, I. R. of Iran
[2]Dept. of Mathematics, Sabzevar Teacher Training University, Sabzevar, I. R. of Iran

Abstract -This paper describes an algorithm for tiling with polyminoes that consider rotation and/or reflection of figures in the steps of 90°. First, we review the previous parallel algorithms for tiling problems. Next, we propose a hybrid approach that is based on genetic algorithms (GA) and artificial neural networks (ANN). In this approach, the production of new members in GA and their evaluation are performed by a Hopfield neural network. Finally we compare our method with the previous works, and show that our method can produce global minima for many problems. The algorithm can be used for solving a variety of 2D-packing problems.

Keywords– Tiling problem, artificial neural network, genetic algorithms, simulated annealing (SA)

1. INTRODUCTION

The tiling problem is to pack a checkerboard (bin) with small pieces. In this study, we consider a special case of its kind, i.e. tiling with polyminoes. Each polymino is a right angled polygon that the length of each edge is a multiple of some predefined unit length. Tiling problem can be considered as a special case of bin packing. Bin packing and cutting stock are popular among optimization problems.

A few parallel algorithms have been reported on these problems. Firstly, two parallel algorithms, based on the modified McCulloch-Pitts neuron model, have been reported [1, 2]. These algorithms are applicable for solving a 7×7 checkerboard tiling problem with 10 polyminoes. The problem is to find a single solution among 1.3×10^{14} possible candidates where there exists one and only one solution in the problem without rotation and reflection. If rotation and reflection is considered, the size of solution space would grow to 1.7×10^{19}. Figure1 shows the solution of the problem found in [1] and [2].

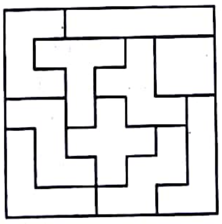


Fig. 1. The only solution for a 7×7 tiling problem without rotation and reflection

*Received by the editors November 2, 1999 and in final revised form October 23, 2001
**Corresponding author

Pargas & Jain [3] proposed a stochastic approach for 2D-bin packing. This technique is similar to those used in GA or in SA algorithms. It is suitable for bin packing, but doesn't do well for tiling problems. In their algorithm, rotation is considered, but it doesn't reach the global minima.

Due to the usage of their terminology in the following section, here we briefly explain their work.

A solution is a structure with the following format:

$$[(f_1,o_1),(f_2,o_2),...,(f_N,o_N),L]$$

where f_i , $0 \le f_i \le N-1$ represents the i^{th} of N figures, and o_j , $0 \le o_j \le 3$ is the figure's current orientation in steps of 90°, and L is the length of the solution (distance between the left edge of the bin and the right edge of the rightmost figure after the figures are placed as can be seen in Fig.4). A bin has a predefined height H. As an optimization problem, L is the objective function that must be minimized.

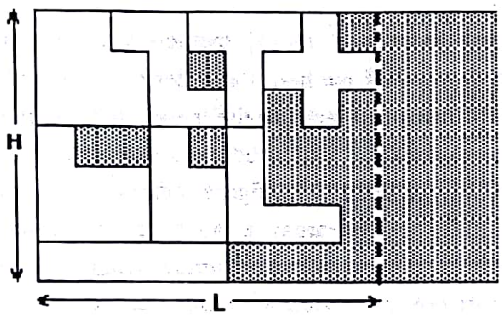


Fig. 4. A bin with height H and solution length L for a hypothetical problem

The population is initialized randomly. Evaluation of each solution produces its length L. Selection of the solutions is performed using a linearly biased random number generator. The effect is that the probability of selecting a solution is linearly proportional to its relative position in the population.

In each iteration, a solution S is selected, with probability p≈0.9. Fifty of its neighbors are randomly generated and evaluated in search for a better solution. Two solutions S_1 and S_2 namely

$$S_1=[(f_1,o_1),(f_2,o_2),...,(f_N,o_N),L_1]$$

$$S_2=[(F_1,O_1),(F_2,O_2),...,(F_N,O_N),L_2]$$

are defined to be neighbors if the sequence in which their figures are placed are identical except for exactly two figures, i.e. if $f_i = F_i$ for all but the two values of i.

With the remaining probability 1-p≈0.1, a new solution is randomly generated and evaluated. The purpose of generating a solution randomly is to introduce new permutations possibly very different from those present in the population in order to prevent the population from converging prematurely.

The evaluation of a solution S involves assigning a location in the bin to each figure such that no two figures overlap. The evaluation algorithm employs a two-dimensional bit array of height H and arbitrary length. In the evaluation algorithm, we attempt to place the figure with the prescribed orientation, and with its leftmost uppermost square unit on the first empty cell. If unsuccessful, we try figure by rotating 90°. If again unsuccessful, we try figure by rotating 180°. If finally unsuccessful, we try figure by rotating 270° Otherwise, we try next empty cell until the figure is successfully placed. Scanning the cells of bin in search of the next empty cell is done by a simple linear scan in column major order. Finally, convergence occurs when all solutions in the population have the same length. Running our simulation, this algorithm can not find any solution for the Asai *et. al.* examples illustrated in Fig.2 with rotation and reflection of figures except for example No.3 shown in Diagram 1.

3. PROPOSED APPROACH

As mentioned earlier, methods based on ANN don't consider rotation and reflection of figures but they produce good results in finding global minima. In addition, they are independent of the order of figures in contrast with GA. On the other hand, GAs can act with rotation and reflection of figures, but doesn't guarantee to find the global minima. Here, we try to take advantage of a combination of the above two methods to find a better solution with rotation and reflection.

In this hybrid approach, we combine the Pargas & Jain and Asai *et. al.* methods in the following manner:

We replace the 'evaluate procedure' in [3] method with the method, i.e. we evaluate the solutions with ANN. Give that in ANN method, the order of figures in solution is not important in contrast to the Pargas & Jain [3], we must redefine the 'neighborhood of a solution' as follows:

The two solutions S_1 and S_2 with previous structure are defined to be neighbors if the orientation of their figures is identical except for exactly one figure, i.e. if $o_i = O_i$ for all but only one value of i.

One difficulty that occurs is that GA (Pargas & Jain [3] method) works with the length more than the minimum length, but ANN works with exact minimum length, and the 'L' in GA is the solution length. To combine them, either GA must work with minimum length, or ANN must work with more than minimum length. In this study, we chose the first case, and considered energy function value that reported with ANN as fitness function. Less energy function means a better solution.

Another difficulty is the low speed of ANN. For this difficulty, we initially try to replace the floating-point operations in Asai *et. al.* [1] method with integer operations. But the quality of the result was so poor that was abandoned. Further, we changed the Asai *et. Al.* [1] method in the following order:

Instead of allowing the figures to compete with each other to the end of the process, we allow only a few competing iterations after which a figure with the highest score is selected and set in its place permanently. The remaining figures compete in the same manner. This number of iteration is called Inner Iteration Count (IIC). We run many simulations to find an appropriate value for IIC such that it is an acceptable solution in a reasonable time. The value of 5 for IIC seemed to be appropriate.

This modification also eliminated the disadvantage of the ANN methods for identical figures previously commented. Furthermore, the patterns produced with ANN are examined. The 'unfit figures' are removed until all the blank spaces are continuous. We define an 'unfit figure' as a figure that sum of its blank spaces around it divided by its surface is maximum in comparison to other figures.

To achieve better performance, we make a change in Pargas & Jain algorithm [3] in order to prevent the population from converging prematurely, generated a new random solution in 10% of the total number of iterations. The weakness of this approach is that in their method when a new random solution is generated, it is not usually as fit as current population members, and cannot win in competing with other members. Thus, it cannot add to population. For this problem, we use a concept used in operating systems namely "multiple level feedback on round robin". It is one of the methods for process scheduling [6]. In this method, a new process arrives, and a few time slices is run to catch up with other processes. Likewise, we try to make these new members fitter. Simply, during the generation of a new solution, we examine a few of its neighbors three times. In each round, the best of

Secondly, Pargas & Jain [3] proposed a stochastic approach, based on GA and simulated annealing (SA), for 2D-bin packing. Their algorithm cannot reach the global minima. Among global optimization methods, GAs has shown good results for many optimization problems. It was showed that, the number of schemata which are effectively being processed in each generation is of the order n^3 where n is the population size.

In this study, we introduce a hybrid approach by trying to achieve a better solution with the aid of the above two methods.

2. PREVIOUS PARALLEL WORKS

Here we try to give a more detailed description of previous works as follows:

Takefuji & Lee [2] used a Hopfield ANN with optimization approach to solve this problem without rotation or reflection of figures. They defined an energy function that its value equals to zero in solution state. As an optimization problem, energy function should be minimized. Hopfield & Tank [5] reached the global minima by choosing the appropriate parameter in all of the runs for the 7×7 problem shown in Fig. 1. Asai *et al* [1]. proposed a modified version of Takefuji & Lee [2] approach. They added a new term called fitting violation function to the energy function proposed by [2] and used an analog neural network array [1]. Their energy function is given by

$$E=\frac{A}{2}\sum_{i=1}^{l}\left(\sum_{q=1}^{m}\sum_{r=1}^{n}V_{iqr}-1\right)^2+\frac{B}{2}\sum_{i=1}^{l}\sum_{j=1}^{m}\sum_{k=1}^{n}f(i)V_{ijk}{}^2+\frac{C}{2}\sum_{i=1}^{l}\sum_{j=1}^{m}\sum_{k=1}^{n}g(i)V_{ijk}{}^2+\frac{D}{2}\sum_{i=1}^{l}\sum_{j=1}^{m}\sum_{k=1}^{n}V_{ijk}(1-V_{ijk})$$

where *m, n* are the width and length of the checkerboard, *l* is the number of figures, *f(i)* is overlap violation function and *g(i)* is fitting violation function for i^{th} polymino. A, B, C and D are adjustable parameters. Their method found the global minima for five examples illustrated in Fig.2.

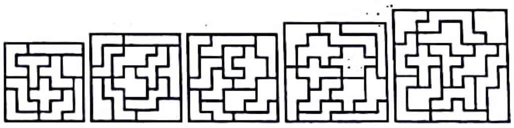
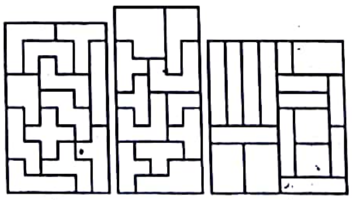


Fig. 2. Examples used in reference [1]

In the above two methods, rotation and reflection of figures were not applicable. Due to the disability of considering rotation and reflection of figures with this method, the correct orientation of figures must be known beforehand, i.e. we must know the solution prior to problem solving. However, in some problems due to the existence of patterns, rotation or reflection are not allowed, but in general it may be considered.

In addition, one disadvantage of the above methods is that they can not be applied to the problems that have more than one figure with the same shape and orientation. Examples No. 7 and 8 in Fig. 3, suffer from this difficulty as well. Our simulation results showed that the mentioned methods do not reach the global minima for these sorts of problem.



6 7 8

Fig. 3. Examples used to demonstrate incapability of reaching global minima for [1] or [2]

these neighbors is selected for the next round. With this modification, our simulation's result showed a better performance.

Based on the above remarks on hybrid approach, we combined the GA and ANN methods together and achieved the following results shown in Table.1 for the five Asai *et. al.* [1] examples of Fig.2 and the three additional examples in Fig.3.

Table.1. Eight problems and percentages to reach global minima with hybrid approach without back tracking

Example No.	Problem Size	Percentage to Reach Global Minima
1	7×7×10	50
2	8×8×11	10
3	8×8×12	30
4	9×9×12	0
5	10×10×14	0
6	10×6×12	0
7	11×5×11	60
8	9×8×17	100

The above results can even be improved further. In fact, in many cases, the algorithm gets close to the global minima, but never reaches it. In these cases a sequential search method such as the employment of back tracking (BT) can be adequate and efficient. Since such methods require a lot of processing time, we must restrict BT with small catches spaces. In this work, if the number of remaining figures are less than or equal to 4, it is suggested that the solution be near the global minima. Then the solution is passed to BT for testing to see whether it can reach the global minima or not. The final result is shown in Table 2.

Our simulations run on a PC with CELERON 300MHz CPU and the population size was chosen to be 100. It should be commended, however, that the existence of a zero on example No. 5 on Table.2, means that the algorithm couldn't reach its global minima. Generally for those problems that our hybrid algorithm couldn't reach its global minima, algorithm still produced a relatively good result. A relatively good result means that algorithm settles in local minima and not the global minima. In our terms, it means that it is able to place all the figures on the bin except perhaps one, but it is not accepted for tiling problem because it requires only the global minima. We were not able to show whether these results are close to the global minima. Because it may be a local minima and far from global minima.

Diagram1 illustrates a comparison between the proposed hybrid approach and the Pargas & Jain method [3].

Table 2. Eight problems, average time taken, and percentage to reach to global minima in the proposed approach

Example No.	Average Time (min)	Percentage to Reach Global Minima
1	1.4	100
2	6.5	40
3	4	100
4	15	20
5	33	0
6	4	90
7	1.2	100
8	0.1	100

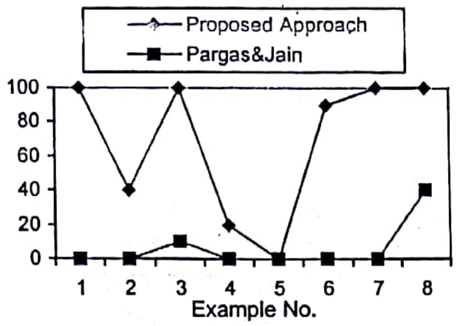


Diagram 1. Comparison of Pargas & Jain [3] method with our proposed hybrid approach for 8 examples against percentage to reach to global minima

As it can be seen in the above diagram, our hybrid approach achieved better results compared to [3] method.

4. ADVANTAGES AND DISADVANTAGES

The advantages of our hybrid approach are as follows:

1. Good performance finding the global minima in contrast to the Pargas & Jain [3] method.
2. The ability to work with problems that contain more than one identical figure, in contrast to the Asai *et. al.* [1] method.
3. The ability to work with problems, whose figures are not rectangular, in contrast to the usual methods that solve these sorts of problem.
4. Likewise Pargas & Jain [3], the ability to distribute work on multi-processors, in contrast to the Asai *et. al.* [1] method and the usual algorithms for these sorts of problem.
5. The ability to consider rotation and reflection of figures, in contrast to many present algorithms.
6. The parallel competing of figures to get the best place that it fits in.

The disadvantages are as follows:

1. It must be approximated by experiment, and a theoretical approach does not exist.
2. Due to the use of 3D-Neural Array, the time, and space complexity of algorithm is large. Thus, this method does not do well for a large problem.
3. Disability to work with problems whose figures are not polyminoes unless it can be approximated to polyminoes.

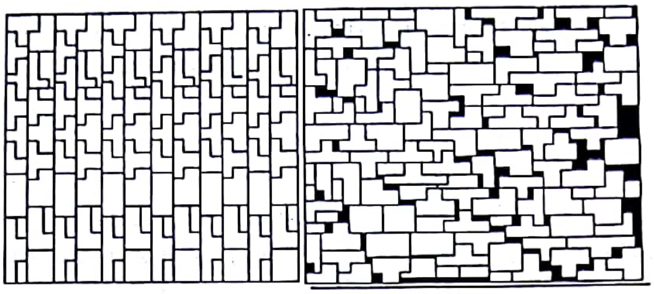
For the last two disadvantages, a remark can be pointed out as follows:

For those large problems that can be divided into smaller equal portions, this hybrid approach can also do well. One instance of these sorts of problem is packing the required figures of N equal objects into a large sheet. Example No. 2 of [3] is provided. This example and Pargas & Jain's result are shown in Table 3:

Table 3. Example No. 2 in [3]

Num Figs.	Pop'n Size	Time (min)	Opt.Len	Best Len	Util (%)
126	200	63	37	40	92

Note that the unit of time reported is in minute on a multi-processor system with 32 processors.
The best solution and their optimum solution is illustrated in Fig.5.



(a) Best Len: 37 (b) Opt Len: 40

Fig. 5. Example No.2 in [3]

As can be seen, the problem can be divided into six small portions. Pargas & Jain [3] don't report the exact length of figures, thus we consider a simple approximation of this problem as follows:

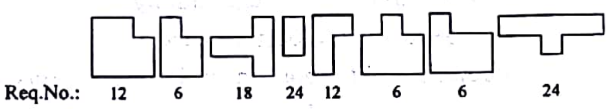


Fig. 6. An approximation for example No. 2. in [3]

In Fig.6, Req. No. is the number of identical figures requested. The smallest figure is 2×1. It assumes that the dimension of bin is 18×30.

By dividing the requested number of each figure into its great common divisor, i.e. 6, we obtain six small problems that could be solved. Appropriate dimensions for small problems can be 18×5 or 9×10. Fig. 7 illustrates five solutions for 18×5 problem.

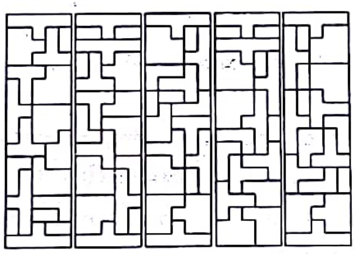


Fig. 7. A few solutions for 18×5 problem in Fig. 6

With the repetition of solution to each small portion, one can get a solution for the original problem. The average time required for solving each portion of the problem with our proposed approach was 3.8 minute in a PC with CELERON 300 MHz CPU.

For some problems whose figures are non-polymino, it can be approximated with polyminoes. Figure 8 illustrates the fitting of word "**Tiling**" in a 6×6 checkerboard.

Fig. 8. Tiling the word "**Tiling**"

5. CONCLUDING REMARKS AND FUTURE WORKS

In this study an approach that integrates artificial neural networks and genetic algorithms for tiling with polyminoes has been discussed. ANNs can reach its global minima, but doesn't allow rotation or reflection of figures. On the other hand, GAs can consider rotation and reflection but do not reach its global minima. To take advantage of the above two methods, both are combined in the following manner:

ANN produces different patterns, depending on the orientation of figures. Each pattern would be a member of GA. In producing each pattern, all of the figures compete with each other to get the best place that can fit each other and edges of the bin. After a few competing iterations, the figure with the highest score is selected and set in its place permanently. The remaining figures compete in the same manner. This pattern is examined and the unfit figures are removed until all the blank spaces are continuous. If the number of unplaced figures were less than a pre-specified number, the pattern would be tested with a BT algorithm. Then, if it doesn't find the global minima, we can add the pattern to GA's population. As GA forwards, population converges to optimal solution.

In many simulation runs performed, a good result was achieved. The following points can be followed in future works:

1. Due to the lack of a theoretical approach for determining ANN adjustable parameters, one may further study this to provide a better overall performance.
2. Is there a mixed non-rectangular shape for portions of large problems to be solved?
3. Is there a systematic approach for dividing large problems into appropriate small portions?

Acknowledgment- The authors wish to thank the anonymous referees for their valuable suggestions and comments.

REFERENCES

1. Asai, H., Nakayama, T. and Ninomya, H., Tiling algorithm with fitting violation function for analog neural array, Proc. Int. Conf. Neural Network, p. 565 (1996).
2. Takefuji, Y. and Lee, K. C., A parallel algorithm for tiling problems, *IEEE Trans. Neural Networks*, **1**, No.1, p. 143 (1990).
3. Pargas, R. P. and Jain, R., A parallel stochastic optimization algorithm for solving 2D bin packing problems, in *Proc. Int. Conf. Neural Network*, p. 18 (1993).
4. Beasley, D., Bull, D. R. and Martin, R. R., An overview of genetic algorithms: Part 1, Fundamentals, *University Computing*, **15(2)**, p. 58 (1993).
5. Hopfield, J. J. and Tank, D. W., Neural computation of decisions in optimization problems, *Biol. Cybern.*, **52**, p. 141 (1985).
6. Madnick, S. E. and Donovan, J. J., Operating Systems, McGraw-Hill (1978).