Yeung, Ka Yiu and Chen, Xun

3D freeform component fixture design optisation using a genetic algorithm

3D FREEFORM COMPONENT FIXTURE DESIGN OPTISATION USING A GENETIC ALGORITHM

K.Y. Yeung[1], X. Chen[2]
[1] University of Nottingham, Nottingham, NG7 2RD, UK
[2] University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK

ABSTRACT

This paper details an application of genetic algorithms (GA) developed for the optimisation of fixture locator positioning for 3D freeform components. Based on the information of the workpiece, a genetic algorithm based approach is applied to determine the most statically stable fixture configuration from a large number of possible candidates. The preliminary implementation is introduced to demonstrate the ability of GA in automated fixture design.

Keywords Genetic Algorithm, Fixture Design Optimisation

1 INTRODUCTION

A fixture is a device, which is used to securely restrain a workpiece in desired position with locators and clamp(s). Therefore the orientations of a workpiece can be maintained during machining, assembly or inspection. The locators and clamp arrangement is critical to produce accurate products repeatedly. When designing a fixture, it relies significantly on the designer's expertise and experience. Often, the cost of designing and fabricating fixtures can be a significant amount of the total manufacturing cost and there is no assurance that the results are optimal or near optimal.
A number of literatures indicated a general consensus that genetic algorithms (GA) are capable of creating fixture designs and thought to have potentials to improve existing fixture designs. In addition, GA is able to reduce the dependency on human designer expertise to produce high quality fixture designs [2]. A reasonable amount of computational time is required to execute a GA search [3] but this mainly depends on the structure of the actual artificial chromosomes [4-6]. Since the GA deals with only the design variables and the objective function value (fitness function) for a particular fixture layout, no gradient to auxiliary information are needed, it allows implementation to be relatively straightforward and to provide easy manoeuvrability [1,4].
The fitness function for the GA applied in this study focuses on error minimisation of workpiece datum features, which are related to geometric variations that may exist in the physical datum features of the component before or after machining. The datum geometric variations will result in contact point errors between the locators and the workpiece. At present there are no standard ways to model a particular problem. In order to investigate the suitability of GA for Automated Fixture Design (AFD), two different types of 3D components is used to evaluate the same algorithms.

2 DESCRIPTION OF THE TEST CASES

Two different 3D components have been selected to test applications of genetic algorithms. case 1: A rectangular box with a truncated corner, and case 2: A turbine blade with locators and clampers on the aerofoil. For both cases the component are created from Pro-Engineer then converted into the render (.slp) and 2D/3D vector graphics format based on the Initial Graphics Exchange Specification (.igs) file format. The .igs file then loaded into ABAQUE to generate a surface mesh; the mesh points are extracted and saved in the form of a text file (.txt). From the graphical user interface (GUI) in Matlab users are allowed to select any render (.slp) file as input information source for the component. After loading the mesh points from text file, which show all the possible clamping and locator points. The component is to be restrained by six locators and a single clamp.
With the component and mesh information input into Matlab, users still required to enter the GA parameters into the GA toolbox before conducting a search for the optimum locator configuration. According to the locator configuration, the minimum maximum clamping forces are calculated. If the clamping and locators position does not fulfil all requirements of the fixture design, users can modify the mesh size or area to refine GA search region, until all the constraints are satisfy before finalising the design. For details of the system operation please follows the flow chart in figure 1.

3 GENETIC ALGORITHM

Genetic algorithms are based on an analogy with biotic genetics and natural selection; it exploits the idea of survival of the fittest and an interbreeding population abstracting the adaptation ability from nature to form a robust search mechanism to solve optimisation problems. It comprises a set of individual elements, which are known as the population and a set of biologically simulated operators to produces next generations. The fixture locator configuration optimisation was implemented using MATLAB® and genetic algorithm direct search toolbox 1.0. The parameters used for this application was determined after trials with different combinations of mutation, crossover and selection methods. Iterative process of GA started by creates the initial generations of solution (population size = 400). This process would continue until a predefined maximum number of generations were reached (500), or until there was no appreciable improvement in the fitness function after 200 consequent iterations. With each new generation, the populations were getting closer to an optimal layout. Once the search was complete, the best layout from all the generations was taken as the optimal solution.

Constraints for solution of locator configuration:

(1) All six locators need to be in contact with the component.

(2) The six locator configuration needs to allow the component to fix into place before installing the clamp.

String representation and convert into solution: The string (artificial chromosomes) corresponds to locator positions based on ratio of the component perimeter from a reference point. Since there are only six locator points, the length of artificial chromosomes consists of six parameters. L1, L2, L3, L4, L5 and L6 indicate fixture locator number 1, 2, 3, 4, 5 and 6. The encoded string for this research uses a real number between 0-1 to represent the locators' position in relation to the workpiece frame for the initial population as showed in figure 2. In order to reduce the Hamming distances [8] of the coding structure, some redundancy is introduced into the artificial chromosomes. The applied method adapted continuous looping search space technique. Although the initial population only uses a real number between 0-1 to represent the position of each locator, subsequence population could mutate to any real number, but it would still remain in the same problem space. Hence 1.34 would equal to 0.34 and - 0.2 is equal to 0.8.
Constraints number 1 and 2 are already built into the individual solution, this ensures all six locators are in contact with the component. The minimum distance between locators and clamp needs to prevent collision between the locator and clamp. Therefore feasible perimeter for positioning the locator is equal to perimeter subtracts the area that is too close to the edge.

Fitness function: The fitness function is the locator matrix determinant. The determinant of a locator matrix W_L is denoted by $| W_L |$.

The given locator number n, locating normal vector $[a_i, b_i, c_i]$ and locating position $[x_i, y_i, z_i]$ for each locator, $i = 1,2,\ldots,n$ ($n = 6$)

$$W_L = \begin{bmatrix} a_1 & b_1 & c_1 & c_1y_1-b_1z_1 & a_1z_1-c_1x_1 & b_1x_1-z_1y_1 \\ . & . & . & . \quad . & . & \\ a_i & b_i & c_i & c_iy_i-b_iz_i & a_iz_i-c_ix_i & b_ix_i-z_iy_i \\ . & . & . & . \quad . & . & \\ a_n & b_n & c_n & c_ny_n-b_nz_n & a_nz_n-c_nx_n & b_nx_n-z_ny_n \end{bmatrix} \quad (1)$$

Example of calculates the 3×3 matrix determinant.

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}. \quad (2)$$

Using the cofactor expansion on the first row of the matrix we get:

$$\det(A) = a\begin{vmatrix} e & f \\ h & i \end{vmatrix} - b\begin{vmatrix} d & f \\ g & i \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix} = aei - afh - bdi + cdh + bfg - ceg. \quad (3)$$

Selection: One of the issues often encountered with generic algorithms is premature convergence. A good solution found early in the search tends to grow in number exponentially over a few consecutive generations. As a result, the algorithm will limit the search space and unable to find better solution. To prevent this, a method of selection called stochastic uniform [7] was used. This method lays out a line in which each parent corresponds to a section of the line of length proportional to its expectation. The algorithm moves along the line in steps of equal size, one step for each parent. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size.

Crossover: Crossover is the operator that creates next generation (offspring) of solution from parent locator configurations. In this application, intermediate operator, offspring parameters are obtained by taking a weighted average of the parents. The function creates the child from parent 1 and parent 2 using the following formula:

child = parent1 + rand * Ratio * (parent2 - parent1) (4)

Ratio is weight represented by a single parameter. If all the entries of ratio lie in the range 0-1, the children produced are within the hypercube defined by placing the parents at opposite vertices. If ratio is not in that range, the children might lie outside the hypercube. If ratio is a scalar, then all the children lie on the line between the parents.

Mutation: Mutation is used to generate new parameters, thus it not only provides randomness and improvement to the search but also selects a parameter to altars the values, usually a very low probability. A new random number is generated from a Gaussian distribution with the range for that entry to replace the mutated parameter. Gaussian distribution (1.0, 0.3) centred on zero. The variance of this distribution can be controlled with two parameters.

4 RESULTS AND DISCUSSION

In order to compare and inspect the performance of GA for different types of component 10 trials were conducted. Table 1 and table 2 showed examples of the results for each test case, while figure 3 and figure 4 illustrated the examples of the results from each case. There is a general consistence of yielding solutions in similar configurations for all trials and for both cases. As shown in figure 3 and 4, the blue crosses represent all the possible locator and clamping position. The red squares in the diagram represent the positions of locators, which suggests that the applied genetic algorithms are searching toward the optimal solution and are getting very close to finding the best possible fixture locator configuration for the specified component. The green crosses represent possible clamping position, while the red cross represent the best clamping position.

On average there is no significant improvement on the quality of results yields after 350 generation. The fitness values yielded from both assessments are very consistent, supported with the small value of standard deviation (SD).

Comparing this research work with other known results and comments from fixturing expertises, which suggests the fitness function applied is capable to yields feasible solution for freeform components. At present this research only solves the locators configuration for 3D freeform component with single clamp, the same evaluation methods can be extended to take into consideration of multi-clamping points to reduce clamping forces.

Further investigations can be carried out applying different mesh sizes as well as specifically in fine-tuning for the performance of GAs, by experiments with different GA parameters and different coding schemes for the artificial chromosomes. Performances are measured in terms of efficiency as well as quality of the final solution. Hence yield the most repeatable solution using the least amount of computational time as well as reaches the optimal solution with the least amount of generation and population size.

Table 1: Result summaries for test case 1

Trial Number	Locator co-ordinate			Clamping position [x, y, z]	Fitness Value
	x	y	z		
1	10 15.0218 15.025 10 37.6518 10	12.5566 10 10 47.0569 15.1494 45.3846	55.6046 12.2894 57.7106 56.117 10 12.6647	37.77778 41.11111 51.11111	0.06853
2	15.0250 15.0218 12.3480 10.0000 10.0000 10.0000	10.0000 10.0000 15.1435 47.4465 44.8711 12.5471	57.7106 12.2894 10.0000 14.4007 57.6515 14.4118	31.11111 47.77778 51.11111	0.068559
3	15.025 15.0218 10 10 12.348 10	10 10 12.5471 47.4465 15.1435 30	57.7106 12.2894 14.4118 14.4007 10 57.3353	31.1111 47.7778 51.1111	0.06805

Table 2: Result summaries for test case 2

Trial Number	Locator co-ordinate			Clamping position	Fitness Value
	x	y	z		
1	5.3768 -10.6500 7.7394 8.4527 9.7725 -10.8980	2.8485 -7.3332 8.7720 -4.9590 10.5738 -2.6915	59.0531 57.9822 62.1622 22.7273 25.5004 31.0878	2.4179 -3.2192 49.4921	0.99258
2	9.7725 8.4527 5.3768 7.7394 -10.65 -10.898	10.5738 -4.959 2.8485 8.772 -7.3332 -2.6915	25.5004 22.7273 59.0531 62.1622 57.9822 31.0878	2.4179 -3.2192 49.4921	0.99258
3	5.3768 7.7394 8.4527 9.7725 -10.898 -10.65	2.8485 8.772 -4.959 10.5738 -2.6915 -7.3332	59.0531 62.1622 22.7273 25.5004 31.0878 57.9822	2.4179 -3.2192 49.4921	0.99258

5 CONCLUSIONS

The fixture optimisation using genetic algorithms has been proven to be effective at finding high quality solutions, which could be integrated into industry automated fixture design (AFD). This paper has demonstrated the robustness of GA with two different components in application of GA fixture design. There are sufficient amount of studies confirmed that genetic algorithms are capable of creating high quality fixture designs and have the potential to improve existing fixture designs. This will reduce the dependency on human designer expertise to produce high quality fixture designs even for complex components.

REFERENCES

[1] WU N.H., CHAN K.C. (1996). *A genetic algorithm based approach to optimal fixture configuration.* Computers & Industrial Engineering. Vol. 31, pp. 919-924

[2] SUBRAMANIAM V., SENTHIL KUMAR A., SEOW K.C. (1999). *Conceptual Design of Fixtures using Genetic Algorithms.* The International Journal of Advanced Manufacturing Technology. Vol. 15, pp. 79-84

[3] LI B., SHIU B.W. (2001). *Principle and Simulation of Fixture Configuration Design for Sheet Metal Assembly with Laser Welding. Part 2: Optimal Configuration Design with the Genetic*

Algorithm. The International Journal of Advanced Manufacturing Technology. Vol. 18, pp. 276-284

[4] KRISHNAKUMAR K., MELKOTE N., SHREYES N. (1999). *Machining fixture layout optimization using the genetic algorithm.* International Journal of Machine Tools and Manufacture. Vol. 40, pp 579-598

[5] VALLAPUZHA S., DE METER E.D., CHOUDHURI S., KHETAN R.P. (2002). *An investigation of the effectiveness of fixture layout optimization methods.* International Journal of Machine Tools and Manufacture. Vol. 42, pp. 251-263

[6] VALLAPUZHA S., DE METER E.D., CHOUDHURI S.,. KHETAN R.P (2002). *An investigation into the use of spatial coordinates for the genetic algorithm based solution of the fixture layout optimization problem.* International Journal of Machine Tools and Manufacture. Vol. 42, pp 265-275

[7] Genetic Algorithms and Direct Search Toolbox

http://www.mathworks.com/access/helpdesk/help/toolbox/gads/index.html

Last check: 8 November 2005

[8] HILLIER R. (1999) *Genetic algorithms: AN investigation into Novel evolution methods with particulate reference to the De Jong test suite*, Dissertations. University of Derby

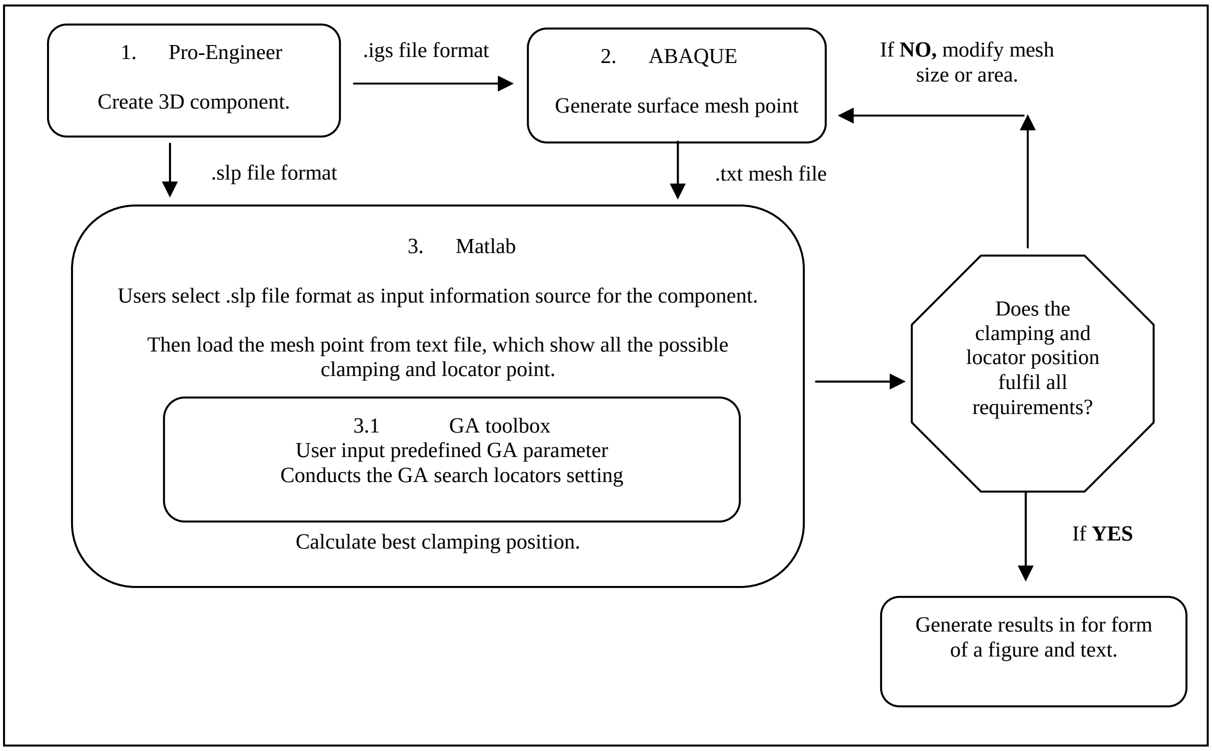

Figure 1: Flow Chart of the system operation

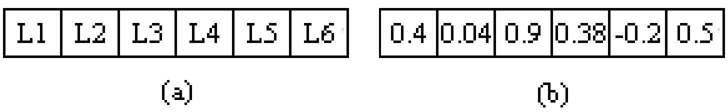

Figure 2: (a) Structure of the artificial chromosome (b) example of the artificial chromosome

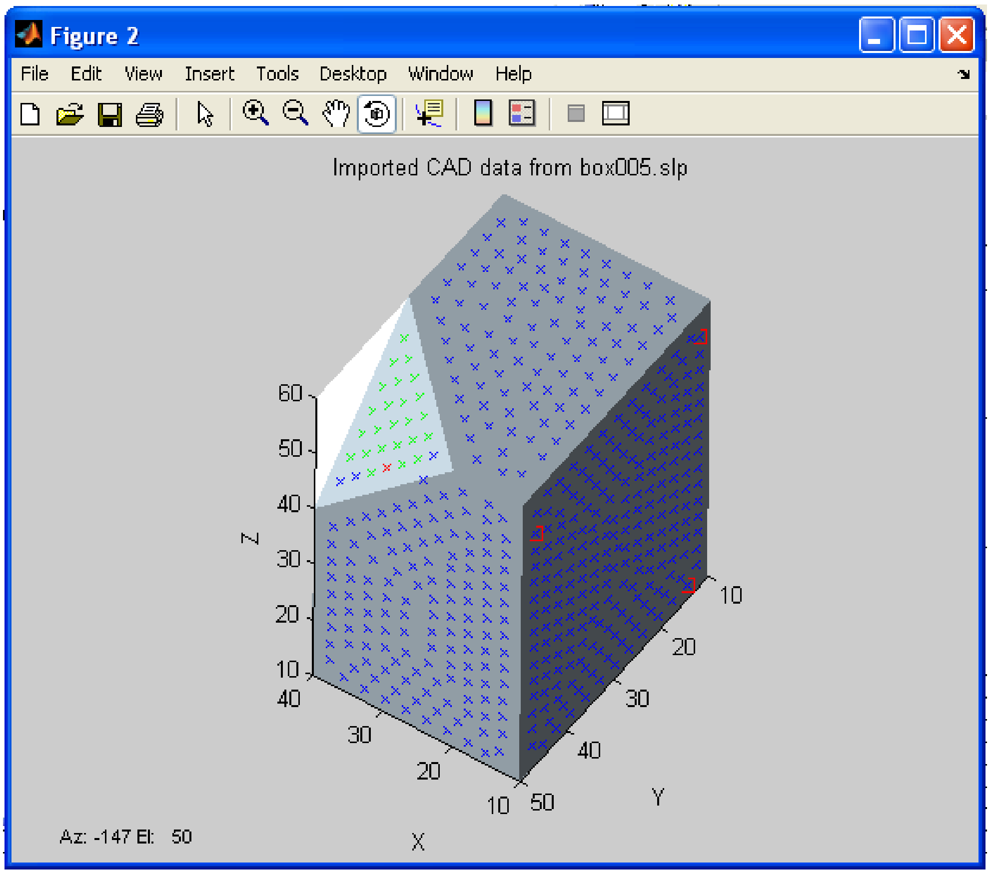

Figure 3: Illustration of an example result from case 1

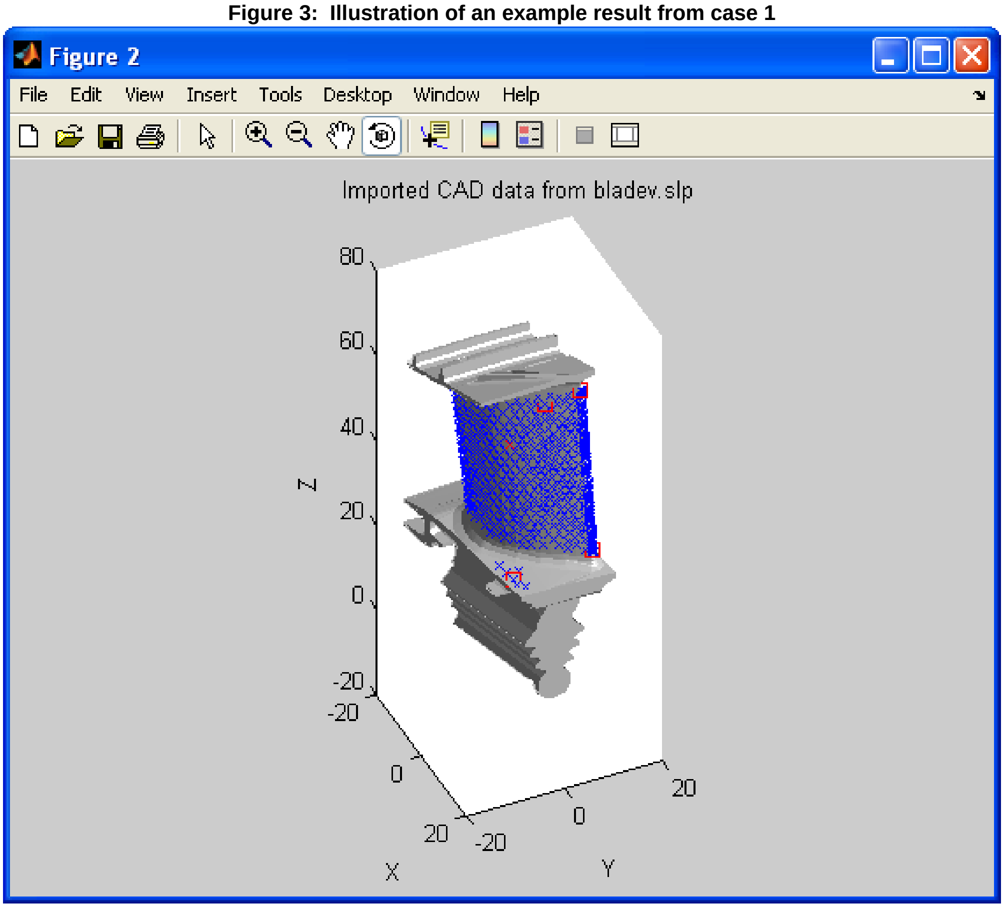

Figure 4: Illustration of an example result from case 2