Optimal process planning for compound laser cutting and punch using Genetic Algorithms

S.Q. Xie* and J. Gan

Department of Mechanical Engineering,
The University of Canterbury,
Private Bag 4800, Christchurch 8140, New Zealand
E-mail: s.xie@auckland.ac.nz
E-mail: j.gan@ec.auckland.ac.nz
*Corresponding author

G. Gary Wang

School of Engineering Science,
Simon Fraser University,
250-13450 102 Ave., Surrey, BC, Canada V4N 2T3
E-mail: gary_wang@sfu.ca

Abstract: This paper investigates the nesting issue and the machining path planning issue for improving the sheet metal machining efficiency. The nesting issue is to maximise sheet metal material utilisation ratio by nesting parts of various shapes into the sheet. The path planning issue is to optimise machining sequence so that the total machining path distance and machining time are minimised. This work investigates the two issues by using Genetic Algorithms (GA). The proposed GA approach uses a genetic encoding scheme and a genetic reproduction strategy to reach an optimum solution. Case studies are carried out to test the GAs. The effectiveness of the GA path planning approach is compared with the Ant Colony (AC) algorithm (Wang and Xie, 2005). The results show that GA achieves better performances in path planning than the AC algorithm.

Keywords: path planning; nesting; GA; genetic algorithms; AC; ant colony algorithm; compound machines.

Reference to this paper should be made as follows: Xie, S.Q., Gan, J. and Wang, G.G. (2009) 'Optimal process planning for compound laser cutting and punch using Genetic Algorithms', *Int. J. Mechatronics and Manufacturing Systems*, Vol. 2, Nos. 1/2, pp.20–38.

Biographical notes: S.Q. Xie received his PhD from Huazhong University of Science and Technology (China), and the University of Canterbury (New Zealand). He is a Senior Lecturer at the University of Auckland, New Zealand. His research interests include intelligent mechatronics systems, vision techniques and applications, smart sensors and actuators, biomechatornics and biomedical robotics. He has more 15 years of teaching and research experience in mechatronics, medical robotics and advanced manufacturing technology. He is the editor of two international journals, guest editor, editorial board member, Reviewer and Scientific Advisory for many international journals and conferences. He has alsopublished more than 150 papers in refereed international journals and conferences.

J. Gan received his Bachelor Degree in Mechanical Engineering from China. He is currently a Master research student at the Department of Mechanical Engineering, University of Auckland. His research is in the area of computer aided process planning and optimisation algorithms.

G. Gary Wang joined the Simon Fraser University Surrey as an Associate Professor in 2008. He is active in research on design optimisation, design for manufacturing, and advanced manufacturing. With more than 80 technical publications, he has gained international recognition, especially for his work on design optimisation. He is the recipient of the 2005 National I.W. Smith award for creative engineering from Canadian Society of Mechanical Engineers (CSME), as well as the 2007 Rh Award from UM for outstanding research contribution in the Applied Science category. He is currently serving as an Editorial Board Member for *CSME Transactions and International Journal of Advanced Mechatronics and Robotics*.

1 Introduction

In sheet metal industry, there are a lot of small- and medium-sized job shops. These small- and medium-sized manufacturing companies have been facing keen competitive pressure in the market. This pressure has forced them to make every effort to shorten product development lead-time, improve production efficiency, approach high-quality standards, but at the same time cut down the costs.

Compound machines (Xie et al., 2001), or combined punch and laser cutting machines (Clark and Carbone, 1980), are developed to increase the functionality and efficiency of sheet metal machining. By using this compound method, the cutting process and punching process can be carried out sequentially or concurrently in the same CNC sheet metal compound machine without altering the fixtures. This compound manufacturing method takes the high efficiency and low cost advantages of the CNC punching and makes use of the high flexibility of the CNC cutting for complex contour cutting (Xie et al., 2001).

Normally sheet metal products design, processes planning, and manufacturing, are achieved by utilising different computer-aided software tools. They normally include a Computer-Aided Design (CAD) system taking care of product design, a computer aided process planning translates design information into the process steps and instructions to efficiently and effectively manufacture products which includes a Computer-Aided Path Planning (CAPP) system responsible for generating optimal tool paths and a Computer-Aided Nesting (CAN) system for optimal nesting of two-dimensional parts with regular and complicated shapes in order to effectively improve the utilisation ratio of sheet metal materials, and a Computer-Aided Manufacturing (CAM) system generating G and M-codes for different sheet metal processing machines (Xie and Xu, 2006).

The machining path planning and the nesting problem are both combinatorial optimisation problems which have been proven to have high computational complexity. In literature, the optimum path planning problem is traditionally addressed as the Travelling Salesman Problem (TSP) which has been the subject research for many decades (Hwang and Ahuja, 1992). The two dimensional sheet metal path planning problem is similar to the traditional TSP problem. It can be described as: Given a set of starting and end points of the machining operations, such as cutting or punching, the objective is to find the shortest path of all of the points of a cutting process or a punching process. The distance between each pair of points is symmetric in sheet metal path planning optimisation.

The nesting problem is defined as the problem of finding an efficient layout of products to be cut in a containing region without overlapping. Its main objective is to maximise the use of material. The nesting problem is characterised by the intrinsic difficulty of dealing with geometry, satisfaction of the no-overlapping and containment constraints, and complex computation. Currently, there are still lack of practical algorithms in industry to nest complex and multiplex parts, which impedes the realisation of effective automatic nesting (Xie et al., 2001).

Though process planning tools have been used on general sheet metal cutting or punching machines, as well as compound machines, the optimisation of process planning dedicated to compound machines, based on our literature search results, is limited. This work addresses the process-planning problems for the compound punch-laser machine by using Genetic Algorithms (GAs). The GAs are developed for optimisation of both the cutting or punching tool path and the nesting of two-dimensional parts with regular and complicated shapes. This enables our future work on the efficient integration of the two algorithms for finding a global optimal solution for both nesting and path planning.

2 Literature review

The branch and bound algorithm is an insertion algorithm (Hendrix et al., 2008) which does a truncated search on the entire solution space. The branching generates all the possible solutions available and bounding limits the search by not expanding a partial tour, if it is already longer than the best solution. Computational experience with this method shows that there is a difficulty in setting the bound which will limit the search without compromising optimality. The Clarke-Wright saving heuristic (Albano and Sapuppo, 1980) is derived from a more general vehicle routing algorithm by choosing a point as a hub. The initial solution starts with the salesman returning after visiting every other point. The construction terminates when the hub is connected to only two other points. The best performance of this algorithm is better than that of the greedy algorithm.

The tabu search (Cordeau et al., 2008) is based on the assumption that all locally optimal solutions are not good global solutions. Therefore, by minimising the randomised starting heuristic using a tabu list (a list close to the solution just found), a global optimum can be found. It is more effective than the original 2-opt and 3-opt since it only considers a tabu list instead of random starting points. The use of a tabu list in preference to the random starting heuristic restricts the algorithm which in some cases ‘misses’ the optimum path.

Meeran and Shafie (1997) implemented the convex hull boundary into system for the given set of points as its initial sub-tour. Then a local search heuristic is applied successively until all the given points are included in the path. Every point is identified in a family hierarchy, hence the relation between each point inside the convex hull boundary and the convex edges can be established without a combinatorial search.

Considering the contours of the sheet products, the sheet metal nesting problem can be classified as two types: regular nesting and irregular nesting.

Regular nesting is specifically considered the two-dimensional rectangular nesting problem. Lesh and Marks (2000) presented a Bottom-Left-Decreasing (BLD) algorithm that includes successive random perturbations of the original four decreasing orderings. Their experiments on both benchmark and randomly generated problems show that BLD substantially outperforms BLD as well as applying Bottom-Left (BL) to randomly chosen orderings. The Bottom-Left heuristic sorted the rectangles by decreasing width, but the heuristic is not competitive when sorted by decreasing height. Hopper and Turton (1999) solved a two-dimensional packing problem frequently encountered in the wood, glass and paper industry, which consists of nesting rectangular shaped parts onto a rectangular object while minimising the used object space. The nesting process has to ensure that there is no overlap between the rectangular parts, which are allowed to rotate by 90°.

Xie et al. (2007) discussed a heuristic nesting algorithm for irregular parts. They represented irregular shapes according to a set of non-overlapping rectangles. The system places each part in an orientation such that its length is larger than its height and always into the bottom-left most direction. The parts are then sorted by non-increasing part height. The shapes are packed into a rectangular scene in a raster fashion, building up layers of intermeshed packed shapes.

Dori and Ben-Bassat (1984) were the first to investigate the nesting of shapes within a polygon rather than a rectangle. They notified the assumption that the packing plane is infinite. The algorithm is only applicable to the nesting of congruent convex shapes. The problem involves cutting a number of similar but irregular pieces from a steel board, this is referred to as the template-layout problem. Considering the contours of the sheet products, the sheet metal nesting problem can be classified as two types: regular nesting and irregular nesting.

Wang and Xie (2005) addressed the process-planning problem for the combined punch-laser machine by integrating knowledge, quantitative analysis, and numerical optimisation approaches. The Ant Colony Optimisation (ACO) algorithms were developed in searching the optimal tool path. Experimental results showed that their proposed method can significantly improve the operation efficiency for the combined punch-laser machine.

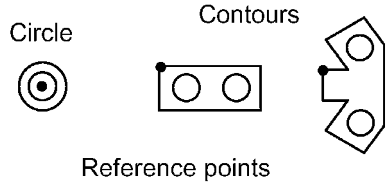
According to our literature review, the optimisation of process planning dedicated to compound laser cutting and punching machines, is still limited. The research in this area is now behind the fast development of the compound machines. This has significantly influenced the efficiency and productivity of the compound machines. This work attempts to develop GAs for the optimisation of sheet metal cutting and punching processes.

3 Genetic approach

The products to be machined in compound machines are normally small in size. They also have completely shapes, sizes and are also different from one another. Normally, one product could need both cutting and punching operations. According to our experience, for a product that requires both cutting and punching operations, the punching operations will be carried out first. This is due to the fact that extra fixtures are required if the product is cut first. This is one of the constraints that needs to be taken into account in process planning.

One assumption needs to be made first before discussing the proposed GA for path planning, which is each product is represented by a starting point for the cutting operation (Xie et al., 2001) as shown in Figure 1. For punching operations, the centre point of the feature to be punched is used as the reference point.

Figure 1 Reference points of different contours



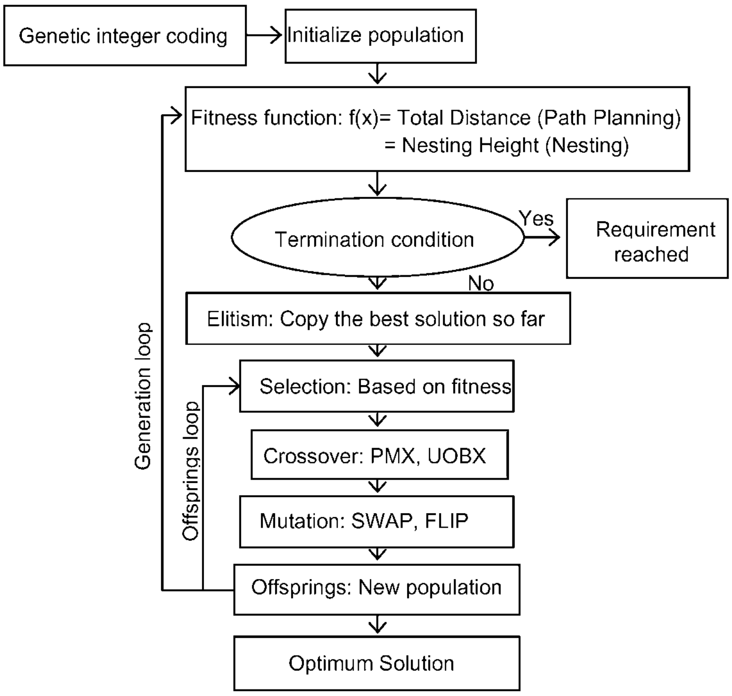
When nesting the contour of a product in computer, the contour is represented by a reference point and the other vertex of this contour is calculated according to the reference point. When manufacturing a sheet metal product, the machining tool first reaches the reference point of this product. Then, different machining processes are carried out such as auxiliary cutting path design. Figure 1 illustrates the reference point of three types of product contours.

The main structure of the proposed GA for path planning is showed in Figure 2. Genetic integer coding scheme creates an initial population for the GA to find the optimum path. The generation loop records the best results found in each generation, while the offspring loop reproduces on each generation to generate the next generation. The process is terminated in two different ways.

- *Manipulated stop*. The optimum requirement is reached. For example, in some cases, it is not necessary to have the optimist path since the computational complexity is proportion to the accuracy of the optimum.
- *Auto stop*. The algorithm stops when it is designed to stop. For instance, if the optimum result is estimated to appear after around the 90th generation, then we can design the algorithm to stop at the 100th generation.

The evolving process of the proposed GA contains two loops, the generation loop and offspring loop. The generation loop includes the offspring loop and it has a function of recording optimum path found in each generation and terminating GA when an ideal solution appears. On the other hand, offspring loop evolves the solution path in each generation in an improving means by using several operators, such as selection, crossover, mutation and replacement operators. In addition, elitism scheme copies the optimum solution found in each generation to the next generation population, which ensures GA evolving towards the optimum.

Figure 2 Structure of a Genetic Algorithm for path planning and nesting

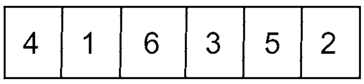


3.1 Genetic coding string

Each product is represented by an integer number. All the numbers combined forms a genetic string. The genetic string contains the sequence of the product machining process. That is which product is manufacturing first, which is second, and so on.

Figure 3 shows an example of representing a string of six products. The product numbers '4, 1, 6, 3, 5, 2' are considered in the same sequence for manufacturing these five products.

Figure 3 A genetic coding string for manufacturing five products



3.2 Initial population

The size of the population significantly influences the search space and the computational time needed to reach an optimal solution. According to the literature, the population size is considered equal to the length of the string (Hopper and Turton, 1999).

3.3 Fitness function

Since the objective of path planning is to find the shortest path for the machining operation, the fitness function equation is defined as following.

$$1/F = D_{1,2} + D_{2,3} + \cdots + D_{i,i+1} + \cdots + D_{n-1,n}$$

where $D_{i,i+1}$ is the distance between the ith and $(i + 1)$th product to be manufactured, n is the total number of the products to be manufactured.

On the other hand, the fitness function of genetic sheet metal nesting is:

$$1/F = \text{Height}$$

where Height is the final Height of all the nested area (we assume that the nesting direction is along the height).

The fitness function for any genetic string represents the effectiveness of the string in manufacturing process. During the genetic evolutionary process, the strings with higher fitness values will survive and those with lower fitness values will be eliminated by the survival of fittest principle in GA.

3.4 Genetic reproduction

The main purpose of reproduction is to preserve the good strings in the population and try to generate better strings in the population. To achieve the first task in the genetic reproduction process, an elitism scheme is applied into the proposed GA. The elitism scheme passes the fittest string into the next generation without any changes. However, the second task is achieved by using different genetic operators.

3.4.1 Selection

Two selection schemes are set to generate mating pool from the previous generation but with different mechanism. On the other hand, Tournament methods are more objective than Roulette Wheel method; it chooses a small group first, and then picks up the highest fitness within the small group.

In Roulette Wheel method, a real-valued interval, *Sum*, is determined as either the sum of the individuals' expected selection probabilities or the sum of the raw fitness values over all the individuals in the current population. Individuals are then mapped one-to-one into contiguous intervals in the range [0, *Sum*]. The size of each individual interval corresponds to the fitness value of the associated individual. For example, in Figure 4 the circumference of the roulette wheel is the sum of all six individual's fitness values. Individual five is the fittest individual and occupies the largest interval, whereas individuals 1 and 4 are the least fit individuals and have correspondingly smaller intervals within the roulette wheel. To select an individual, a random number is generated in the interval [0, *Sum*] and the individual whose segment spans the random number is selected.

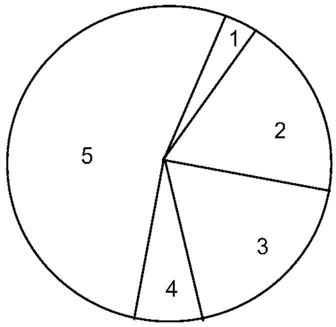
On the other hand, Tournament methods are more objective than Roulette Wheel method; it chooses a small group first, and then picks up the fittest string within the small group. It works in three steps:

Step 1: Set a random number t as the tournament size.

Step 2: Choose t individuals from the population.

Step 3: Return the fittest individual of these t.

Figure 4 Roulette wheel selection



3.4.2 Crossover

Two types of crossover operators are the alternatives: the Uniform Order-Based Crossover (UOX) and the Partially Matched Crossover (PMX). Each crossover operator is subjected to the probability of crossover.

Uniform Order-based Crossover (*UOX*): two parent paths are selected according to crossover probability for crossover. UOX operator creates a mask of equal length with parent whose position value is '1' or '0' generated randomly. Starting from the first position on the mask, if the value is '1', the two children inherit the same gene of the same position from the two parents respectively; while the value is '0', the first child receives the corresponding gene from the second parent and the second child receives the corresponding gene from the first parent. For example, as shown in Table 1, the middle part of two paths is crossover by UOX, while creating new chromosome 1 and 2, the first, third and sixth genes are inherited form parents, chromosome 1 and 2, respectively without change; while the second, fourth and fifth are switched. According to the values of the positions in the mask, the first value of the mask is one, so the 10th gene value of the new path 1 is copy the 10th gene value of path 1, while 10th gene of new path 2 gets the 10th gene value of path 2. The second value of the mask is 0, then, the 11th gene value of the new path 1 is the same as the 11th gene value of path 2.

Table 1 An example of applying Uniform Order-Based Crossover

Position of gene	*1st*	…	*10th*	*11th*	*12th*	*13th*	*14th*	*15th*	…
Chromosome 1	1	…	16	7	22	31	10	25	…
Chromosome 1	1	…	25	31	7	16	22	10	…
Mask			1	0	1	0	0	1	
New_Chrom 1	1	…	16	31	22	7	10	25	…
New_Chrom 1	1	…	25	16	7	22	31	10	…

Partially Matched Crossover (*PMX*): Instead of mask, two crossover points are generated randomly in PMX. Firstly, PMX proceeds by position-wise exchanges between the two points. Then it maintains the crossed parts and transfers the rest which has the same value with genes in crossed parts in each chromosome into the gene lost during crossing operation.

After the gene positions are picked up for crossover, such as 11th, 13th and 14th genes. The PMX operator changes the values between them as first step, following a elimination step which checks the same genes in one chromosome and replaces them by missing gene during the crossover process.

As shown in Table 2, 12th and 14th position is selected to be crossover points. The PMX operator processes in two steps. The integers between 12th and 14th positions of chromosome 1 and 2 are exchanged in the first step. In the second step, since the 10th and 11th gene has the same integer with 13th and 12th respectively, the 10th and 11th location is changed to the values which different to the integer value of crossing parts.

Table 2 An example of applying Partially Matched Crossover

Position of gene	…	*10th*	*11th*	*12th*	*13th*	*14th*	*15th*	…
Chromosome 1	…	16	7	22	31	10	25	…
Chromosome 1	…	25	31	7	16	22	10	…
Step 1				7	16	22		…
				22	31	10		…
New_Chrom 1	…	31	10	7	16	22	25	…
New_Chrom 1	…	25	16	22	31	10	7	…

3.4.3 Mutation

As the crossover operation operates on two strings and changes the sequence in which the products are to be manufactured, the mutation operator operates on one string. Two types mutation operators are applied in the proposed GA.

Flip operator: It operators within two flip sites of a chromosome. The values insides the sites are reordered (inversed). For example, in a path chromosome, from 10th to 15th genes are reversed by the Flip mutator as shown in Table 3.

Table 3 An example of applying two mutation operators

Position of gene	*1st*	…	*10th*	*11th*	*12th*	*13th*	*14th*	*15th*	…
Parent 1	1	…	16	7	22	31	10	25	…
Flip mutation	1	…	25	10	31	22	7	16	…
Swap mutation	1	…	16	10	22	31	7	25	…

Swap mutation: The values of two positions are switched under the Swap operator. Figure 4 presents the Swap mutation which exchange 11th and 14th gene value.

4 Case study

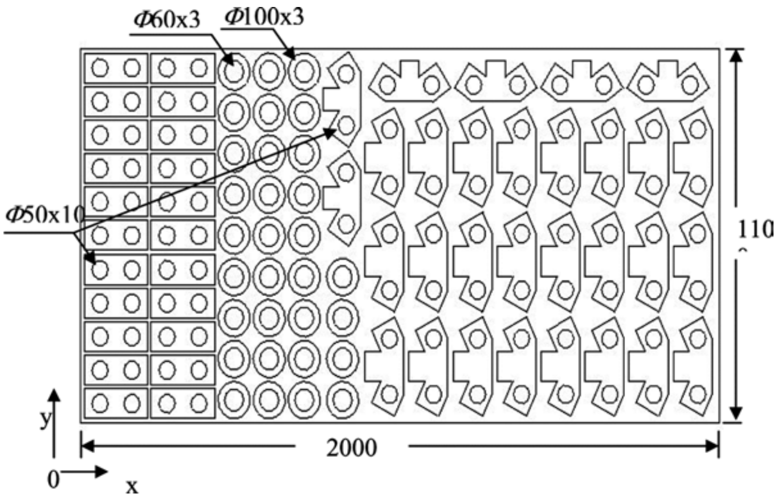
The case studies to be carried out are to test the feasibility of the proposed GAs for the two optimisation issues. The performance of the algorithms are compared with some of the existing algorithms developed in literature.

4.1 Sheet metal path planning problem

The purpose of this case study is to apply the proposed genetic path planning algorithm to industrial cases and to demonstrate the performance of the proposed algorithm in sheet metal path planning process. Wang and Xie (2005) has proposed an ant algorithm for solving the path planning issue for combined sheet metal machines. The results showed that the ant algorithm achieved a better performance than the normally used intuitive method. In this research, the ant algorithm is used as a comparison for the proposed genetic path planning algorithm.

Figure 5 shows an example that is selected from Wang and Xie (2005). It includes a batch of work pieces that is used for the case study. There are four different types of components to be cut or punched: 104 small holes of diameter Φ50, 31 holes of Φ60, 31 Φ100 contouring, 22 rectangular block contours and 30 contours of clips.

Figure 5 A batch of work piece



There are two machining methods used in the case study: punching and cutting. Following on the defined cutting and punching rules in Wang and Xie (2005), the 22 rectangular block and the 30 clips are to be cut, while the rest of the features are to be punched. Since there are three sizes of holes features Φ50, Φ60 and Φ100, three types of punch tools are used.

The case study is to investigate which algorithm of the two gives the shortest cutting/punching path. The optimisation result and the time that it takes to arrive the solution will be compared.

4.1.1 Genetic path planning

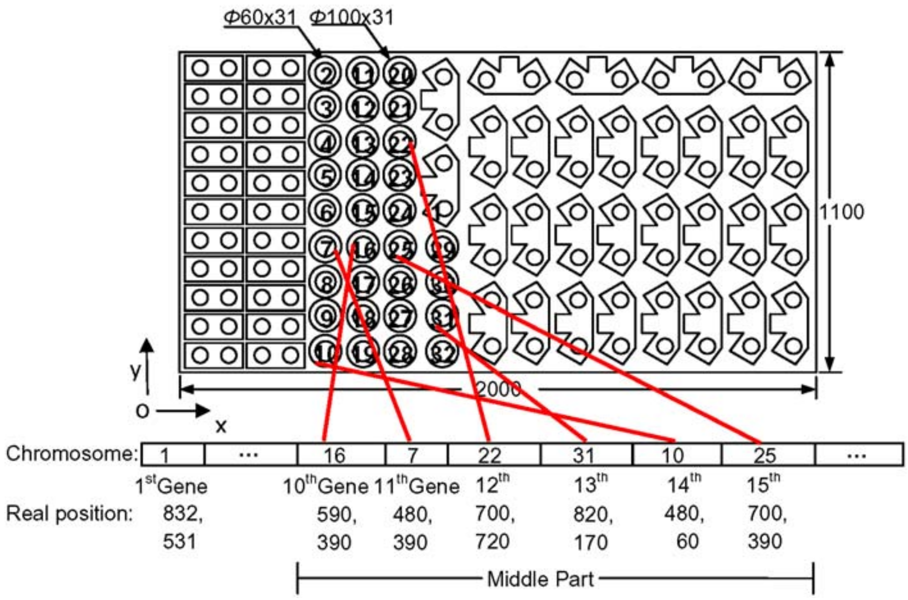
A GA is developed to find the shortest machining of the example as illustrated in Section 2. This algorithm includes the following key modules/steps: genetic encoding, fitness function definition, genetic reproduction and population replacement.

Genetic encoding

The coding strategy takes into consideration of the notion of a point and a path. For example, a point is represented as a gene in genetic coding string, which encodes to a positive integer. The points chosen depend on types of features. However, a path is comprised of this reference points and is encoded into a chromosome, which is thus subsequently the ordered set of integers. Following the chromosome string from left to right, the order of integers in a chromosome is therefore the same order of machining sequence.

Figure 6 illustrates the genetic path planning encoding according to the particular case. The name of each hole is represented as in the graph with integers from 1 to 32 as part identity integers. The chromosome in Figure 6 lists the possible machining sequence before an optimisation is carried out. Each feature in the chromosome is represented as a gene. The genes are represented by 32 integers. Furthermore, the location of an integer in a chromosome string indicates the sequence of the machining operation. For instance, part of this chromosome is defined as a middle path for explanation, 16, 7, 22, 31, 10 and 25. Integer 16 places at the position of 10th gene, which means feature 16 is the 10th feature to be machined in terms of this path chromosome. In a similar manner, feature seven is the 11th feature to be machined, then feature 22, 31 and 10. Feature 25 thus is the last one to be machined.

Figure 6 Genetic integer encoding scheme of path planning (see online version for colours)



Fitness function

According to the fitness equation defined in Section 2, the chromosome illustrated before, the fitness function of the middle parts is calculated as an example.

$$f(x) = 1/(d_{16,7} + d_{7,22} + d_{22,31} + d_{31,10} + d_{10,25})$$

where, $d_{16,7}$ is the distance between feature 16 and 7.

Genetic reproduction

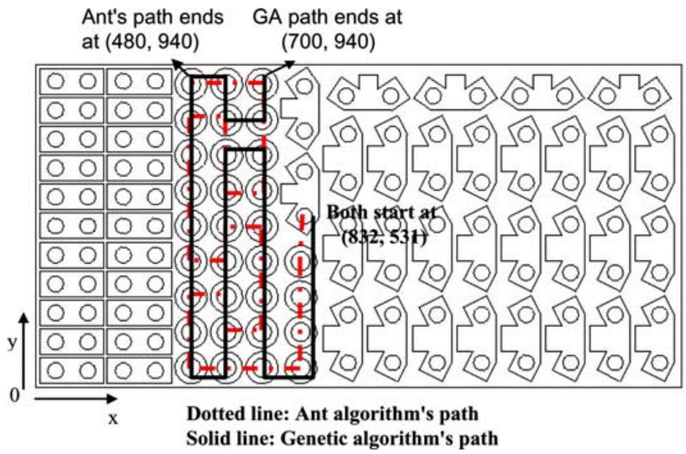
Several genetic operators are set up for reproduction. As selection methods, roulette wheel selection and tournament selection are alternatives. PMX and uniform crossover with high crossover probability are two options for genetic crossover. Two points swap mutation and flip mutation operators are designed for avoiding early convergence in the proposed GA.

4.1.2 Experimental results and discussion

Φ60

punching path optimisation. Figure 7 shows an optimal path (solid line) found by the GA, while the ant algorithm gives another different path as shown in the dotted line.

Figure 7 Optimum paths found by the Genetic Algorithm and the Ant Algorithm (see online version for colours)



The GA used elite of one individual, 81individuals in mating pool initially, a crossover probability of 0.8 and a mutation rate of 0.1.

Table 4 shows a comparison of the two machining routes generated by the ant algorithm and the proposed GA. The total lengths of the two machining routes are calculated. It can be found that the genetic algorithm produces a better search result. Figure 8 shows the path generated by the genetic algorithm. The algorithm is able to find a machining path that is 169 mm shorter than using the Ant Colony (AC) algorithm. The computational time is also shorter than the AC algorithm. In the experiment, the GA runs ten times with 500 loops. The results are recorded together with the AC algorithm as shown in Table 5.

As shown in Table 5, the average performance of the proposed GA is much better than the Ant Colony algorithm. Though both algorithms are evolutionary algorithms, the GA is superior to the AC algorithm with regard to time in this path planning. This is important especially when the optimisation problem becomes more complex.

Table 4 Performance comparison between the Genetic Algorithm and the Ant Colony algorithm

	Φ60 punching	
	Time (s)	*Path length* (mm)
Genetic Algorithm	15.51	3546
Ant Colony algorithm	23	3715

Figure 8 Optimum paths of punching and cutting machining process generated by ant algorithm and Genetic Algorithm (see online version for colours)

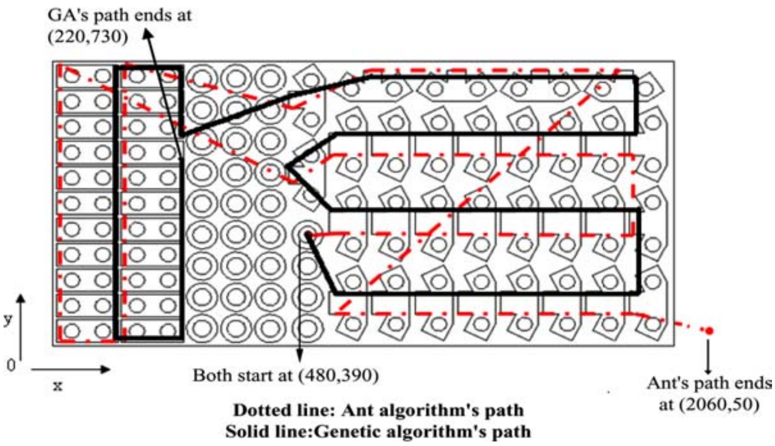


Table 5 Optimum punching path results by both Genetic Algorithm (GA) and Ant Colony (AC) algorithm

AC	Time (s)	22.89	22.87	22.83	22.73	22.66	22.72	22.83	23	22.89	23
	L (mm)	3611	3674	3542	3993	3827	3545	3611	3591	3452	3715
GA	Time (s)	16.14	15.95	15.89	15.75	16.15	15.95	15.78	15.67	15.92	15.51
	L (mm)	3499	3477	3540	3545	3560	3542	3568	3499	3495	3546

Genetic parameter investigation

An ideal choice of the genetic parameters guarantees a better exploration of the solution space and a quicker convergence towards the optimal solution. A good balance between crossovers and mutations is needed. Crossover allows the exploration of a wider neighbourhood of solutions, while mutation allows the diversification of the population. Similarly, a good trade-off between the size of the population and the number of generations is necessary to guarantee good quality solutions in short run times.

In this case study, there are in total 32 parts to be punched. The optimisation of the path of which a punch tool travels involves a large number of combinations. The results of different combinations of crossover and mutation operators, crossover possibility and mutation possibility, generation loops are listed in Appendix. The results indicate the combination of a PMX with crossover possibility of 0.9, a flip mutation with mutation possibility of 0.1 and initial population size of 61 yields the optimum path at around 20,000th generation.

Cutting path optimisation. The encoding scheme and genetic operators are set up as the same as punching path optimisation, but this GA has a large size of the initial population, 151, since large problem size. Table 6 presents different numbers of generations (loops) and population size are tested and summarised.

Table 6 Optimum cutting path results by both Ant Colony algorithm and Genetic Algorithm

Ant Colony algorithm	*Loops = 60*	*Loops = 80*	*Loops = 100*	*Loops = 120*
Time (s)	156.547	227.422	286.094	396.047
Length (mm)	9663.64	9716.586	9501.618	9492.532
Genetic Algorithm	*Pop = 81, Loops = 3000*	*Pop = 81, Loops = 5000*	*Pop = 151, Loops = 3000*	*Pop = 151, Loops = 2000*
Time (s)	216.844	544.688	197.016	288.016
Length (mm)	8914.5	8225.5	8637.7	8278.4

From Table 6, the proposed GA has better results than the ant algorithm no matter in time or quality.

Paths integration

It is time to consider the punching and cutting machining process at the same time, since in real manufacturing settings the two processes are operated sequentially. Starting point is the same in two algorithms to make comparison accurately. The optimum paths are plotted in the same graph below.

The total distance of cutting process using two algorithms are shows in Table 7.

Table 7 Optimum paths results of punching and cutting

Cutting optimisation	*Time* (s)	*Path length* (mm)
Genetic Algorithm	321.234	7911.7
Ant Colony algorithm	–	9348-212.64 = 9135.36

212.64 mm is the distance between last second point (1856, 110) and end point (2060, 50). The end point means the machining header's finial position after whole manufacturing process.

It is undeniable that a significant improvement is achieved by using the GA. A total machining path of 1223.66 mm is shortened in this example. This is due to the constraint on the Ant algorithm that the path has to finish at (2060, 50). The GA shows time advantage as well. By running the ant algorithm ten times, the results are contrast to the GA result. During the investigation of both algorithms, the GA finds optimum quickly while ant algorithm improves smoothly with the time.

Results presented of case study carried out in the preceding section suggest that the proposed GA could be an effective optimisation methodology. GA has been proved its performance in sheet metal path planning optimisation problem.

4.2 Sheet metal nesting

The difference between the sheet metal path planning and nesting is the fitness function. A bottom left heuristic algorithm is applied in sheet metal nesting fitness function. We only concern the rectangular sheet metal nesting problem in this paper.

The case study is carried out in a six rectangular with various sizes nesting problem. The genetic coded nesting string contains only integers which is a sequence of parts coding identification. A typical string is formulated as shown in Table 8.

Table 8 A string coded by Genetic Algorithm and dimensions of the nesting parts

5	*4*	*1*	*2*	*3*	*6*
15 × 25	20 × 10	10 × 30	15 × 35	25 × 20	20 × 10

In this particular string, the six elements represent the six nesting parts. In view of the complexity in the optimal placement of nesting parts, the present approach considers the nesting parts in a sequential manner. The No. 5 part of size 15 × 25 is first placed in the heuristic algorithm, and then No. 4 part is placed second. Following the sequence, the No. 6 part is the last to be nested into the sheet.

The heuristic algorithm considers each rectangular part in the same order that appears in the string for generating a nested pattern. For nesting any particular parts, these parts are arranged from the bottom left position of the sheet. It is ensured that the part does not overlap with the previous parts or cross the boundary of the sheet. After placing each part on the sheet, new positions for the next part to be placed are identified. The positioning of part on the sheet is based on a two dimensional translation. For translating the part to any node, the bottom-left corner of the part, chosen as reference point, is coincided with that node. The following procedure is adopted for the generation of a placement of the coding string shown in Table 8.

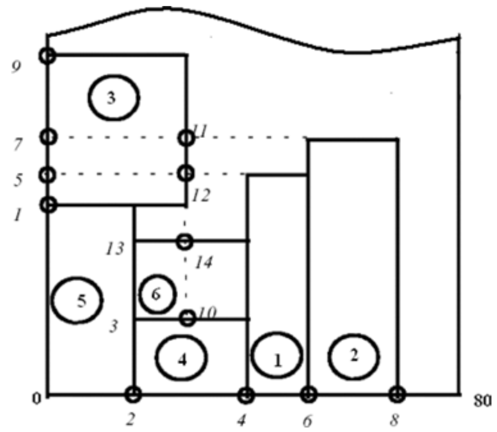
Step 1: The first rectangular part in the sequence, i.e., the fifth rectangular part (15 × 25 mm), is translated to the bottom-left corner of the first rectangular sheet in the sequence, i.e., the third sheet, in such a way that the first dimension (15 mm) is positioned along the x-axis and the second dimension (25 mm) along the y-axis. After translating this part, positions, e.g., left-top corner and right-bottom corner of the part are obtained for positioning the next part. These two nodes are represented as nodes 1 and 2 in Figure 9.

Step 2. The next part in the sequence is the second part (20 × 10 mm), which is translated to node two since this particular node is located at the bottom-left position on the remaining sheet. This part is positioned by 20 mm side along the x-axis and the 10 mm side parallel to the y-axis, as shown in Figure 9. Nodes 3 and 4 are identified as new nodes for translating the next part on the sheet. In a similar placing manner, the first and second parts are also nested, and nodes 5–8 are identified as the new nodes for arranging the next part. Nodes 5 and 7 are obtained by projecting the top horizontal edges of the parts 1 and 7 onto the vertical edge of the sheet.

Step 3: The next part in the string sequence is No. 3. It is translated to the first node where the part does not overlap with those that are already nested and ensures the bottom-left-most position. After translating the part, nodes 9–12 are identified as new nodes. Node ten is obtained by projecting the right vertical edge of the fifth part onto the top horizontal edge of the second part. Nodes 11 and 12 are projections of horizontal edges of part 1 and 7 on the right vertical edge of part 3. In a similar way, the next and the last part in the sequence, is translated to the third node. Node 14 is obtained by projecting the right vertical edge of part 5 onto the top horizontal edge of part 3.

With the bottom left heuristic, the case of above reaches a material utilisation ratio of 75% using the proposed genetic nesting approach.

Figure 9 A heuristic placement considering the coding string: 5 4 1 2 3 6



4.3 Discussion

The case studies show that the proposed GAs are able to provide good solutions for sheet metal machining problems. It has been shown that the GA is able to produce comparable results than the AC algorithm that was originally developed in our research group. The experimental results of the proposed GA show that the algorithm is able to find efficient machining path. This is achieved by designing effective genetic operators, which can improve algorithm performance towards optimum.

A genetic nesting algorithm is also proposed for solving a rectangular nesting problem. However, this area requires further work to accommodate the nesting of parts with irregular shapes (Xie and Xu, 2006), which is more often encountered in sheet metal manufacturing process. The problem will be more complex as the shape of parts becomes more complicated. Future study is to be carried out to explore how a GA can be developed for this nesting issue. AC algorithm will be also an alterative methodology for the nesting of irregular parts.

Moreover, it is well known that the sheet metal path planning process is conducted after a computer aided nesting process, which generates a compact layout based on minimising the wastage of the sheet materials. However, this does not take consideration of the efficiency of manufacturing process. It is better to put the products, which have the same operation type, together from the operation efficiency point of view. Future study on the global optimisation of both path planning and nesting integration is a valuable research topic.

5 Conclusion

A new search methodology based on evolutionary process is introduced and studied in this work and its application to the solution of a classical optimisation problem in sheet metal manufacturing industry, sheet metal machining path planning and sheet metal nesting issues.

This research investigates the optimal process planning issues in sheet metal product development. GAs are proposed and developed for the sheet metal path planning issue and nesting issue. Case studies are carried out to demonstrate the performance of the proposed algorithms on the path planning optimisation issues.

For the sheet metal path planning optimisation problem, the proposed GAs are tested in a sheet metal industrial case against the AC algorithm. The proposed GAs are programmed in Matlab. The experimental results are compared with the results from the AC algorithm. The performance of the proposed GA shows better performance than the AC algorithm.

References

Albano, A. and Sapuppo, G. (1980) 'Optimal location of two-dimensional irregular shapes using Heuristic search methods', *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 10, No. 5, pp.242–248.

Clark, S.C. and Carbone, V.T. (1980) *Laser Cutting Head Attachment for Punch Press*, United States Patent and Trademark Office, Houaille Industries, Inc., Patent No. 4201905, USA.

Cordeau, J.F., Laporte, G. and Pasin, F. (2008) 'Iterated tabu search for the car sequencing problem', *European Journal of Operational Research*, Vol. 191, No. 3, pp.945–956.

Dori, D. and Ben-Bassat, M. (1984) 'Efficient nesting of congruent convex figures', *Communications of the ACM*, pp.228–235.

Hendrix, E.M.T., Casado, L.G. and García, I. (2008) 'The semi-continuous quadratic mixture design problem: description and branch-and-bound approach', *European Journal of Operational Research*, Vol. 191, No. 3, pp.803–815.

Hopper, E. and Turton, B. (1999) 'A genetic algorithm for a 2D industrial packing problem', *Computers and Industrial Engineering*, Vol. 37, pp.375–378.

Hwang, Y. and Ahuja, N. (1992) 'Gross motion planning: a survey', *ACM Computing Surveys*, Vol. 24, No. 3, pp.219–291.

Lesh, N. and Marks, J. (2000) *Human Guided simple Search*, A Mitsubishi Electric Research Laboratory, http://www.merl.com

Meeran, S. and Shafie, A. (1997) 'Optimum path planning using convex hull and local search heuristic algorithms', *Mechatronics*, Vol. 7, No. 8, pp.737–756.

Wang, G.G. and Xie, S.Q. (2005) 'Optimal process planning for a combined punch and laser cutting machine using ant colony optimisation', *International Journal of Production Research, International Journal of Production Research*, Vol. 43, No. 11, June, pp.2195–2216.

Xie, S.Q. and Xu, X. (2006) 'A STEP-compliant integrated system for sheet metal parts', *International Journal of Integrated Manufacturing Technology*, Vol. 19, No. 6, pp.627–638.

Xie, S.Q., Tu, Y.L., Liu, J.Q. and Zhou, Z.D. (2001) 'Integrated and concurrent approach for compound sheet metal cutting and punching', *International Journal of Production Research*, Vol. 39, No. 6, pp.1095–1112.

Xie, S.Q., Wang, G.G. and Liu, Y. (2007) 'Nesting of two-dimensional irregular parts: an integrated approach', *International Journal of Computer Integrated Manufacturing*, Vol. 20, No. 8, pp.741–756.

Appendix: Experimental results of investigation genetic parameters

PMX	Partially Matched Crossover
UOX	Uniform Order-Based Crossover
SWAP	Swap two random positions in a single route
FLIP	Flip function to rearrange the order in single route
Loop	Generations
Pc	Possibility of crossover
Pm	Possibility of mutation
Popsize	Initial population pool size

PMX CROSSOVER + SWAP MUTATION, FLIP MUTATION
Loop = 10,000; Pc = 0.9; Pm = 0.05 – 0.1;
Popsize = 31; Parts = 32. Distance = 4.0384e+003
Loop = 5000; Pc = 0.8; Pm = 0.1;
Popsize = 51; Parts = 32. Distance = 3.8703e+003
Loop = 10,000; Pc = 0.9; Pm = 0.05 – 0.1;
Popsize = 31; Parts = 32. Distance = 3.8730e+003
Loop = 5000; Pc = 0.8; Pm = 0.1;
Popsize = 81; Parts = 32. Distance = 3.6694e+003
Loop = 5000; Pc = 0.8; Pm = 0.1;
Popsize = 81; Parts = 32. Distance = 3.8623e+003
Loop = 5000; Pc = 0.8; Pm = 0.1;
Popsize = 81; Parts = 32. Distance = 3.9582e+003
PMX CROSSOVER + SWAP MUTATION Loop = 5000
Loop = 2000; Pc = 0.9; Pm = 0.05;
Popsize = 61; Parts = 32.
Distance = 5.3182e+003
UOX CROSSOVER + SWAP MUTATION
Loop = 3000; Pc = 0.9; Pm = 0.1;
Popsize = 61; Parts = 32. Distance = 3.7232e+003
PMX + FLIP MUTATION
Loop = 2000; Pc = 0.7; Pm = 0.15;
Popsize = 61; Parts = 32.
Distance = 6.8509e+003
PMX + SWAP MUTATION
Loop = 2000; Pc = 0.7; Pm = 0.15;
Popsize = 61; Parts = 32.
Distance = 6.2296e+003
PMX CROSSOVER + FLIP MUTATION

Appendix: Experimental results of investigation genetic parameters (continued)

Loop = 5000; Pc = 0.9; Pm = 0.1;
Popsize = 61; Parts = 32. Distance = 3.5682e+003
PMX CROSSOVER + FLIP MUTATION
Loop = 20000; Pc = 0.9; Pm = 0.1;
Popsize = 61; Parts = 32. Distance = 3.5227e+003