A Distributed Stock Cutting using Mean Field Annealing and Genetic Algorithm

Chuleui Hong, *Member, KIMICS*

Abstract— **The composite stock cutting problem is defined as allocating rectangular and irregular patterns onto a large composite stock sheet of finite dimensions in such a way that the resulting scrap will be minimized. In this paper, we introduce a novel approach to hybrid optimization algorithm called MGA in MPI (Message Passing Interface) environments. The proposed MGA combines the benefit of rapid convergence property of Mean Field Annealing and the effective genetic operations. This paper also proposes the efficient data structures for pattern related information.**

Index Terms— **Genetic Algorithm, Mean Field Annealing, Optimizations, Parallel and Distributed Computing, Stock Cutting**

I. INTRODUCTION

The stock cutting problem is one of common combinatorial optimization problems in material related industries [1, 2]. In this problem, rectangular and irregular patterns are allocated onto a large stock sheet of finite dimensions in such a way that the resulting scrap will be minimized. This problem is common to many applications in aerospace, shipbuilding, VLSI design, steel construction, and shoe manufacturing. This problem is commonly known as the oriented 2D bin packing problem. Since this problem and associated theoretical problems are NP-Complete [3], there is no polynomial algorithm to solve them. Thus, we return to heuristics which produce near optimal solutions with expensive run time cost. Linear programming, dynamic programming and tree-search methods have been proposed to solve this problem [4,5]. Fig. 1 shows the sample placement.

This paper adopts Bottom-up Left-justified (BL) algorithm because BL algorithm can be implemented with genetic algorithm. BL algorithm represents the order of patterns to be packed as a list. This method places the pattern to the bottom as down as possible and moves it to the left (Fig. 2). It repeats BL operation to all patterns in the order of a list. It is proven that the height of the solution of BL algorithm is less than the three times of optimal height [6]. Since the ordered list of BL algorithm represents the all possible placements, it can be viewed as a chromosome in genetic algorithm. The genetic operations are applied for finding the optimal state which has the minimal height [7,8].

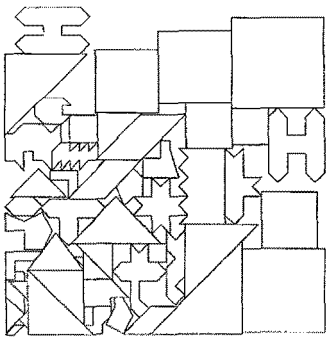

Fig. 1. An example of composite stock cutting.

However BL algorithm has been developed for rectangular patterns. So in this paper, first we make the bounding box including each irregular pattern. Next, after finding an optimal list by using the proposed MGA algorithm, spaces among patterns are removed.

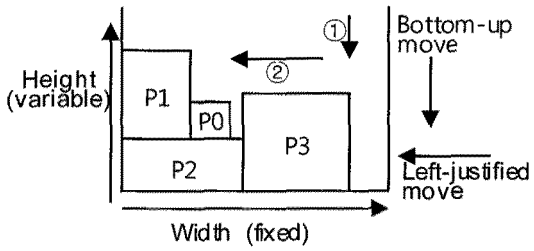

Fig. 2. Bottom-up Left-justified (BL) algorithm.

The proposed Mean Field Genetic Algorithm (MGA) is a hybrid algorithm based on mean field annealing (MFA) and genetic algorithm (GA) [9,10]. MFA has the characteristics of rapid convergence to the equilibrium state while the simulated annealing (SA) [11] takes long time to reach the equilibrium state. In the proposed method, the typical genetic algorithm is modified in such a way that the evolved new states are accepted by the Metropolis criteria as in simulated annealing. This modified Simulate annealing-like Genetic Algorithm is called SGA. The simulation results show that the new MGA is better than MFA and GA alone.

In this paper, we propose a distributed hybrid algorithm for the stock cutting problems. This algorithm is verified practical in the solution quality and computation time.

Manuscript received December 28, 2009; revised January 5, 2010; accepted January 16, 2010.

Chuleui Hong is with the Department of Computer Science, Sangmyung University, Seoul, 110-743, Korea (Email: hongch@smu.ac.kr)

II. DATA STRUCTURES FOR COMPOSITE STOCK CUTTING

In this section we present how to implement the data structure for the proposed method. Before the MGA process, a bitmap is generated for each pattern at all allowed rotations. The affinity relation is calculated for every pair of patterns in advance. Since the stock sheet is composite material, the allowable rotations are predefined to 2 (0 and 180 degree) or 4 (0, 90, 180, and 270 degree). A bitmap representation is used for the patterns because the overlap of two patterns can be detected easily by examining the bitmap elements. The affinity relation between any pair of patterns displays the contribution to the packing when they are placed next.

For the bitmap of a pattern, first the bounding box or Minimum Bounding Rectangle (MBR) is determined for the pattern, then the outside, the boundary and the inside elements of bitmap are different each other for easy calculation of affinity relation and overlap detection. The bounding box is defined as the minimal rectangle which contains the pattern (Fig. 3).

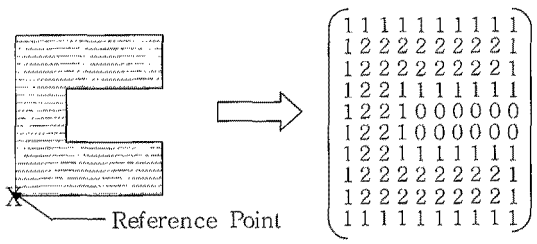

Fig. 3. The Example of Bitmap for a pattern.

Packing density increases generally as the boundaries of two patterns overlap more and the area of bounding box including two patterns is smaller. The affinity relation is defined as follows for arbitrary two patterns i and j.

$$a_{ij} = \alpha\left(a_1 + \frac{a_2}{2}\right) + \beta\rho \qquad (1)$$

The variable a_1 and a_2 are for the edgewise adjacency of two patterns as in Fig. 4, where α and β are weight constants. The minimal bounding box surrounding two patterns cannot be guaranteed by only the edgewise adjacency (Fig. 5). When two patterns that form the minimal bounding box fit together, the size of the final stock sheet is likely to be small. Therefore the density of the bounding box surrounding two patterns, ρ, is incorporated to the affinity relation.

The affinity relation for every pair of patterns is calculated on all allowed rotations in advance. We use the stored value instead of calculating it on every state transition in MGA. This reduces the computing time dramatically.

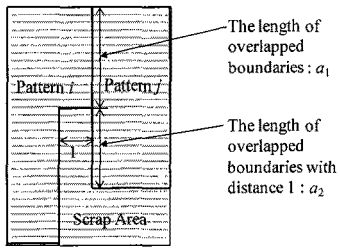

Fig. 4. The Affinity Relation.

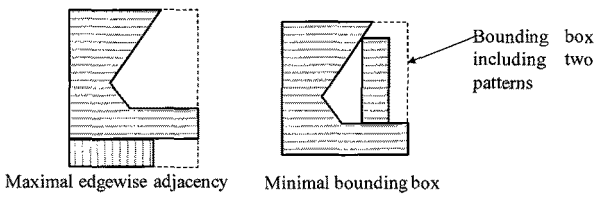

Fig. 5. Edgewise adjacency and bounding box affect the packing density.

III. DISTRIBUTED MEAN FIELD GENETIC ALGORITHM

The mean field annealing (MFA) is derived from simulated annealing (SA) based on mean field approximation method in physics [11]. Though SA provides the best solutions for various kinds of general optimization problems, it takes long time to achieve those because SA changes the states randomly. But MFA makes the system reach the equilibrium state very fast using the mean value estimated by mean field approximation while the solution quality is not better than that of SA.

Genetic algorithm (GA) is a powerful heuristic search method based on an analogy to the biological evolution model. GA has the powerful and diverse operations such as selection, reproduction, crossover, and mutation comparing with other heuristics. Therefore it provides an effective means for global optimization in a complex search space such as NP-complete optimization including mapping problem, function optimization and etc.

The proposed algorithm takes benefits of MFA and GA. Since genetic algorithm does not have a concept of temperature, genetic operations break the convergence property of mean field annealing. In the proposed method, the typical genetic algorithm is modified where the evolved new states are accepted by the Metropolis criteria as in simulated annealing. The modified Simulate annealing-like Genetic Algorithm is called SGA. SGA can keep the convergence property of MFA operations at each temperature.

In MFA, a spin matrix is used to represent the solution state. First, MFA is applied on a spin matrix to reach the thermal equilibrium fast. After the thermal equilibrium is reached, the population for GA is made according to the distribution of pattern sequences in the

spin matrix. Next, GA operations are applied on the population while keeping the thermal equilibrium by transiting to the new state with Metropolis criteria. MFA and GA are applied by turns until the system freeze.

The solution state is represented by the cost function, C, and its minimal value is the optimal solution. Since MGA uses BL algorithm and the patterns are placed without overlaps between adjacent patterns. The cost function is made up of the affinity relation, the distance and their probabilities.

$$C = \sum_{i=0}^{N-1}\sum_{j\neq i}\sum_{p=0}^{N-1}\sum_{q\neq p} s_{ip}\, s_{jq}\frac{d_{ij}}{a_{ij}} \qquad (2)$$

N : The number of patterns

s_{ip} : The probability of pattern i in the p_{th} place in BL operations

s_{jq} : The probability of pattern j in the q_{th} place in BL operations

a_{ij} : The affinity relation between pattern i and pattern j

d_{ij} : The distance between pattern i and pattern j

3.1 Distributed Mean Field Annealing (MFA)

The mean field annealing (MFA) is derived from simulated annealing (SA) based on mean field approximation method in physics [10]. While SA changes the states randomly, MFA makes the system reach the equilibrium state very fast using the mean value estimated by mean field approximation.

A spin matrix consists of N rows representing patterns and N columns representing their places in BL operations. The value of spin element (i,p), s_{ip}, is the probability of placing pattern i to the p^{th} order in BL operations. So, the range of s_{ip} is $0 \le s_{ip} \le 1$ and the sum of each row is 1. The initial value of s_{ip} is $1/N$ and s_{ip} converges to 0 or 1 as the algorithm reaches to the final optimal solution state. $s_{ip} = 1$ means that pattern i is placed to the p^{th} order in BL algorithm. TABLE I displays the initial and final optimal solution spin matrix of Figure 2.

TABLE I
THE SPIN MATRIX OF FIG. 2.

	0	1	2	3
0	1/4	1/4	1/4	1/4
1	1/4	1/4	1/4	1/4
2	1/4	1/4	1/4	1/4
3	1/4	1/4	1/4	1/4

(a) The Initial State

	0	1	2	3
0	0	0	0	1
1	0	0	1	0
2	1	0	0	0
3	0	1	0	0

(b) The Solution State

The mean field, ϕ_{ip}, is defined as follows from the cost function of equation (2).

$$\phi_{ip} = -\frac{\partial C}{\partial s_{ip}} = -\sum_{j\neq i}^{N-1}\sum_{q\neq p}^{N-1} s_{jq}\frac{d_{ij}}{a_{ij}} \qquad (3)$$

ϕ_{ip} denotes decrease of cost when the pattern i places to the p^{th} order in BL algorithm or a string.

Since each spin value, s_{ip}, is proportional to $e^{\phi ip/T}$, we can normalize s_{ip} as follows where T is the temperature.

$$S_{ip} = \frac{e^{\phi_{ip}/T}}{\sum_{q=0}^{N-1} e^{\phi_{iq}/T}} \qquad (4)$$

Since cost, C, is proportional to s_{ip} linearly, the cost change, ΔC, is also proportional to Δs_{ip} which is the change of s_{ip}.

$$\Delta C = \sum_{p=0}^{N-1}\Delta C_{ip} = \sum_{p=0}^{N-1}\phi_{ip}\Delta S_{ip} \qquad (5)$$

where $\Delta S_{ip} = S_{ip}^{new} - S_{ip}^{old}$.

For making the sequential MFA be distributed, the $N\times N$ spin matrix is partitioned column-wise such that each column or a group of columns in a spin matrix is assigned to a node in MPI (Message Passing Interface). When pattern-i is selected at random in a particular iteration, all nodes calculate mean fields of assigned columns and adds them to get the new spin values, which is the i^{th} row, for pattern-i. After computing the cost change due to spin changes using global sum operations in MPI, each node updates the spin values of assigned columns.

The pseudo code for the distributed mean field annealing of each node is as follows.

< Distributed Mean Field Annealing>

while *ΔC is less than ε for continuous N iterations*

Select a same pattern-i at random by using same seed

Compute the local mean field

$$\phi_{ip} = -\sum_{j\neq i}^{N-1}\sum_{q\neq p}^{N-1} s_{jq}\frac{d_{ij}}{a_{ij}} \quad \text{for } 0 \le p \le N-1$$

Compute the new spin values at the i^{th} row by using global sum operation

$$S_{ip}^{new} = \frac{e^{\phi_{ip}/T}}{\sum_{q=0}^{N-1} e^{\phi_{iq}/T}}$$

Compute the cost change due to spin changes by using global sum operation

$$\Delta C = \sum_{p=0}^{N-1}\phi_{ip}(s_{ip}^{new} - s_{ip})$$

Update the spin values at the i^{th} row

$s_{ip} = s_{ip}^{new}$ for $0 \le p \le N-1$

Perform global collect for spin values in the i^{th} row

In implementing MFA, the cooling schedule has a great effect on the solution quality. Therefore the cooling schedule must be chosen carefully according to the characteristics of problem and cost function. The Markov chain length at a certain temperature is the number of state transition to reach the equilibrium state. The Markov chain length is set to the number of state transitions where the cost change is less than ε=0.5 for continuous N annealing process.

3.2 Distributed Simulated annealing-like Genetic Algorithm (SGA)

We modified GA such that the new evolved state is accepted with a Metropolis criterion like simulated annealing in order to keep the thermal equilibrium of MFA.

$$\Pr[\Delta C \text{ is accepted }] = \min\left(1, \exp\left(-\frac{\Delta C}{T}\right)\right) \qquad (6)$$

In the above equation, T is the current temperature. ΔC is the cost change of new state from old state. It is made by subtracting the cost of new state from that of old one. The modified GA is called SGA.

The individual of SGA is made up of a string in the sequence of patterns placing to the sheet in BL algorithm. For example, a string, "3,2,0,1", means that patterns are placed to a sheet in the order of pattern 3, 2, 0, and 1.

The individuals are generated randomly with the probability as same as that of spin matrix in MFA. For example, if spin values of an arbitrary pattern i, which is the elements of i^{th} row, is 0.2, 0.4, 0.1, 0.1, 0.2, an individual is made such that the i^{th} character in a string can be 0 with a probability of 0.2, 1 with that of 0.4, 2 with that of 0.1, 3 with that of 0.1 and so on.

In the experiment, the population size is set to $10\times N$. Therefore the size of subpopulation size in each node is $10\times N/P$ where N is the number of patterns and P is the number of nodes in MPI.

The linear cost function is chosen as same as that of MFA. In our synchronous distributed SGA, each node generates subpopulation randomly from the MFA's spin matrix. And then the subpopulation and its fitness value are broadcast to all other nodes and they form the global population.

Next, the individuals are selected as many as the size of subpopulation from the global population randomly. A proportionate selection scheme is used. The individuals are chosen randomly with a ratio of their costs over sum of population's cost. Each node executes the sequential genetic algorithm in parallel. Independent genetic operations are implemented and evaluated in its subpopulation.

The duration of isolated evolution is called one *epoch* and the *epoch length* is the number of predefined generations for a node before synchronizing communication among the nodes. The *epoch length* is set to the $10\times N/P$. *max_epoch* is the number of synchronous communications. It is set to P.

For crossover operation, after 2 individuals are chosen by turns from population, the part of string is exchanged. So the patterns' orders are exchanged. The size of exchanged part is randomly set less than 1/4 of string length. The probability of crossover is 0.8.

An individual is selected with a probability of 0.05 in mutation. Exchange and displacement operation is implemented on the selected individual. The probability of exchange is 0.1 and displacement is 0.9.

To keep best individual, the individual with the lowest cost is kept in the next population. This prevents genetic operations from losing the best solution of the previous stage.

The pseudo code for the distributed SGA of each node is as follows.

```
<Distributed SGA>
Initialize subpopulation(P_sub) from MFA spin matrix
for iteration is less than max_epoch
  Calculate fitness for P_sub
  for generations = 1 until epoch_length
    Select individuals from subpopulation
    Reproduce next population
    for select 2 individuals by turns
      Perform crossover with probability of
        crossover
      Calculate the cost change (ΔC)
      if exp(-ΔC/T)> random[0,1] then
        Accept new individuals
    for all individuals
      Perform mutation with probability of
        mutation
      Calculate the cost change (ΔC)
      if exp(-ΔC/T)> random[0,1] then
        Accept new individuals
  Broadcast P_sub to all other nodes
  Select new P_sub randomly
  Keep the best individual
```

3.3 Distributed MGA Hybrid Algorithm

A new hybrid algorithm called MGA combines the merits of mean field annealing (MFA) and simulated annealing-like genetic algorithm (SGA). MFA can reach the thermal equilibrium faster than simulated annealing and GA has powerful and various genetic operations such as selection, crossover and mutation.

First, MFA is applied on a spin matrix to reach the thermal equilibrium fast. After the thermal equilibrium is reached, the population for SGA is made according to the distribution of spin matrix. Next, GA operations are applied on the population while keeping the thermal equilibrium by transiting the new state with Metropolis criteria. Distributed MFA and distributed SGA are applied by turns until the system freeze.

The followings are the pseudo code for the distributed MGA algorithm of each node.

```
<Distributed MGA Hybrid Algorithm>
Forms the spin matrix,s=[s11, ..., sip, ..., sNN]
Get the initial temperature T0, and set T= T0
while T ≥ Tf
  Executes distributed MFA
  Forms SGA population from a spin matrix
    of MFA
  Executes distributed SGA
  Forms the spin matrix of MFA from SGA
    population
  T= α×T    /*decrease the temperature*/
```

Initial temperature, T_0, is set such that the probability where the cost change is less than ε is more than 95% for the number of patterns(=N) annealing process where ε is set to 0.5. Final temperature (T_f) is set to the temperature where the value of the cost change is in ε/1,000 for continuous N temperature changes. A fixed decrement ratio, α, is set to 0.9 experimentally. This strategy decreases the temperature proportional to the logarithm of the temperature.

IV. SIMULATION AND EVALUATION

The simulation is implemented in distributed environments which are made up of 600 MHz personal computers running Linux operating system connected via 10Mbps Ethernet. The sequential version of proposed distributed MGA hybrid algorithm is compared with MFA and GA in Table II. The packing density of MGA is greater than that of MFA, but the execution time of MGA is longer than that of MFA. In comparing with GA, the packing density of MGA is almost same while the execution time of MGA is shorter.

TABLE II
THE PERFORMANCE COMPARISONS OF DIFFERENT ALGORITHMS

# of patterns	packing density			Execution time(min)		
	MFA	GA	MGA	MFA	GA	MGA
18	0.87	0.87	0.87	28	46	37
38	0.79	0.88	0.89	42	73	69
43	0.81	0.91	0.89	62	121	85
75	0.90	0.97	0.97	41	82	51
97	0.77	0.91	0.90	46	87	72
106	0.89	0.95	0.94	76	135	83

The experiment shows that the crossover probability is optimal at 0.8 and the mutation probability has a great effect on the solution quality. When the mutation probability is large, it becomes quite a random search. While the mutation probability is small, it is difficult to escape from the local optimal configurations. The mutation probability varies in the range of 0.01 to 0.05. We simulated the proposed algorithm using various crossover and mutation probabilities.

The inherent nature of a mean field annealing and genetic algorithm makes it possible to be paralleled in distributed environment. The spin matrix and population is almost equally divided to nodes in MPI and MFA and GA operations are implemented independently. Thus this distributed version keeps the solution quality as much as the sequential one with less computation time.

The parallel speedup generally increases proportional to the number of patterns (Fig. 6). However, for the Pat-43 problem, it is much complicated and the computing time is extremely long, so the speedup is larger than other problems. This shows that proposed distributed MGA algorithm is useful as the problem size increases.

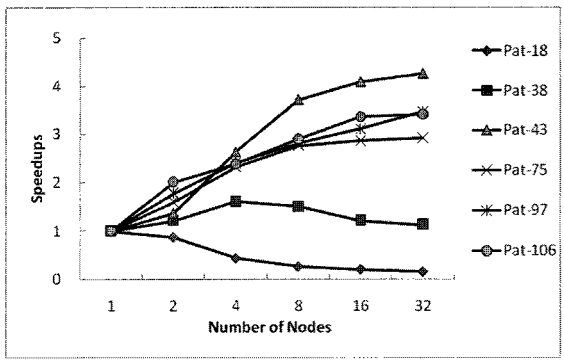

Fig. 6. Speedups of Different Patterns.

V. CONCLUSIONS

In this paper, we proposed a new distributed hybrid algorithm called MGA. This proposed MGA algorithm was verified successful to find the optimal solution. The proposed approach combines the merits of MFA and GA on a cutting stock problem in MPI. The execution time of MGA is shorter than that of GA alone keeping the almost same solution quality.

MGA was also verified by producing more promising and useful results as the problem size and complexity increases. The proposed algorithm can be easily developed as a distributed algorithm since MFA and GA can be parallelized easily.

In addition, this algorithm may be easily modified to solve the other optimization problems as long as the cost function can be linear and genetic representation is possible.

ACKNOWLEDGMENT

This work was supported by the Sangmyung University Research Grant of 2009.

REFERENCES

[1] G.F. Cintra, F.K. Miyazawa, Y. Wakabayashi, E.C. Xavier, "A note on the approximability of cutting stock problems," European Journal of Operational Research, Volume 183, Issue 3, pp. 1328-133216 December 2007.

[2] Wonil Kim, Chul-Eui Hong, Yeong-Joon Kim, "Distributed Simulated Annealing for Composite Stock Cutting Problem," pp. 238-242, PDCAT 2004.

[3] Fowler, R.J., Paterson, M.S., and Tanimoto, S.L., "Optimal packing and covering in the plane are NP-complete," Information Processing Letters, Vol.12, No.3, pp. 133-137, 1981.

[4] Vassilios S. Vassiliadis, "Two-dimensional stock cutting and rectangle packing: binary tree model representation for local search optimization methods," Journal of Food Engineering, Volume 70, Issue 3, pp. 257-268, October 2005.

[5] Cláudio Alves, J.M. Valério de Carvalho, "A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem," Computers & Operations Research, Volume 35, Issue 4, pp. 1315-1328, April 2008.

[6] Baker, B.S., Coffman, E.G., and Rivest, R.L., "Orthogonal packings in two dimensions," SIAM Journal on Computing, Vol.9, No.4, pp. 846-855, 1980.

[7] Y. Khalifa, O. Salem, A. Shahin, "Cutting stock waste reduction using genetic algorithms," GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation , July 2006

[8] T. W. Leung, C. H. Yung, Marvin D. Troutt, "Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem," Computers & Industrial Engineering, Volume 40, Issue 3, pp. 201-214, July 2001.

[9] Yonghuai Liu, "A mean field annealing approach to accurate free form shape matching," Pattern Recognition, Volume 40, Issue 9, pp. 2418-2436, September 2007.

[10] Kyeongmo Park, Sungcheol Kim, Chuleui Hong, "A Distributed Hybrid Algorithm for Optimized Resource Allocation Problem," pp. 1118-1127, ICONIP 2006.

[11] Pau Bofill, Roger Guimerà, Carme Torras, "Comparison of simulated annealing and mean field annealing as applied to the generation of block designs," Neural Networks, Volume 16, Issue 10, pp. 1421-1428, December 2003.

Chuleui Hong received B.S. degree in Hanyang University, Seoul, Korea in 1985, and his M.S and Ph.D. degrees in Computer Science at New Jersey Institute of Technology, USA and University of Missouri-Rolla, USA in 1989 and 1992, respectively. He was a senior researcher in Electronic and Telecommunications Research Institute, Korea, from 1992 to 1997. He joined Sangmyung University as a faculty member in 1997 and is currently professor of Computer Science Department at Sangmyung University, Seoul, Korea. His research interests include Parallel and Distributed System, Optimization Algorithm, multimedia Application, and Intelligent Agent.