15th Triennial World Congress, Barcelona, Spain

AN IMPROVED GENETIC ALGORITHM FOR RECTANGLES CUTTING & PACKING PROBLEM

Wang Shoukun, Wang Jingchun, Jin Yihui

Tsinghua University, Beijing 100084, P. R. China
Department of Automation, Tsinghua University, Beijing 100084, P. R. China

Abstract: After reviewing the background and description of cutting & packing problem and packing algorithms, this paper proposed an improved genetic algorithm for non-guillotine rectangles packing problem. First, self-adaptive mutation probability is adopted to avoid pre-maturity, some commonly used crossover operators are compared, and a stochastic hill-climbing operator is designed to improve the local searching ability. Second, the greedy mechanism is introduced to acquire good packing pattern after packing order is determined by the GA. Computational results show the effectiveness of these improvements. *Copyright © 2002 IFAC*

Keywords: Genetic algorithm, Random search, Heuristic search, Global optimization, Packing Problem.

1. INTRODUCTION

The packing problems have been widely studied during the last three decades, as they are often faced in industries such as metal, cloth, and papermaking. The rectangular pieces packing problem, cut also from rectangular stock, is one particular case of this set of problems. The aim is often to achieve the minimum trim loss.

Packing problems have been proven as NP complete problems in many literatures. Since the geometrical characteristics of packing objects should also be taken into consideration, their computational algorithms are more complex to design. Based on their descriptions and solutions, Dyckhoff (1990) classified the general stocking cutting and packing problems into five categories in his classical literature:

- Cutting stock and trim loss;
- Bin packing, strip packing, and knapsack;
- Vehicle loading, pallet loading, and container loading;
- Assortment, depletion, design, dividing, layout;
- Capital budgeting, memory allocation, and multi-processor scheduling.

Generally, the stock cutting problems can be divided into regular packing problems and irregular packing problems by their shape, also into one-dimensional, two-dimensional, three-dimensional and multi-dimensional packing problems by their range of the solution space.

The earliest literature on stock cutting problems appeared in 1939, but actually the academic research works have boomed since 1970s'. Gyson and Gregory (1974) advanced a heuristic packing algorithm in the first time. And then, traditional optimization algorithms such as rule based packing algorithm (Madsen, 1979), branching and bounding (Golden, 1976), dynamic programming (Golden, 1976), and integral programming (Farley, 1988) were practiced on this subject comprehensively. Coffman and Shor (1990) brought forward a Benchmark to evaluate the effectiveness of packing algorithms, which was described as packing a number of squares with borders randomly generated between (0, 1) into a 1 unit wide infinite long rectangular stock.

Modern optimization technologies have made a remarkable progress since 1990s'. Genetic algorithms, simulated annealing, fuzzy searching, neural network and other meta-heuristic searching algorithms showed a strong advantage especially in combinatorial optimization problems. Jakobs (1996) designed a BL (bottom-left) -algorithm based genetic algorithm for packing of rectangles; Hopper and Turton (1999), Liu and Teng (1999) brought forward their improved solutions on that algorithm respectively; Faina (1999) discussed the application of simulated annealing algorithm on packing problems in detail. Based on the four papers above, an improved genetic algorithm is presented for rectangular packing problems with a new set of algorithm evaluation indicators in this paper. Simulation results are included to show its efficiency.

This paper is organized as follows. In Section 2, the genetic algorithm, model of packing problems and their evaluation indicators are introduced. In Section 3, the implementation of non-guillotine cut algorithms is discussed in detail. In Section 4, the simulation results of numerical examples are showed and the advantages of the algorithms are explained. Finally, conclusion and problems for further study are proposed in Section 6.

2. GENETIC ALGORITHMS AND DESCRIPTION OF PACKING PROBLEMS

Genetic Algorithms (GAs) is a powerful global optimum strategy based on the simulation of natural genetics and evolution (Holland, 1975). Its applications in cutting & packing problems are scarce, and roughly divided into two methods. One is based on the coordinates of the small items and the binary or decimal coding [12]. The other is based on the order of the placement or cutting and the integer coding [7, 8, 9]. Through the exhaustive simulations and GAs design experience, it is discovered that the first method calls for high designing arts on the penalty item of the fitness function, and is easily trapped into the in-feasible solution and local optimum; the second method should be combined with other algorithm to generate packing patterns, but it can somehow overturn the shortcomings of the first method. Based on the second method, a new packing algorithm and genetic operators are designed to improve the structure and control parameters of GAs, and they also enhance the exactness and stability of the algorithm.

The general structure of cutting & packing problems is as follows:
1. One or more stocks of certain figures, forms and sizes;
2. A number of small items to be cut or packed with different figures, forms and sizes;
3. Some cutting & packing restrictions;
4.Evaluating criteria based on one or more objectives.

For the rectangular cutting & packing problem, the third part mentioned above is very important. As to some industries such as glass and polystyrene, it is required that the cut should be orthogonal form one edge to the other, which is called guillotine cut. The cut without that requirement are called non-guillotine

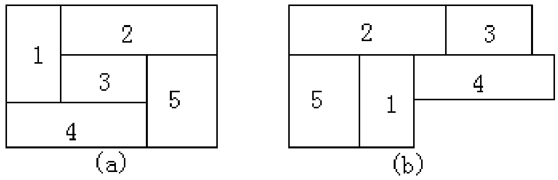

Fig. 1. Non-Guillotine cut and guillotine cut.

cut. As shown in Fig. 1., (a) is non-guillotine cut, and (b) is guillotine cut.

Generally, non-guillotine cut is more efficient than guillotine cut, but it requires larger searching space and more complex cutting and packing algorithm. Considering that non-guillotine cut is more general than guillotine cut, an improved GAs for non-guillotine cut is presented and it can be easily modified to fit the guillotine cut situation.

The general objective of packing problems is to minimize the waste area rate, and the most widely used waste area rate is as follows: let S_t be the area of the stock, the area of rectangles are respectively S_i and the number of them is N, and then the waste area rate W is

$$W = \frac{S_t - \sum_{i=1}^{N} S_i}{S_t} \times 100\% \qquad (1)$$

Based on the area rate index above, the perimeter index is used to improve the quality of the waste stock. Let C_w be the perimeter of the waste stock, expected waste stock perimeter is C_E, and then the perimeter index is

$$C = \frac{C_w - C_E}{C_w} \times 100\% \qquad (2)$$

$$C_E = 4\sqrt{S_t - \sum_{i=1}^{N} S_i} \qquad (3)$$

C_E represents the perimeter of the square whose area is equal to the waste stock's. Minimizing the perimeter index C can make the shape of the waste stock to be a square as close as possible, and improve the reuse quality especially for expensive stocks.

3. ALGORITHM DESCRIPTION

The most widely used simple heuristic packing algorithm is BL algorithm. There are two steps in the packing process of BL algorithm. First, randomly generate the packing order of the rectangles. Second, according to the packing order, place each rectangle into the stock and move it downward and leftward to the bottom-left point of the stock until impossible. Hopper and Turton (1999) uses SGA (Simple Genetic Algorithm) as the first step of the BL algorithm to generate and optimize the packing order, however, the BL packing process easily causes gaps among the packed rectangles and these gaps cannot be used again. Faina (1999) repeats the BL packing process to find the gaps, and Liu and Teng (1999) tries to find the lowest point of the unpacked stock the make up the gaps. Both of them improve the quality of the solutions but don't solve the gap problem completely. In this paper, a greedy packing algorithm is presented based on all the unpacked potential optimal point of the stock and the computational results show that it can completely avoid gaps. Furthermore, the uniform coding schema is used to diversify the chromosome population, adaptive crossover and mutation probabilities to prevent the pre-maturation, and a Stochastic Hill-Climbing operator is also designed to improve the searching quality.

Step 1. Let generation g=0, initialize M individuals according to uniform coding schema. For any individuals M_i and M_j, define the Hamming distance H_{ij} of the two individuals as:

$$H_{ij} = \sum_{k=1}^{N} \left| M_i^k - M_j^k \right| \tag{4}$$

$$\left| M_i^k - M_j^k \right| = \begin{cases} 0 & if \quad M_i^k = M_j^k \\ 1 & otherwise \end{cases} \tag{5}$$

M_i^k represents the kth bit of the ith individual. If $H_{ij}<T$, then remove one individual of i and j randomly, and regenerate a new individual. T is a pre-decided threshold, and a larger T will help to generate a more diversified population.

Step 2. Pack and evaluate individuals of the gth population according to the following greedy algorithm:

1. Initialize the singly linked list P of optional unpacked points. The first element of P is the up-left point of the stock p^0.

2. According to the packing order, place the up-left point of the next rectangle on the optional point p^j of the list P, and calculate its fitness f_j. After all the optional points in the list P are calculated, find the smallest fitness f_k =min f_j and remark the relative point p^k.

3. Place the up-left point of the rectangle on the point p^k, and delete p^k, at the same time insert the bottom-left point p^k, bottom-right point p^{k+1}, and up-right point p^{k+2} into the list P, then change the pointers.

4. If all the rectangles are place, turn to next step. Else go to 2.

The fitness function is $f = \alpha W + \beta C$. For any individual, W is the area index of the waste stock, and C is the perimeter index. The α and β are indexes related to the generation g. At the beginning α>0.9 and β<0.1, after a certain number of generations when the waste area index is close to the satisfied solution, increase the value of β to improve the quality of the waste stock.

Step 3. Do select operation on the current population by the roulette and elitism schema. The probability P_i of individual i to be chosen into the next generation is:

$$P_i = \frac{1 - f_i}{\sum_{j=1}^{N} (1 - f_j)} \tag{6}$$

Preserve 5%～10% individuals with best fitness and put them directly into the next generation.

Step 4. Do crossover operation on the chosen population. Four crossover operators including Single-point crossover, partial mapping crossover (PMX), order crossover (OX) and Non-ABEL crossover are tried and compared for their efficiency through the computational results.

Step 5. Do mutation operation on the population. The swap and inverse operators are used, and the probability of mutation P_m is:

$$P_m = P_0 \left[\mu(\bar{f} - f_{\min}) + \frac{\nu}{\overline{H}} \right] \tag{7}$$

P_0 is the initial value, μ and ν are control parameters, $\bar{f}$ is the average fitness value, f_{min} is the best fitness value, and $\overline{H}$ is the average Hamming distance of the population. Furthermore, a new stochastic-hill-climbing (SHC) mutation operator is proposed. For the 5% individuals M with the best fitness value and their candidate children M' generated by the mutation operation, if $f_{M'} < f_M$, accept M' with probability 1; otherwise accept M' with probability $e^{-(f_{M'} - f_M)/T_g}$. T_g is dependent on the generation g. At the beginning T_g is larger, helping to enlarge the search space, and decreases as the generation g increasing.

Step 6. If $g > g_{max}$ or the best fitness value does not change in 10 generations, turn to the next step; otherwise let g=g+1, and go to Step 2.

Step 7. Output the optimal solution.

4. COMPUTATIONAL RESULTS

The benchmark designed by Coffman and Shor (1990) is used to compare the efficiency of different crossover operators and algorithm of this paper and Faina's (1999). First, initialize a stock with width w and infinite length; second, give a certain number of rectangles (between 8 and 64) with width and length randomly generated between (0, w). Each simulation runs 100 times. Table 1 compares the four crossover operators, two mutation operators and the stochastic-hill-climbing operator. When comparing crossover operators, the stochastic-hill-climbing operator is used, and when comparing mutation operators, PMX crossover operators are used. Table 2 compares the efficiency of the algorithm used in this paper and in Faina's (1999).

Computational results in Table-1 show that except the Non-ABEL operator, there is not much difference

Table 1 Comparison of Operators
Number of rectangles: 32

Crossover	Average %	Min/Max %	Mutation	Average %	Min/Max %
Single-point	96.12	94.14 / 96.91	Swap	93.89	92.17 / 95.55
PMX	96.33	93.98 / 97.71	Inverse	94.47	92.08 / 96.13
OX	96.07	93.24 / 97.32	SHC	96.33	93.98 / 97.71
Non-ABEL	94.46	91.08 / 97.65			

Table 2 Comparisons of Algorithms

Number of rectangles	This paper			Faina's algorithm		
	Average (%)	Min/Max (%)	Maximum Deviation (%)	Average (%)	Min/Max (%)	Maximum Deviation (%)
8	92.34	90.11 / 95.28	2.94	90.31	82.34 / 96.36	7.97
16	95.15	93.16 / 97.09	1.99	92.95	87.96 / 96.91	4.99
32	96.33	93.98 / 97.71	2.35	93.29	91.44 / 94.45	1.85
64	93.44	91.86 / 96.21	2.77	92.29	88.56 / 94.88	3.73

of the efficiency among the rest of other three operators. Since the Non-ABEL operator is more like random search, it is not good to preserve the good schema, which is the key technique of the GAs, and the performance is not stable either. As for the mutation operators, it's obvious that the SHC operator has better performance. The SHC operator not only chooses offspring with higher fitness value, but also accepts individuals with lower fitness value by certain probability, which is a good example on the combination of the GAs and neighborhood local-search techniques to improve the performance.

In Table-2, the performance of the algorithm used in this paper is better than that of Faina's, and one point to be mentioned is that the total evaluation of fitness value in Faina's algorithm is 100 times of the number of the rectangles, whereas 40 times in this paper, which is much more time efficient. After packing order is generated by the GAs, greedy algorithm is used to pack the rectangles into the stock, and get a better packing pattern each time. Whereas Faina's algorithm uses a random packing algorithm, which does not guarantee better packing pattern even though the packing order is good. Furthermore, the min/max stock usage rate of this paper is obviously smaller than that of Faina's algorithm. It's because of the hidden parallel and mass evolution of GAs, whereas the simulating annealing algorithm used in Faina's algorithm is more sensitive on the initial value when applied to combinatorial optimization problems.

Jakobs (1996), Hopper and Turton (1999), and Liu and Teng (1999) also use GA, but the performance cannot be compared because they do not give the form and sizes of the stock and rectangles they used. Since they only try on simple genetic algorithms and bottom-left packing algorithm, our refined crossover and SHC operator with greedy search algorithm is more efficient.

5. CONCLUSION

In this paper, an improved GA combined with greedy search algorithm is proposed for non- guillotine rectangles packing problem. First, self-adaptive mutation probability is adopted to avoid pre-maturity, compare and refine some common used crossover operators, and design a stochastic hill-climbing operator is to improve the local search ability. Second, a greedy searching mechanism is introduced to acquire good packing pattern after packing order determined by the GA. Finally, computational results are presented to show the effectiveness of these improvements.

This algorithm is also applied on guillotine rectangles packing problem and got satisfied results. Now research on polygon packing problems are undergoing.

REFERENCE

Coffman, E.G. and Jr. P. W. Shor (1990). Average-case analysis of cutting and packing in two dimensions, Eur. J. Operational Research, **44**, 134-144

Dyckhoff, H. (1990). A typology of cutting and packing problrms. Eur. J. Operational Research, **44**, 145-159

Faina, L. (1999). An application of simulated annealing to the cutting stock problem. Eur. J. Operational Research, **114**, 542-556

Farley, A. A. (1988). Mathematical Programming Models for Cutting-Stock Problems in the Clothing Industry. J. Ope. Res. Soc., **39**, Vol. 1, 41-53

Golden, B.L. (1976). Approaches to the cutting stock problem. AIIE Trans, **8**, 265-274

Gyson, R. G. and A. S. Gregory (1974). The Cutting Stock Problem in the Flat Glass Industry, Operational Research Quarterly, **25**, 41-53

Holland J H. (1975). Adaptation in natural and artificial systems. MI: University of Michigan

Press.

Hopper, E. and B. Turton (1999). A Genetic Algorithm for a 2D Industrial Packing Problem. Computers & Industrial Engineering, **37**, 375-378

Jakobs, S. (1996). On genetic algorithms for the packing of polygons. Eur. J. Operational Research, **88**, Vol. 1, 165-181

Kazarlis, S. (1994). Varying quality function in genetic algorithms and the cutting problem, in Proc. 1st IEEE Conf. Evolutionary Computation, IEEE Service Center, Vol.1, 166-169

Liu, D. and H. Teng (1999). An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. Eur. J. Operational Research, **112**, 413-420

Madsen, O.B.G. (1979). Glass cutting in a small firm. Math. Prog. **17**, 85-90