Genetic Algorithms for Optimal cutting

Toshihiko Ono and Gen Watanabe
Dept. of Communication and Computer Eng.,
Fukuoka Institute of Technology
3-30-1 Wajiro-higashi, Higashi-ku,Fukuoka 811-02, Japan
e-mail: ono@cs.fit.ac.jp

1 Introduction

This chapter deals with two methods for the optimal cutting of different shapes of material. The first is the optimal cutting of one-dimensional material. For instance a steel bar manufacturing process produces required number of bars(hereafter referred to as finished bars) for various lengths according to customer's requirements, by cutting a series of long raw bars. In this process, the cutting should be done to make losses minimal to achieve the highest productivity. Therefore, the optimal cutting method by GAs introduced here has to satisfy two requirements simultaneously : 1) the minimum length of total scrap produced to make waste minimal, 2) the number of finished bars for each length should conform to orders from customers not to produce unnecessary amounts of bars. To satisfy these two requirements, the optimal cutting method adopted here consists of two procedures: minimum scrap length cut and production balance cut, combined into one system. In the early stage of production, the combination of cutting lengths of a raw bar is determined mainly by making the length of each piece of scrap produced minimal and as the production proceeds, the ratio between the numbers of finished bars of each length gets more consideration.

The second optimization problem is how to cut a set of various two-dimensional patterns from a sheet, to make the required sheet length minimal. Concerning the two-dimensional optimal cutting, there are two kinds of problems : the optimal cut of rectangular patterns and that of free patterns. The optimal cutting of rectangular pieces from a sheet has been studied by many researchers[1, 2, 4, 13]. On the other hand, as to the optimal cutting of free patterns, which will be required in the cutting processes of clothes, metal sheets and the like, there have not been so many studies[7, 12], because of the difficulty of theoretical treatment. After much consideration, we have succeeded to solve it by GAs which incorporate special algorithms.

The given problem is originally a kind of two-dimensional search. As the space of two-dimensions is defined by the product of each dimension, it be-

comes quite large compared with that of one-dimension. Therefore, with the application of GAs to this problem, it is difficult to get proper results with a usually adopted tactic, in which genes represent the two-dimensional positions of patterns and the search is done two-dimensionally. In the method adopted here the problem is solved as a (one-dimensional) ordering by incorporating layout determining algorithms(LDAs) into GAs to reduce the dimension of search space. At the same time, LDAs are designed to get the solution as quickly as possible.

When applying GAs to various problems, it is important to consider how to represent the given problem with genes, how to calculate the fitness value and how to incorporate the GAs with other suitable systems. These are illustrated in our applications.

2 Optimal cutting of bars

The problem dealt with here is how to determine the optimal combination of lengths of finished bars in a raw bar, to attain the minimum amount of loss, when the finished bars of different length are produced by cutting raw bars of different length. Figure 1 shows an example of such production processes for steel bars, where a series of raw bars coming one after another are cut to finished bars of various lengths ordered by customers. Since the lengths of raw bars are different from each other due to the losses in the previous processes, the first problem to solve is how to determine the combination of cutting lengths in each raw bar so that the length of the scrap produced is minimal.

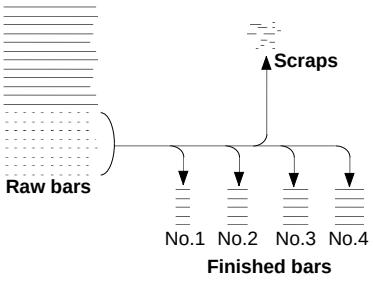

Figure 1: Production of finished bars from raw bars

If the production proceeds with this method only, the imbalance will occur between the number of finished bars ordered by customers and that of those actually manufactured for each length and this will bring about the losses in production. Therefore, keeping balance among the numbers of the finished bars of various length at the end of production is the second requirement to

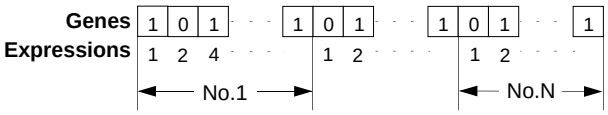

Figure 2: Representation of genes

satisfy. As the result, the optimal selection method of cutting length should satisfy these two requirements at the same time. Now, we will explain what system is adopted to solve these requirement by GAs [10, 11].

1)Expression of genes : As the purpose of the system is to get the combination of cutting lengths in a raw bar, genes are designed to express the number of finished bars to be cut from each raw bar. Figure 2 shows the composition of the genes, where N stands for the number of kinds of the finished bar lengths.

2)Fitness function : The fitness function consists of following two functions to satisfy the above two requirements :

1. Fitness function $F_1(k)$ which is used to make the length of scrap minimal.
2. Fitness function $F_2(k)$ which is introduced to keep balance among the numbers of finished bars for all lengths ordered.

These two functions are combined to make a fitness function $F(k)$ as follows :

$$F(k) = a(k)F_1(k) + b(k)F_2(k), \tag{1}$$

where k stands for the k-th raw bar, and $a(k)$ and $b(k)$ are weighting coefficients explained later.

The function $F_1(k)$ is reduced as follows :

$$F_1(k) = f_1(\Delta L(k)); \tag{2}$$

$$\Delta L(k) = L(k) - \sum_{i=1}^{N} L_i n_i(k), \tag{3}$$

where $L(k)$ is the length of the k-th raw bar, N is the number of kinds of finished bar lengths, L_i is the i-th finished bar length and $n_i(k)$ the number of finished bars of i-th length cut from the k-th raw bar. Since $\Delta L(k)$ is the length of the scrap produced, it should be non-negative. To make the scrap length minimal, the function $f_1(x)$ is selected as a decreasing function whose value is non-negative in the operating range and maximum at $x = 0$. As a function to satisfy these requirements, $f_1(x) = \exp(-c_1 x^2)$ is adopted.

On the other hand, the function $F_2(k)$ expresses how well the number of finished bars for each length matches the number of those ordered by customers and is determined as follows :

$$F_2(k) = \sum_{i=1}^{N} f_2(\Delta r_i(k)); \tag{4}$$

$$\Delta r_i(k) = \frac{r_i}{\sum_{j=1}^{N} r_j} - \frac{p_i(k) + n_i(k)}{\sum_{j=1}^{N} (p_j(k) + n_j(k))}, \tag{5}$$

where r_i stands for the target value of production ratio for the i-th finished bars and p_i stands for the number of the i-th finished bars produced. Hence $\Delta r_i(k)$ becomes the deviation of production ratio from its object value. The function $f_2(*)$ is also to be a non-negative decreasing function to make the deviation of the ratios minimal and therefore the same function as $f_1(*)$ is applied.

Since the lengths of raw bars are assumed to be known only at cutting time, the combination of cutting lengths can't be determined before the time of cutting and therefore, it is imposible to take the lengths of succeeding raw bars into consideration. On the other hand the number of finished bars for each length should match the object value at the end of production. Considering these requirements, to attain the minimum length of scrap we adopted an inclining coefficient method (referred to as Inclining Weight Method), so that the combination of cutting lengths in each raw bar is determined, in the early stage of production mainly to make the scrap length minimal, in the final stage to keep the balance of products, and in the intermediate stage to partly satisfy both requirements. To realize this, the coefficient $a(k)$ is set large at the initial stage and is gradually decreased as the production proceeds. On the other hand, $b(k)$ is changed in the opposite way, that is, from small to large. The actual pattern of change is determined from the results of simulation as shown later.

3) Genetic operation : We adopted simple genetic algorithms(SGA), which consist of the following three operations:

1. Selection by roulette wheel method,
2. One point crossover,
3. Mutation with fixed probability.

Figure 3 shows the flow chart of simulation. The combination of cutting lengths in each raw bar is determined by GAs. Before transition to the next bar, the parameters such as p_i, $a(k)$ and $b(k)$ are reset to the new values calculated by the data collected. After the above procedure having been repeated to each raw bar, the final performance figures were obtained and evaluated.

2.1 Results of simulation

The following two kinds of simulation studies were conducted to investigate the optimizing characteristics of the proposed method, primarily to see how

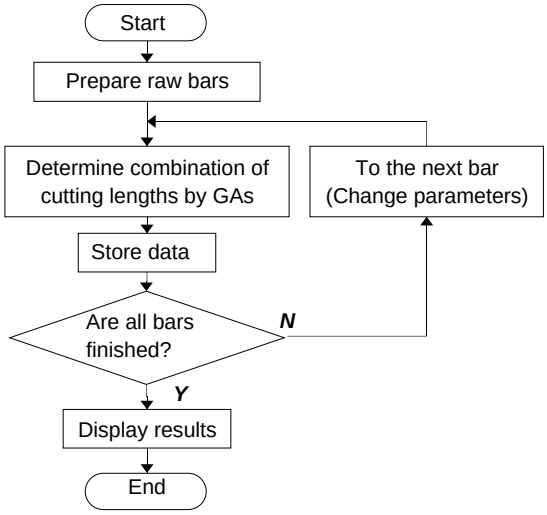

Figure 3: Flow chart of simulation

well the two requirements of both the minimum length of scrap and the balance among the number of finished bars are attained.

1. Optimization for each bar, in which only the minimum length of scrap for each bar is considered. Therefore, the coefficient $b(k)$ is set at zero.

2. Optimization as a whole, in which both the minimum scrap length and the balance of products are satisfied. The simulations are conducted under various production ratios.

Table 1 shows the specifications of raw and finished bars, As to the parameters related to GAs, the population is 50, the probability of crossover 0.1 and the probability of mutation 0.1.

Figure 4 shows the patterns of variation for the weighting coefficients $a(k)$ and $b(k)$ as to k, the number of raw bars processed.

Table 2 shows the results of the simulation for Case 1, in which only the optimization for each bar is considered. From the table we can see that the total length of scrap is kept to zero, and perfect optimization is attained. Both Table 3 and Table 4 show the results of optimization as a total system at various production ratios. In the both cases the production ratios attained by GAs are nearly equal to the taget values, while the total length of scrap is kept fairly small. Ideally this length is to be zero, but it is almost impossible to attain it, because it is difficult to achieve the optimization of all the objectives at the same time and therefore some trade-off among the

Table 1: Specifications of raw and finished bars

Raw bars	finished bars			
Length[m]	Length[m]	Target of ratio[%]		
		Case-1	Case-2	Case-3
123	11	-	25.0	33.3
131	13	-	25.0	33.3
137	17	-	25.0	16.7
145	21	-	25.0	16.7
151				

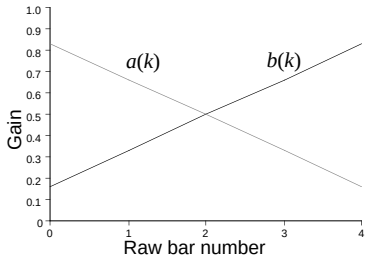

Figure 4: Patterns of weighting coefficients $a(k)$ and $b(k)$

Table 2: Optimal cut for each bar

		Each raw bar					Sum	Ratio[%]	
Length[m]		123	131	137	145	151	687	Target	Result
Number	11m	5	0	2	2	1	10	-	23.3
of	13m	1	1	3	1	3	9	-	20.9
cuts	17m	2	2	2	4	1	11	-	25.6
	21m	1	4	2	2	4	13	-	30.2
L. of scrap[m]		0	0	0	0	0	0	-	100.0

Table 3: Optimal cut for all of bars : case-1

		Each raw bar					Sum	Ratio[%]	
Length[m]		123	131	137	145	151	687	Target	Result
Number	11m	3	1	2	3	2	11	25.0	25.0
of	13m	1	2	3	4	1	11	25.0	25.0
cuts	17m	2	3	2	1	3	11	25.0	25.0
	21m	2	2	2	2	3	11	25.0	25.0
L. of scrap[m]		1	1	0	1	2	5	100.0	100.0

Table 4: Optimal cut for all of bars : case-2

		Each raw bar					Sum	Ratio[%]	
Length[m]		123	131	137	145	151	687	Target	Result
Number	11m	3	3	2	5	4	17	33.3	35.4
of	13m	4	2	3	4	2	15	33.3	31.2
cuts	17m	1	3	2	1	1	8	16.7	16.7
	21m	1	1	2	1	3	8	16.7	16.7
L. of scrap[m]		0	0	0	0	1	1	100.0	100.0

optimized values of those objectives is indispensable, in a multi-objective optimal problem such as this. We also conducted studies to realize a similar system using the neural networks of cross-connected Hopfield type [9]. The GAs method, however, proved to be much better than the neural networks. Especially, we had problems of a local minimum phenomenon at the neural networks, even though we added a proper annealing operation to get rid of this problem. From these experiences the GAs method seems suitable to the problems having many local minima such as those treated here.

3 Optimal cutting of two-dimensional patterns

In this section, how to lay out a set of various two-dimensional patterns(polygons including concave ones) on a sheet to satisfy the following two requirements is discussed, when these patterns are produced by cutting the sheet.

1. To lay them out on the sheet without mutual overlapping,
2. To make the required sheet length minimal.

In applying GAs to this kind of problem, the normally adopted representation of genes will be the position of each pattern on a sheet. From the estimation of

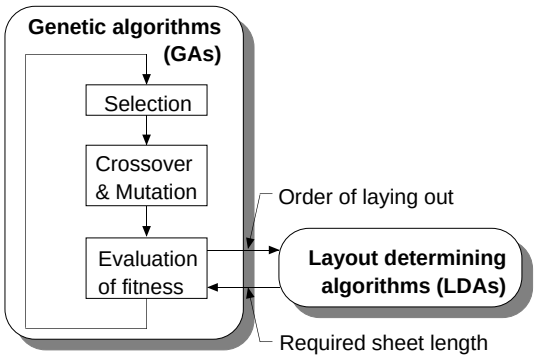

Figure 5: Configuration of systems

the size of search space, however, we have concluded that this representation method is impractical. In this method, since the genes will be a binary representation consisting of both the positions in width and those in length to all of the patterns, the length of the genes is $N(\log_2 W + \log_2 L) = N\log_2(WL)$ and thus the size of search space becomes $2^{N\log_2(WL)} = W^N L^N$, where N is the number of patterns, and W and L are the width and length of the sheet respectively.

On the other hand, if it is possible to solve the problem as an ordering one, the size of search space becomes $N!$, that is, the permutation of N. Considering the amount of W,L,N , we can easily understand that a considerable reduction of search space can be obtained with this method. From these results, we decided to solve the problem by an ordering GAs, introducing layout determining algorithms(LDAs) to change a two-dimensional search into a one-dimensional (ordering) one. Figure 5 shows the flow chart of the systems. The GAs and LDAs work sequentially, communicating each other. The GAs determine the order of patterns to arrange them on a sheet and send it to the LDAs, while the LDAs fix the position of each pattern according to the order given by the GAs so that each pattern on the sheet doesn't have an overlap and an extra opening. After the layout of all the patterns is fixed, the LDAs return the required sheet length to arrange the patterns to the GAs. With this length data, the GAs determine the order of pattern arrangement to make the object value(the required sheet length) minimal, by genetic operations such as selection, crossover and mutation.

As the applications of GAs to the ordering problem have been already studied in a traveling salesman problem [8], the results gained there were used in our systems.

For the convenience of explanations, the arrangement of polygon pieces is treated at first, and then the method will be expanded to free patterns later.

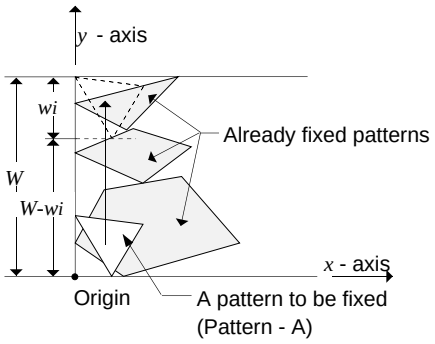

Figure 6: Search of overlapping

3.1 Layout determining algorithms(LDAs)

To determine the position of each pattern on a sheet without overlapping and extra openings between patterns according to the order given by the GAs, we have introduced layout determining algorithms. As shown in Figure 6, the allowable position for each pattern is searched on the sheet, each starting from the origin of the sheet and walking in the direction of width(Y-axis), while stepping in the longitudinal direction(X-axis). The allowable position being found, the pattern is set there and a marking is given to the sheet data to store the occupied area by the pattern. To minimize the required search time, the followings are considered.

Let pattern-A be the pattern to be searched for its position on the sheet. At first the check is done to see if there is an overlap between the pattern-A and those already fixed on the sheet. Even though this check is originally two-dimensional, it can be done under an one-dimensional basis or less, by checking overlapping on the boundary line and at the vertices of the pattern-A, as shown in Figure 7. There is no overlap, as long as no part of the boundary line is inside of already fixed patterns on the sheet. Therefore, by checking if the boundary line is inside the already fixed patterns, the necessary overlap-check can be made and time saving is attained because this checking is done along the boundary line of one-dimension. Moreover, if at least one vertex is within the fixed patterns, it means that there is an overlap. As the vertices are fewer than the points on the boundary line in general, the check at vertex prior to the same on boundary line contributes to another savings in calculation time.

As Figure 8 shows, the check on boundary line and the same at vertex are integrated in the LDAs and activate serially. If one of them fails, the pattern-A jumps to the next search position by an amount equal to the distance calculated by the following method. After both of these checks have

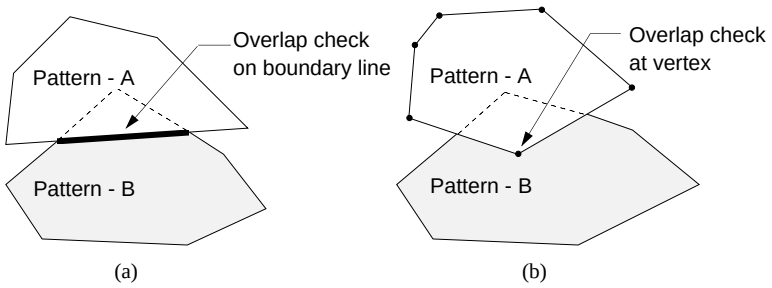

Figure 7: Boundary line and vertices

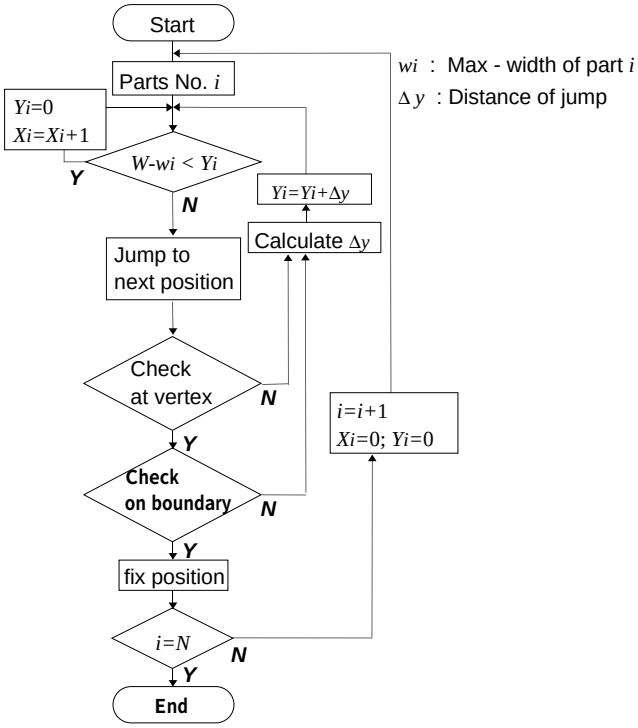

Figure 8: Flow chart of LDAs

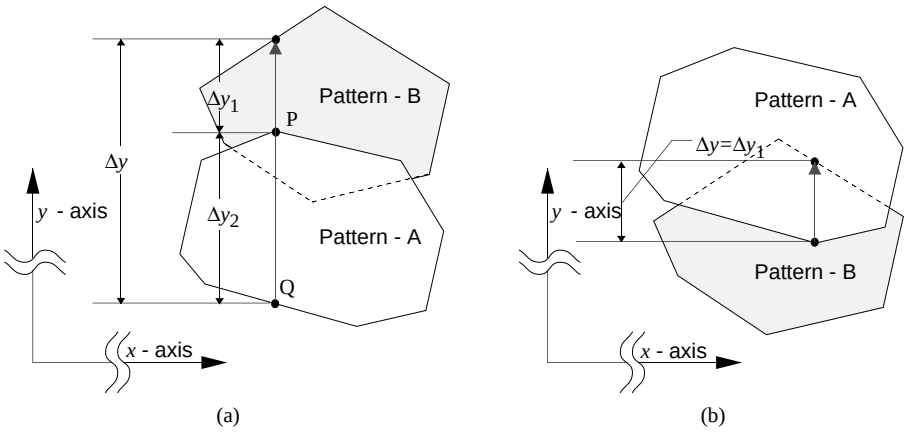

Figure 9: Calculation of jump distance

succeeded, the pattern-A is set there.

When an overlap is found, the distance of the jump to the next position is calculated, referring to the mutual position between the pattern-A and each of the overlapped fixed patterns on the sheet. As shown in Figure 9, the method is divided into two methods according to the mutual position. Taking the check at vertex as an example, these methods are explained. At first the calculation method at one of the vertices is explained. As shown in (a)of Figure 9, to the vertex P in the upper side of the pattern-A, the necessary jump distance Δy of the pattern-A in the Y-axis direction to get out of the pattern-B consists of Δy_1 and Δy_2, because the portion of the pattern-A between the vertices P and Q should be out of the pattern-B. The distance Δy_1 is obtained by scanning upward from the vertex P until the end of the fixed pattern-B. On the other hand, since Δy_2 is inherent to each vertex, it can be calculated beforehand and stored in memory. As for the vertex in the lower side, Δy_2 is zero as can be seen in (b) of Figure 9, hence only the calculation of Δy_1 is required. Therefore, the required jump distance is calculated as the maximum value of Δy for all the vertices of the pattern-A.

The calculation method of the jump distance for the boundary line is the same as the above, except for the vertices being replaced by the points on the boundary line.

The flow chart also shows the procedures at the near end of the sheet in the width side. In the scanning in the Y-direction, It is not necessary to scan up to the upper end of the sheet, but it is enough to scan for the range reduced by as much as the width of the pattern-A.

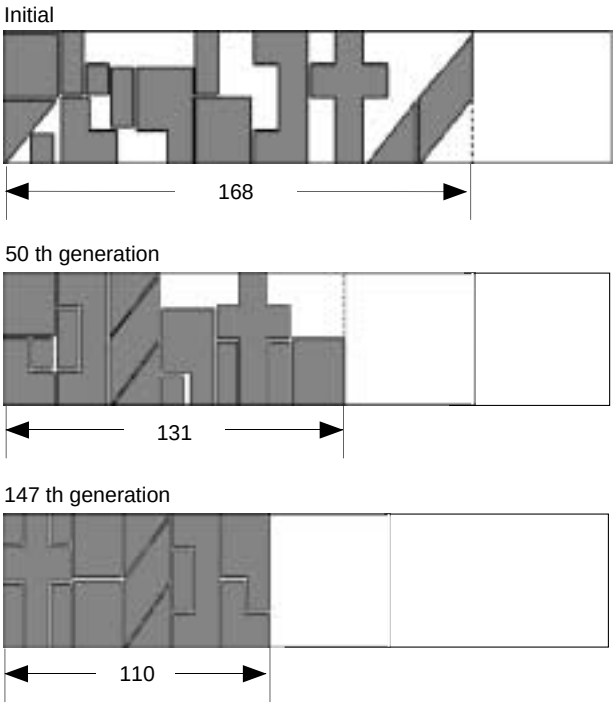

Figure 10: Optimal arrangement of 14 patterns

3.2 Genetic algorithms(GAs)

As to the application of GAs to an ordering problem, various methods have been studied. Among them, we have used Genitor [3][8] as a software workbench of GAs. The genes are represented by the path representation, with the rank method for the selection. For the crossover, three methods have been compared among various methods applied to the path representation. As to the mutation, the method exchanging one random element of genes with another random one is added to Genitor.

3.3 Results of simulations

Among the various results obtained by simulation studies, two typical examples will be explained here. The first simulation is for the case where the optimal value is clear. As shown in Figure 10, the optimal arrangement of a set of 14 patterns has been obtained at the 147th generation with 50 individuals, starting from a random state. Since one generation in Genitor corresponds to the calculation of each individual, the 147th generation in

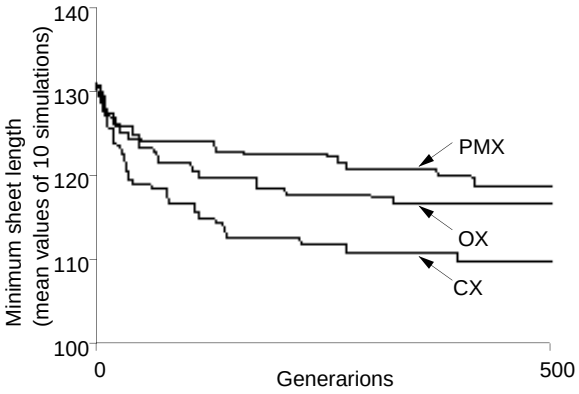

Figure 11: Comparison of convergence characteristics

Genitor are equivalent to the third generation by the ordinary representation, considering the number of individuals being 50.

Figure 11 shows the comparison of the convergence characteristics for three crossover methods : CX(Cyclic crossover), PMX(Partially-mapped crossover) and OX(Order crossover)[8]. The CX method resulted in the best performance of all.

We have investigated the effect of the check at vertex on the overlap check. The calculation time needed to obtain the optimal state was 298.4 seconds without the check at vertex, while 34.4 seconds with it, on a workstation(Sun sparc classic), that is, the calculation time is reduced to 1/9 by the check at vertex.

The second simulation is for the optimal layout of 36 polygons as shown in Figure 12. With 50 individuals, the results that seem optimal are gained at the 497th generation (tenth generation by ordinary representation). Even though there is no theoretical method to confirm that the results gained here are optimal, the above result seems optimal, from the fact that the same results were obtained on various initial random states and from the results of the first simulation.

3.4 Extension to free patterns

We have explained the optimal layout of polygons(including concave one) so far. We can, however, easily extend this method to the free patterns of curved boundary line, by applying both the check-at-vertex method to proper number of the points selected from the points on the boundary line and the check-on-boundary method to the boundary line; those are the same as the methods explained. Figure 13 shows one of examples, for 22 free pattens,

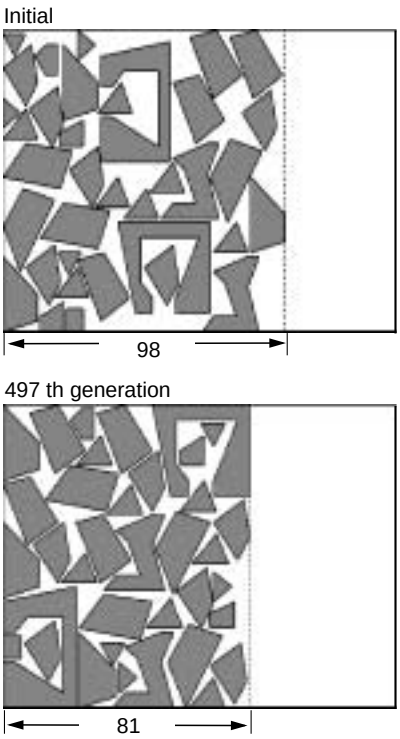

Figure 12: Optimal arrangement of 36 patterns

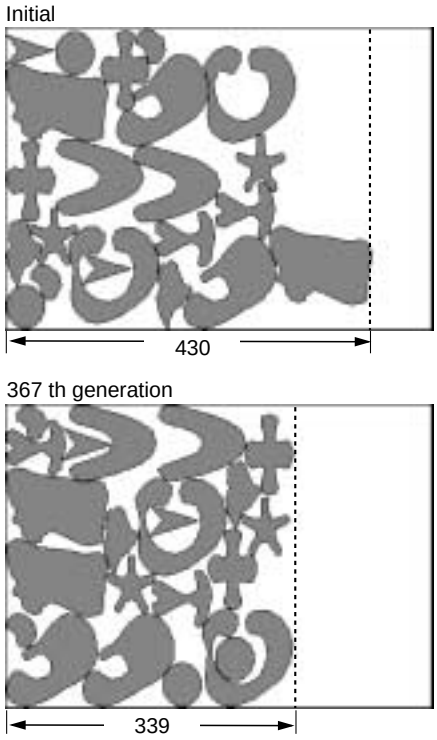

Figure 13: Optimal arrangement of 22 free patterns

which was obtained with 20 individuals.

4 Conclusions

In this chapter we have explained two kinds of application of GAs to the optimal cutting method. One is the optimal cut of bars, which requires multi-objective optimization. By applying GAs in multiple stages with varying weight coefficients two requirements of minimum scrap length and production balance are attained. The other is the optimal cut of two dimension free patterns from a sheet. By combining GAs with layout determining algorithms to solve a two-dimensional problem as an one-dimensional order one, the required calculation time is considerably reduced. Hopefully these results will be applied to not only these problems but similar ones as well.

References

[1] Chauny, F. and Loulou, R. et al. : A Two-phase Heuristic for the Two-dimensional Cutting-stock Problem, Journal of the Operational Research Society, Vol.42, No.1, pp.39–47, 1991

[2] Christofides, N. and Hadjiconstantinou, E. : An exact algorithm for orthogonal 2-D cutting problems using guillotine cuts, European Journal of Operational Research 83, pp.21–38, 1995

[3] Davis, L. Ed. : Handbook of Genetic Algorithms, Van Nostrand Reinhold, 385p, 1991

[4] Dyson, R. G. and Gregory, A. S. : The Cutting Stock Problem in the Flat Glass Industry, Operational Research Quarterly, Vol.25, No.1, pp.41–53, 1974

[5] Goldberg, D.E. : Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Pub. Co., 412p, 1989

[6] Holland, J.H. : Adaptation in Natural and Artificial Systems, The University of Michigan Press, 211p, 1975

[7] Jakobs, S. : On genetic algorithms for the packing of polygons, Proc. of the KI-94 Workshop, pp.84–85, 1994

[8] Michalewicz, Z. : Genetic Algorithms + Data Structures = Evolution Programs, Second, Extended Edition, Springer-Verlag, 340p, 1994

[9] Ono, T. : Application of Neural Networks to Optimal Selection of Cutting Length of Bars (in Japanese), T.IEE Japan, Vol. 113–D, No.12, pp.1371–1377,1993

[10] Ono, T. and Watanabe, G. : Application of Genetic Algorithms to Optimizing Problems(In Japanese), Language and Information Processing (Fukuoka Institute of Technology), Vol.6, pp.89-96, 1995

[11] Ono, T. and Watanabe, G. : Application of Genetic Algorithms to Optimal Selection of Cutting Length of Bars, Proceedings of Fifth FIT-Ajou University Joint Seminar, pp.24-31, 1995

[12] Petridis, V. and Kazarlis, S. : Varying Quality Function in Genetic Algorithms and The Cutting Problem, Proc. of the First IEEE Conf. on Evolutionary Computation, pp.166–169, 1994

[13] Vasko, F. J. : A computational improvement to Wang's two-dimensional cutting stock algorithm, Computers ind. Engng, Vol.16, No.1, pp.109–115, 1989

[14] Whitley, L.D. and Vose, M.D. Ed. : Foundations of Genetic Algorithms.3, Morgan Kaufmann Pub., 336p, 1995