Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers in Industry 57 (2006) 112–120

COMPUTERS IN INDUSTRY

www.elsevier.com/locate/compind

# Machining fixture locating and clamping position optimization using genetic algorithms

Necmettin Kaya *

*Department of Mechanical Engineering, Uludag University, Görükle, Bursa 16059, Turkey*

Received 8 July 2004; accepted 26 May 2005
Available online 6 September 2005

**Abstract**

Deformation of the workpiece may cause dimensional problems in machining. Supports and locators are used in order to reduce the error caused by elastic deformation of the workpiece. The optimization of support, locator and clamp locations is a critical problem to minimize the geometric error in workpiece machining. In this paper, the application of genetic algorithms (GAs) to the fixture layout optimization is presented to handle fixture layout optimization problem. A genetic algorithm based approach is developed to optimise fixture layout through integrating a finite element code running in batch mode to compute the objective function values for each generation. Case studies are given to illustrate the application of proposed approach. Chromosome library approach is used to decrease the total solution time. Developed GA keeps track of previosly analyzed designs, therefore the number of function evaulations are decreased about 93%. The results of this approach show that the fixture layout optimization problems are multi-modal problems. Optimized designs do not have any apparent similarities although they provide very similar performances.


*Keywords:* Fixture design; Genetic algorithms; Optimization

## 1. Introduction

Fixtures are used to locate and constrain a workpiece during a machining operation, minimizing workpiece and fixture tooling deflections due to clamping and cutting forces are critical to ensuring accuracy of the machining operation. Traditionally, machining fixtures are designed and manufactured through trial-and-error, which prove to be both expensive and time-consuming to the manufacturing process. To ensure a workpiece is manufactured according to specified dimensions and tolerances, it must be appropriately located and clamped, making it imperative to develop tools that will eliminate costly and time-consuming trial-and-error designs. Proper workpiece location and fixture design are crucial to product quality in terms of precision, accuracy and finish of the machined part.

Theoretically, the 3-2-1 locating principle can satisfactorily locate all prismatic shaped workpieces. This method provides the maximum rigidity with the minimum number of fixture elements. To position a part from a kinematic point of view means constraining the six degrees of freedom of a free moving body (three translations and three rotations). Three supports are positioned below the part to establish the location of the workpiece on its vertical axis. Locators are placed on two peripheral edges and intended to establish the location of the workpiece on the $x$ and $y$ horizontal axes. Properly locating the workpiece in the fixture is vital to the overall accuracy and repeatability of the manufacturing process. Locators should be positioned as far apart as possible and should be placed on machined surfaces wherever possible. Supports are usually placed to encompass the center of gravity of a workpiece and positioned as far apart as possible to maintain its stability. The primary responsibility of a clamp in fixture is to secure the part against the locators and supports. Clamps should not be expected to resist the cutting forces generated in the machining operation.

For a given number of fixture elements, the machining fixture synthesis problem is the finding optimal layout or positions of the fixture elements around the workpiece. In this paper, a method for fixture layout optimization using genetic algorithms is presented. The optimization objective is to search for a 2D fixture layout that minimizes the maximum elastic deformation at different locations of the workpiece. ANSYS program has been used for calculating the deflection of the part

* Tel.: +90 224 4428176; fax: +90 224 4428021.
*E-mail address:* necmi@uludag.edu.tr.

0166-3615/$ – see front matter 
doi:10.1016/j.compind.2005.05.001

under clamping and cutting forces. Two case studies are given to illustrate the proposed approach.

## 2. Review of related works

Fixture design has received considerable attention in recent years. However, little attention has been focused on the optimum fixture layout design. Menassa and DeVries [1] used FEA for calculating deflections using the minimization of the workpiece deflection at selected points as the design criterion. The design problem was to determine the position of supports. Meyer and Liou [2] presented an approach that uses linear programming technique to synthesize fixtures for dynamic machining conditions. Solution for the minimum clamping forces and locator forces is given. Li and Melkote [3] used a nonlinear programming method to solve the layout optimization problem. The method minimizes workpiece location errors due to localized elastic deformation of the workpiece. Roy and Liao [4] developed a heuristic method to plan for the best supporting and clamping positions. Tao et al. [5] presented a geometrical reasoning methodology for determining the optimal clamping points and clamping sequence for arbitrarily shaped workpieces. Liao and Hu [6] presented a system for fixture configuration analysis based on a dynamic model which analyses the fixture–workpiece system subject to time-varying machining loads. The influence of clamping placement is also investigated. Li and Melkote [7] presented a fixture layout and clamping force optimal synthesis approach that accounts for workpiece dynamics during machining. A combined fixture layout and clamping force optimization procedure presented. They used the contact elasticity modeling method that accounts for the influence of workpiece rigid body dynamics during machining. Amaral et al. [8] used ANSYS to verify fixture design integrity. They employed 3-2-1 method. The optimization analysis is performed in ANSYS. Tan et al. [9] described the modeling, analysis and verification of optimal fixturing configurations by the methods of force closure, optimization and finite element modeling.

Most of the above studies use linear or nonlinear programming methods which often do not give global optimum solution. All of the fixture layout optimization procedures start with an initial feasible layout. Solutions from these methods are depend on the initial fixture layout. They do not consider the fixture layout optimization on overall workpiece deformation.

The GAs have been proven to be useful technique in solving optimization problems in engineering [10–12]. Fixture design has a large solution space and requires a search tool to find the best design. Few researchers have used the GAs for fixture design and fixture layout problems. Kumar et al. [13] have applied both GAs and neural networks for designing a fixture. Marcelin [14] has used GAs to the optimization of support positions. Vallapuzha et al. [15] presented GA based optimization method that uses spatial coordinates to represent the locations of fixture elements. Fixture layout optimization procedure was implemented using MATLAB and the genetic algorithm toolbox. HYPERMESH and MSC/NASTRAN were used for FE model. Vallapuzha et al. [16] presented results of an extensive investigation into the relative effectiveness of various optimization methods. They showed that continuous GA yielded the best quality solutions. Li and Shiu [17] determined the optimal fixture configuration design for sheet metal assembly using GA. MSC/NASTRAN has been used for fitness evaulation. Liao [18] presented a method to automatically select the optimal numbers of locators and clamps as well as their optimal positions in sheet metal assembly fixtures. Krishnakumar and Melkote [19] developed a fixture layout optimization technique that uses the GA to find the fixture layout that minimizes the deformation of the machined surface due to clamping and machining forces over the entire tool path. Locator and clamp positions specified by node numbers. A built-in finite element solver was developed.

Some of the studies do not consider the optimization of the layout for entire tool path and chip removal is not taken into account. Some of the studies used node numbers as design parameters.

In this study, a GA tool has been developed to find the optimal locator and clamp positions in 2D workpiece. Distances from the reference edges as design parameters are used rather than FEA node numbers. Fitness values of real encoded GA chromosomes are obtained from the results of FEA. ANSYS has been used for FEA calculations. A chromosome library approach is used in order to decrease the solution time. Developed GA tool is tested on two test problems. Two case studies are given to illustrate the developed approach. Main contributions of this paper can be summarized as follows:

(1) developed a GA code integrated with a commercial finite element solver;
(2) GA uses chromosome library in order to decrease the computation time;
(3) real design parameters are used rather than FEA node numbers;
(4) chip removal is taken into account while tool forces moving on the workpiece.

## 3. Genetic algorithm concepts

Genetic algorithms were first developed by John Holland. Goldberg [10] published a book explaining the theory and application examples of genetic algorithm in details. A genetic algorithm is a random search technique that mimics some mechanisms of natural evolution. The algorithm works on a population of designs. The population evolves from generation to generation, gradually improving its adaptation to the environment through natural selection, fitter individuals have better chances of transmitting their characteristics to later generations.

In the algorithm, the selection of the natural environment is replaced by artificial selection based on a computed fitness for each design. The term fitness is used to designate the chromosome's chances of survival and it is essentially the objective function of the optimization problem. The chromosomes that define characteristics of biological beings are replaced by strings of numerical values representing the design variables.

GA is recognized to be different than traditional gradient-based optimization techniques in the following four major ways [10]:

1. GAs work with a coding of the design variables and parameters in the problem, rather than with the actual parameters themselves.
2. GAs make use of population-type search. Many different design points are evaluated during each iteration instead of sequentially moving from one point to the next.
3. GAs need only a fitness or objective function value. No derivatives or gradients are necessary.
4. GAs use probabilistic transition rules to find new design points for exploration rather than using deterministic rules based on gradient information to find these new points.

Algorithm of the basic GA is given as follows:

1. *Initial population*: Generate random population of chromosomes.
2. *Fitness*: Evaluate the fitness of each chromosome in the population.
3. *Test*: If the end condition is satisfied, stop, and return the best solution in current population.
4. *New population*: Create a new population by repeating following steps until the new population is complete.
   *Reproduction*: Select two parent chromosomes from the population according to their fitness.
   *Crossover*: With a crossover probability, crossover the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
   *Mutation*: With a mutation probability, mutate new offspring at each locus (position in chromosome).
5. *Replace*: Use new generated population for a further run of algorithm.
6. *Loop*: Go to step 2.

### 3.1. Individual representation

The first and most important step in preparing an optimization problem for a GA solution is that of defining a particular coding of the design variables and their arrangement into a string of numerical values to be used as the chromosome by the GA.

In most GAs, finite length binary coded strings of ones and zeros are used to describe the parameters for each solution. In a multiparameter optimization problem, individual parameter coding are usually concatenated into a complete string which is shown in Fig. 1.

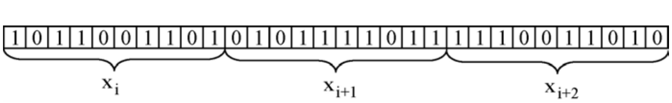


Fig. 1. Binary representation in GA.

In this paper, real representation of binary string is used. The length of the string depends on the required precision. The mapping from a binary string to a real number is completed in two steps:

*Step 1*: Find code length for $x_i$ ($i = 1, \ldots, n$):

$$c = (x_i^{\max} - x_i^{\min}) \times r$$

where $r$ is the required precision ($10^1$, $10^2$, $10^3$, ...).

Code length for $x_i$ is as follows:

$$l_{x_i} = n + 1$$

where,

$$2^n < c < 2^{n+1}$$

Total string length is given by:

$$l = \sum_{i=1}^{n} l_{x_i}$$

*Step 2*: Mapping from a binary string to a real number:

$$x_i = x_i^{\min} + \frac{x_i^{\max} - x_i^{\min}}{2^n - 1} \sum_{j=1}^{n} q_{ij} 2^{j-1}$$

where $q_{ij} \in [0, 1]$.

In order to generate the chromosomes, the length of the chromosome is calculated first. Then random numbers in the range of {0, 1} are generated to form the chromosome. Random function is used in Delphi programming language as a random number generator.

### 3.2. Genetic operators

Establishing the GA parameters is very crucial in an optimization problem because there are no guidelines [20]. The genetic algorithms contains several operators, e.g. reproduction, crossover, mutation, etc.

#### 3.2.1. Reproduction

The reproduction operator allows individual strings to be copied for possible inclusion in the next generation. After assesing the fitness value for each string in the initial population, only a few strings with high fitness value are considered in the reproduction. There are many different types of reproduction operators which are proportional selection, tournament selection, ranking selection, etc. In this study, tournament selection is selected, since it has better convergence and computational time compared to any other reproduction operator [11]. In tournament selection, two individuals are choosen from the population at random. Then the string which has best fitness value is selected. This procedure is continued until the size of the reproduction population is equal to the size of the population.

#### 3.2.2. Crossover

Crossover is the next operation in the genetic algorithm. This operation partially exchanges information between any two

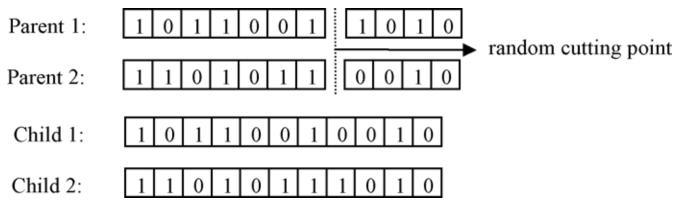


Fig. 2. Illustration of crossover operator.

selected individuals. Crossover selects genes from parent chromosomes and creates new offsprings. Like reproduction operator, there exist a number of crossover operators in GA. In a single-point crossover operator which is used in this paper, both strings are cut at an arbitrary place and the right-side portion of both strings are swapped among themselves to create two new strings, as illustrated in Fig. 2.

In order to carry out the crossover operation, two individuals are selected from the population at random. Then a random number in the range of {0, 1} is generated. If this random number is less than the probability of crossover then these individuals are subjected to crossover, otherwise they are copied to new population as they are. Also the crossover point is selected at random. Probability of crossover ($P_c$) is selected generally between 0.6 and 0.9.

*3.2.3. Mutation*

This is the process of randomly modifying the string with small probability. Mutation operator changes 1–0 and vice versa with a small probability of mutation ($P_m$). The need for mutation is to keep diversity in the population [11]. This is to prevent falling all solutions in population into a local optimum of solved problem. Fig. 3 illustrates the mutation operation at seventh bit position.

In order to determine whether a chromoseme is to be subjected to mutation, a random number in the range of {0, 1} is generated. If this random number is less than the probability of mutation, selected chromosome will be mutated. Probability of mutation should be selected very low as a high mutation will destroy fit chromosomes and degenerate the GA into a random walk. $P_m$ should be selected between 0.02 and 0.06 [21].

*3.2.4. Constraint handling*

In most application of GAs to constrained optimization problems, the penalty function method has been used. In this study a method proposed by Deb [12] is used. Although a penalty term is added to the objective function, this method differs from conventional GA implementations. The method proposes to use a tournament selection operator, where two solutions are compared at a time and the following criteria are always enforced:

- Any feasible solution is preferred to any infeasible solution.
- Among two feasible solutions, the one having better fitness value is preferred.
- Among two infeasible solutions, the one having smaller constraint violation is preferred.

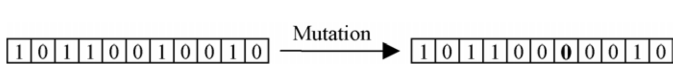


Fig. 3. Illustration of mutation operator.

*3.2.5. Elitist strategy*

In this strategy, some of the best individuals are copied into the next generation without applying any genetic operators. Elitist strategy always clones the best individuals of the current generation into the next generation. This guarantees that the best found design is never lost in future generations.

## 4. Approach

### *4.1. Fixture positioning principles*

In machining process, fixtures are used to keep workpieces in a desirable position for operations. The most important criteria for fixturing are workpiece position accuracy and workpiece deformation. A good fixture design minimizes workpiece geometric and machining accuracy errors. Another fixturing requirement is that the fixture must limit deformation of the workpiece. It is important to consider the cutting forces as well as the clamping forces. Without adequate fixture support, machining operations do not conform to designed tolerances. Finite element analysis is a powerful tool in the resolution of some of these problems [22].

Common locating method for prismatic parts is 3-2-1 method. This method provides the maximum rigidity with the minimum number of fixture elements. A workpiece in 3D may be positively located by means of six points positioned so that they restrict nine degrees of freedom of the workpiece. The other three degrees of freedom are removed by clamp elements. An example layout for 2D workpiece based 3-2-1 locating principle is shown in Fig. 4.

The number of locating faces must not exceed two so as to avoid a redundant location. Based on the 3-2-1 fixturing principle there are two locating planes for accurate location containing two and one locators. Therefore, there are maximum of two side clampings against each locating plane. Clamping forces are always directed towards the locators in order to force the workpiece to contact all locators. The clamping point

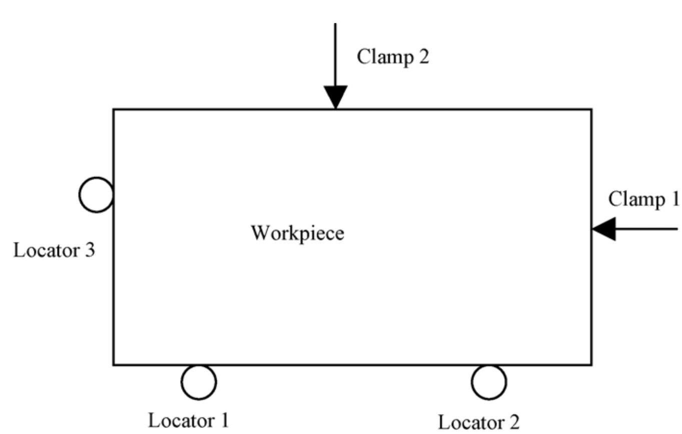


Fig. 4. 3-2-1 locating layout for 2D prismatic workpiece.

should be positioned opposite the positioning points to prevent the workpiece from being distorted by the clamping force.

Since the machining forces travel along the machining area, it is necessary to ensure that the reaction forces at locators are positive for all the time. Any negative reaction force indicates that the workpiece is free from fixture elements. In other words, loss of contact or the separation between the workpiece and fixture element might happen when the reaction force is negative. Positive reaction forces at the locators ensure that the workpiece maintains contact with all the locators from the beginning of the cut to the end. The clamping forces should be just sufficient to constrain and locate the workpiece without causing distortion or damage to the workpiece. Clamping force optimization is not considered in this paper.

### 4.2. Genetic algorithm based fixture layout optimization approach

In real design problems, the number of design parameters can be very large and their influence on the objective function can be very complicated. The objective function must be smooth and a procedure is needed to compute gradients. Genetic algorithms strongly differ in conception from other search methods, including traditional optimization methods and other stochastic methods [23]. By applying GAs to fixture layout optimization, an optimal or group of sub-optimal solutions can be obtained.

In this study, optimum locator and clamp positions are determined using genetic algorithms. They are ideally suited for the fixture layout optimization problem since no direct analytical relationship exist between the machining error and the fixture layout. Since the GA deals with only the design variables and objective function value for a particular fixture layout, no gradient or auxiliary information is needed [19].

The flowchart of the proposed approach is given in Fig. 5.

Fixture layout optimization is implemented using developed software written in Delphi language named GenFix. Displacement values are calculated in ANSYS software [24]. The execution of ANSYS in GenFix is simply done by WinExec function in Delphi. The interaction between GenFix and ANSYS is implemented in four steps:

(1) Locator and clamp positions are extracted from binary string as real parameters.
(2) These parameters and ANSYS input batch file (modeling, solution and postprocessing commands) are sent to ANSYS using WinExec function.
(3) Displacement values are written to a text file after solution.
(4) GenFix reads this file and computes fitness value for current locator and clamp positions.

In order to reduce the computation time, chromosomes and fitness values are stored in a library for further evaluation. GenFix first checks if current chromosome's fitness value has been calculated before. If not, locator positions are sent to ANSYS, otherwise fitness values are taken from the library. During generating of the initial population, every chromosome

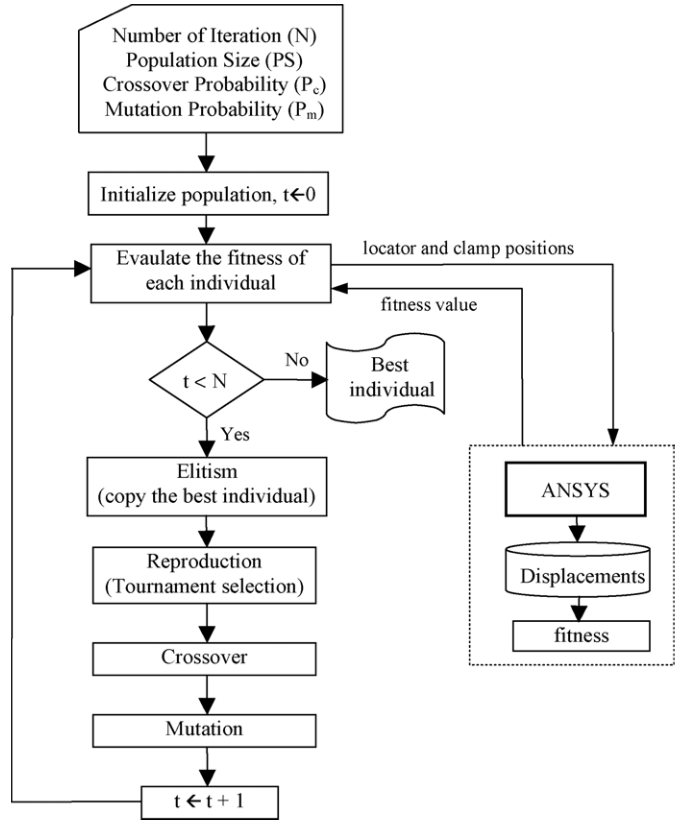


Fig. 5. The flowchart of the proposed methodology and ANSYS interface.

is checked whether it is feasible or not. If the constraint is violated, it is eliminated and new chromosome is created. This process creates entirely feasible initial population. This ensures that workpiece is stable under the action of clamping and cutting forces for every chromosomes in the initial population.

The written GA program was validated using two test cases. The first test case uses Himmelblau function [21]. In the second test case, the GA program was used to optimise the support positions of a beam under uniform loading.

### 4.3. Test case 1: Himmelblau function

The Himmelblau function has several local minimum points and only one global minimum point. The function is:

$$y = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + 0.1[(x_1 - 3)^2 + (x_2 - 2)^2]$$

where $-6 \leq x_1 \leq 6$ and $-6 \leq x_2 \leq 6$.

The function has four minima within the boundary. Only one of them is the global minimum which occurs at (3, 2) with a function value of zero. Other minima occur at (−2.805, 3.131), (−3.779, −3.283) and (−3.584, −1.848) with values of 3.498, 7.386 and 1.515, respectively.

Using a population size of 100, maximum iteration number of 200 and the crossover and mutation probability of 0.85 and 0.02, respectively, results of 10 runs are given in Table 1.

It can be seen that the GA successfully converges to the global minimum.

Table 1
Results of solving the Himmelblau function

| | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 | Run 6 | Run 7 | Run 8 | Run 9 | Run 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 2.9957 | 3.0117 | 2.9829 | 2.9823 | 2.9989 | 3.0088 | 3.0000 | 3.0005 | 3.0084 | 2.9971 |
| $x_2$ | 2.0156 | 1.9687 | 2.0625 | 2.0625 | 2.0039 | 1.9687 | 2.0625 | 1.9980 | 1.9687 | 2.0156 |
| Fitness | 0.0035 | 0.0143 | 0.0581 | 0.0582 | 0.0002 | 0.0138 | 0.0689 | 0.00005 | 0.0138 | 0.0036 |

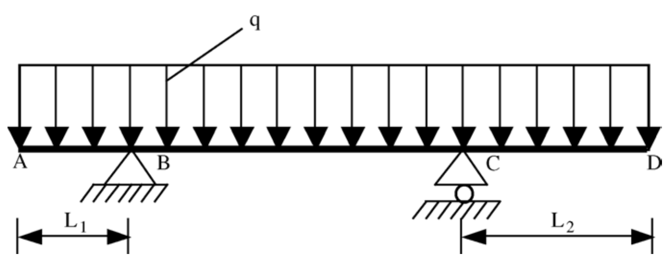


Fig. 6. A continuous beam subjected to a uniform loading.

### 4.4. Test case 2: Beam

A beam is subjected to a uniform distributed load of intensity $q$ is shown in Fig. 6 and its properties are given in Table 2. The goal is to find the optimum $L_1$ and $L_2$ that minimizes the maximum deflection of the beam under a uniform load. Hence, $L_1$ and $L_2$ are selected as design variables. The results obtained in this test case have been compared to the analytical solution.

The optimization problem can be stated as follows:

find $L_1$ and $L_2$,
which minimize the maximum deflection of the beam;
subject to $0 < L_1 < 89$
$91 < L_2 < 180$

Analytical solution for this problem is:

$$L_1 = 40.1668, \qquad L_2 = 40.1668.$$

Maximum deflection is found as 0.2748 mm at points A, D and middle of the beam. GA parameters for this test case are given in Table 3.

Table 2
Beam properties

| | |
|---|---|
| Total length (mm) | 180 |
| $q$ (N/mm) | 100 |
| $E$ (N/mm$^2$) | 210000 |
| Section area (mm$^2$) | 78.5 |
| $I$ (mm$^4$) | 490.625 |

Table 3
GA input parameters for test case

| | |
|---|---|
| Number of iteration | 100 |
| Population size | 40 |
| Crossover probability | 0.85 |
| Mutation probability | 0.02 |
| Total string length | 28 |

Table 4
Results of five runs for beam

| | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 |
|---|---|---|---|---|---|
| $L_1$ (mm) | 39.67 | 41.49 | 39.49 | 40.72 | 41.18 |
| $L_2$ (mm) | 40.53 | 38.12 | 40.52 | 39.32 | 38.62 |
| Fitness (mm) | 0.2878 | 0.3129 | 0.2914 | 0.2901 | 0.3035 |

Results from five runs of the GA tool are given in Table 4. It can be seen from the Table 4 that, best fitness value is found to be 0.2878 and best design variables are $L_1 = 39.67$ and $L_2 = 40.53$ which are very close to the analytical solution. These results showed that GA converged near the optimal solution with a fitness value equal to 0.2878.

## 5. Fixture layout optimization case studies

The fixture layout optimization problem is defined as: finding the positions of the locators and clamps, so that workpiece deformation at specific region is minimized. Note that number of locators and clamps are not design parameter, since they are known and fixed for the 3-2-1 locating scheme. Hence, the design parameters are selected as locator and clamp positions. Friction is not considered in this paper. Two case studies are given to illustrate the proposed approach.

### 5.1. Case study 1

A 2D case study part and optimum values are taken from Krishnakumar and Melkote [19]. 2D fixture layout parameters and tool path are shown in Fig. 7. Cutting forces are applied sequentially to each machining surface node and the maximum deformation for each load application is computed. The maximum deformation for the entire process is then determined

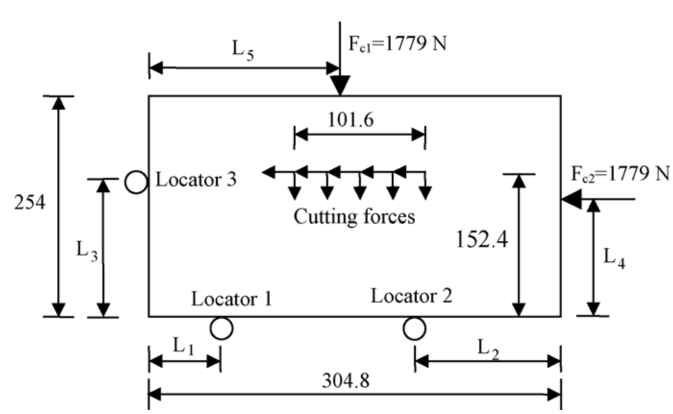


Fig. 7. 2D fixture layout parameters and tool path directions.

Table 5
GA input parameters for case study 1

| | |
|---|---|
| Number of iteration | 150 |
| Population size | 40 |
| Crossover probability | 0.85 |
| Mutation probability | 0.02 |
| Total string length | 73 |

from the maximums for each load application. The GA is used to minimize the maximum deformation for the generation by varying the positions of the locators and clamps. Krishnakumar and Melkote [19] used workpiece node numbers as the locator and clamp positions and a finite element solver was developed and embedded in the GA. Machining force of 889.6 N is assumed for both $F_x$ (←) and $F_y$ (↓).Their work has three drawbacks: (1) they used nodal points as design parameters, so that optimum values can be found only on these nodes; (2) number of generation of GA is very low; (3) machining forces are applied through a line rather than an area.

The optimum layout parameters found by Krishnakumar and Melkote [19] are: $L_1 = 50.4$, $L_2 = 101.6$, $L_3 = 101.6$, $L_4 = 50.8$, $L_5 = 152.4$.

In this paper, same workpiece and fixture layout method (3-2-1) and a new finite element model are used to find the optimal locator and clamp positions. Main differences are the string representation and the finite element solver. Real representation and ANSYS finite element solver are used. Real representation of strings allow more search space to possible solutions. The objective function is defined as the maximum nodal displacement on the part surface being machined.

The GA optimization problem can be stated as follows:

find $L_1, L_2, L_3, L_4, L_5$,
which minimize the maximum nodal displacement on the part surface being machined;

$$\text{subject to}\quad 5 < L_1 < 148$$
$$5 < L_2 < 148$$
$$5 < L_3 < 249$$
$$5 < L_4 < 249$$
$$5 < L_5 < 300.$$

GA input parameters used in this study are given in Table 5.

During generating of the initial population, the feasibility is checked for every chromosome. For given chromosome, if the reaction forces at locators are all positive during machining, it means that locators and workpiece remain in contact. If not, this means that constraint is violated. Then this chromosome is eliminated and new chromosome is created. This process creates entirely feasible initial population.

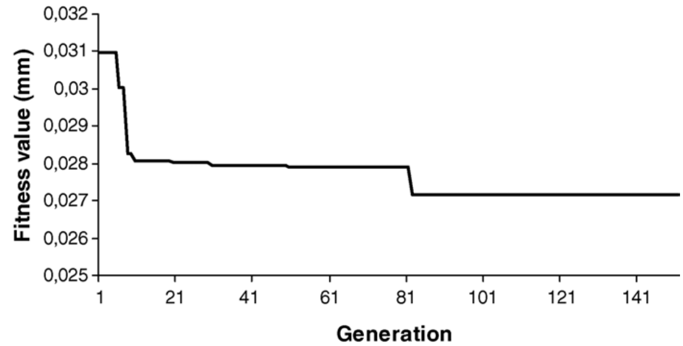


Fig. 8. The convergence of GA for case study 1.

In this study, the value of fitness is costly to compute, because the 40 × 150 = 6000 analysis have to be done. The need for speeding up the computation for GAs is obvious. As the generation goes by, chromosomes in the population are getting similar. Calculated fitness values are stored in a chromosome library, thus, recalculation of fitness values for the same chromosomes is prevented. In this case study, population size is selected as 40 and GA optimization tool runs for 150 generations. This corresponds with 6000 ANSYS FE evaluations. Since the ANSYS has computed only new chromosome values, only 415 new fitness values have been calculated in ANSYS. This results in a tremendous gain in computational efficiency.

Several runs have been performed but only the results of 10 runs are given in Table 6. The first column of the Table 6 shows the optimum layout obtained by Krishnakumar and Melkote [19].

It is seen from the Table 6 that the fitness values are better than the work done by Krishnakumar and Melkote [19]. Although the other fitness values are close to each other, locator and clamp positions are all different. It is observed that this fixturing problem is multi-modal in nature. As the name suggest, multi-modal problems have multiple optimum solutions. The objective in a multi-modal optimization problem is to find multiple optimal solutions having either equal or near equal objective function values. The knowledge of multiple global optimum solutions in the search space is particularly useful in obtaining an insight into the function landscape. In our problem, this will help the designer to have alternative layouts

Table 6
The results of ten runs for case study 1

| | Krishnakumar and Melkote [19] | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 | Run 6 | Run 7 | Run 8 | Run 9 | Run 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $L_1$ (mm) | 50.4 | 95.2 | 124.8 | 126.7 | 110.0 | 111.9 | 70.8 | 107.2 | 129.2 | 138.9 | 131.4 |
| $L_2$ (mm) | 101.6 | 71.1 | 40.3 | 74.3 | 86.8 | 13.8 | 66.6 | 77.8 | 72.2 | 56.3 | 56.4 |
| $L_3$ (mm) | 101.6 | 58.6 | 58.3 | 97.5 | 72.4 | 78.4 | 60.4 | 82.2 | 104.1 | 98.7 | 75.4 |
| $L_4$ (mm) | 50.8 | 23.9 | 6.9 | 69.8 | 58.9 | 12.7 | 8.5 | 66.3 | 5.8 | 6.4 | 16.7 |
| $L_5$ (mm) | 152.4 | 291.3 | 294.1 | 235.2 | 242.3 | 299.7 | 272.2 | 253.7 | 162.6 | 220.1 | 249.2 |
| Fitness (mm) | 0.0393 | 0.0298 | 0.0272 | 0.0333 | 0.0332 | 0.0321 | 0.0338 | 0.0339 | 0.0332 | 0.0293 | 0.0287 |

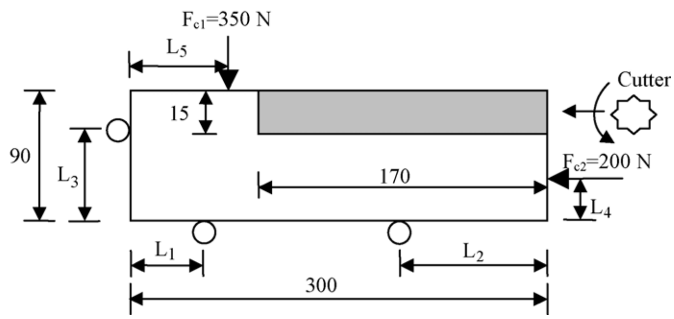


Fig. 9. The end milling process simulated for case study 2.

if needed. This property is not available when using traditional optimization such as linear programming. Adding constraints to the problem may make some optimum solutions infeasible.

In this case study, the global fitness value is found to be 0.0272 refers to Run 2 in Table 6. Variation of fitness value with number of generation is given in Fig. 8. It can be seen that considerable improvement in the objective function is obtained.

### 5.2. Case study 2

This case study considers a step milling operation on a 2D workpiece. The cutting forces for the milling process were calculated as 100 N (←) and 286 N (↓). Clamping forces are assumed as $F_{c1} = 350$ N and $F_{c2} = 200$ N, which are shown in Fig. 9.

The entire tool path is discretized into 13 load steps (see Fig. 10). The model is analysed with respect to tool movement and chip removal using the element death technique [22,24,25]. In order to calculate the fitness value for given chromosome, displacements are stored for each load step. Then maximum displacement is selected as fitness value for this chromosome.

The objective function for this problem is defined as the maximum nodal displacement on the part surface being machined. The GA optimization problem can be stated as follows:

find $L_1, L_2, L_3, L_4, L_5$,

which minimize the maximum nodal displacement on the whole surface;

subject to
$$5 < L_1 < 148$$
$$5 < L_2 < 148$$
$$5 < L_3 < 85$$
$$5 < L_4 < 65$$
$$5 < L_5 < 125.$$

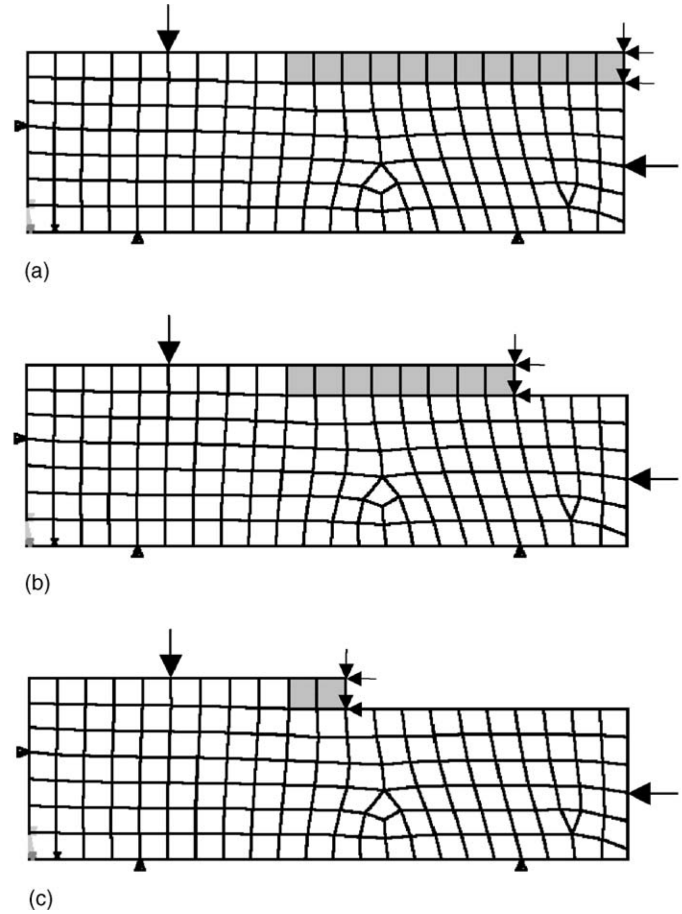


Fig. 10. Finite element models and cutting forces at different load steps. (a) Cutting forces at load step 1; (b) cutting forces at load step 5; (c) cutting forces at load step 11.

Fixture layout optimization approach and input parameters for GA are the same used in case study 1 (see Table 5). Results of 10 runs are given in Table 7.

It can be seen from the Table 7 that the global fitness value is found to be 0.01661 refers to Run 9 in Table 7. Variation of maximum displacement with number of generations is given in Fig. 11.

In this case study, chip removal from the tool path is taken into account. The removal of the material during machining alters the geometry, thus the structural stiffness of the workpiece, which in turn leads to higher deformation. Thus, there is a need to consider chip removal effects for achieving the required machining accuracy.

Table 7
The results of 10 runs for case study 2

| | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 | Run 6 | Run 7 | Run 8 | Run 9 | Run 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $L_1$ (mm) | 98.8 | 86.7 | 101.9 | 103.7 | 113.9 | 59.8 | 110.8 | 125.3 | 62.1 | 81.9 |
| $L_2$ (mm) | 32.7 | 31.8 | 22.6 | 36.3 | 24.8 | 54.7 | 29.4 | 29.5 | 9.11 | 29.3 |
| $L_3$ (mm) | 55.6 | 56.7 | 28.1 | 40.1 | 57.3 | 8.57 | 51.5 | 72.5 | 21.3 | 38.1 |
| $L_4$ (mm) | 34.4 | 29.3 | 5.18 | 50.9 | 5.87 | 29.1 | 18.2 | 20.0 | 8.7 | 8.9 |
| $L_5$ (mm) | 51.0 | 27.2 | 96.23 | 80.0 | 84.2 | 49.1 | 84.6 | 95.0 | 62.2 | 44.6 |
| Fitness (mm) | 0.01816 | 0.01851 | 0.01855 | 0.01831 | 0.01777 | 0.01770 | 0.01757 | 0.01831 | 0.01661 | 0.01846 |

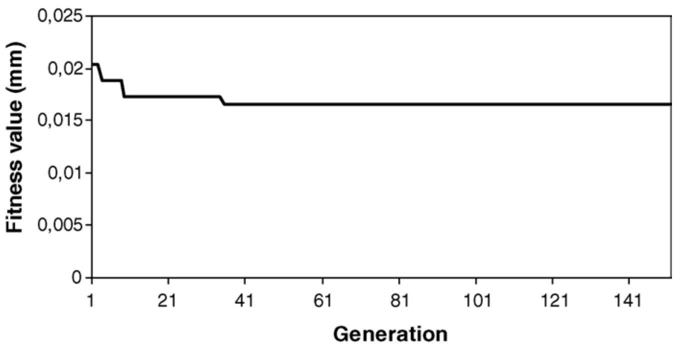


Fig. 11. The convergence of GA for case study 2.

## 6. Conclusion

In this paper, an evolutionary optimization technique of fixture layout optimization is presented. ANSYS has been used for FE calculation of fitness values. It is seen that the combined genetic algorithm and FE method approach seems to be a powerful approach for present type problems. GA approach is particularly suited for problems where there does not exist a well-defined mathematical relationship between the objective function and the design variables. The results prove the success of the application of GAs for the fixture layout optimization problems.

In this study, the major obstacle for GA application in fixture layout optimization is the high computation cost. Re-meshing of the workpiece is required for every chromosome in the population. But, usages of chromosome library, the number of FE evaluations are decreased from 6000 to 415. This results in a tremendous gain in computational efficiency. The other way to decrease the solution time is to use distributed computation in a local area network.

The results of this approach show that the fixture layout optimization problems are multi-modal problems. Optimized designs do not have any apparent similarities although they provide very similar performances. It is shown that fixture layout problems are multi-modal therefore heuristic rules for fixture design should be used in GA to select best design among others.

## References

[1] R.J. Menassa, W.R. DeVries, Optimization methods applied to selecting support positions in fixture design, ASME Journal of Engineering for Industry 113 (1991) 412–418.
[2] R.T. Meyer, F.W. Liou, Fixture analysis under dynamic machining, International Journal of Production Research 35 (5) (1997) 1471–1489.
[3] B. Li, S.N. Melkote, Improved workpiece location accuracy through fixture layout optimization, International Journal of Machine Tools and Manufacture 39 (1999) 871–883.
[4] U. Roy, J. Liao, Geometric reasoning for re-allocation of supporting and clamping positions in the automated fixture design system, IEEE Transactions 31 (1999) 313–322.
[5] Z.J. Tao, A.S. Kumar, A.Y.C. Nee, A computational geometry approach to optimum clamping synthesis of machining fixtures, International Journal of Production Research 37 (15) (1999) 3495–3517.
[6] Y.J. Liao, S.J. Hu, Flexible multibody dynamics based fixture–workpiece analysis model for fixturing stability, International Journal of Machine Tools and Manufacture 40 (2000) 343–362.
[7] B. Li, S.N. Melkote, Optimal fixture design accounting for the effect of workpiece dynamics, International Journal of Advanced Manufacturing Technology 18 (2001) 701–707.
[8] N. Amaral, J.J. Rencis, Y. Rong, Development of a finite element analysis tool for fixture design integrity verification and optimisation, International Journal of Advanced Manufacturing Technology 21 (2004) 411–419.
[9] E.Y.T. Tan, A.S. Kumar, J.Y.H. Fuh, A.Y.C. Nee, Modeling, analysis and verification of optimal fixturing design, IEEE Transactions on Automation Science and Engineering 1 (2) (2004) 121–132.
[10] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[11] K. Deb, An introduction to genetic algorithms, Sadhana Journal 24 (4–5) (1999) 293–315.
[12] K. Deb, An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering 186 (2000) 311–338.
[13] A.S. Kumar, V. Subramaniam, K.C. Seow, Conceptual design of fixtures using genetic algorithms, International Journal of Advanced Manufacturing Technology 15 (1999) 79–84.
[14] J.L. Marcelin, Genetic search applied to selecting support positions in machining of mechanical parts, International Journal of Advanced Manufacturing Technology 17 (2001) 344–347.
[15] S. Vallapuzha, E.C. De Meter, S. Choudhuri, R.P. Khetan, 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 42 (2002) 265–275.
[16] S. Vallapuzha, E.C. De Meter, S. Choudhuri, R.P. Khetan, An investigation of the effectiveness of fixture layout optimization methods, International Journal of Machine Tools and Manufacture 42 (2002) 251–263.
[17] B. Li, B.W. Shiu, Principle and simulation of fixture configuration design for sheet metal assembly with laser welding. Part 2. Optimal configuration design with the genetic algorithm, International Journal of Advanced Manufacturing Technology 18 (2001) 276–284.
[18] Y.G. Liao, A genetic algorithm-based fixture locating positions and clamping schemes optimization, Proceedings of the Institute of Mechanical Engineers. Part B, Journal of Engineering Manufacture 217 (2003) 1075–1083.
[19] K. Krishnakumar, S.N. Melkote, Machining fixture layout optimization, International Journal of Machine Tools and Manufacture 40 (2000) 579–598.
[20] I. Rajendran, S. Vijayarangan, Optimal design of composite leaf spring using genetic algorithms, Computers and Structures 79 (2001) 1121–1129.
[21] K. Deb, Genetic algorithm in search and optimization: the technique and applications, in: Proceedings of International Workshop on Soft Computing and Intelligent Systems, Calcutta, India, 1998, pp. 58–87.
[22] N. Kaya, F. Öztürk, Algorithms for grouping machining operations and planning workpiece location under dynamic machining conditions, International Journal of Production Research 39 (15) (2001) 3329–3351.
[23] G. Renner, A. Ekart, Genetic algorithms in computer aided design, Computer Aided Design 35 (2003) 709–726.
[24] ANSYS 8.0, ANSYS Inc., 2003.
[25] N. Kaya, F. Öztürk, The application of chip removal and frictional contact analysis for workpiece–fixture layout verification, International Journal of Advanced Manufacturing Technology 21 (2003) 411–419.

**Necmettin Kaya** received his BSc degree in mechanical engineering at the Uludag University in 1990. He obtained his MSc degree in 1992 from the Uludag University also in mechanical engineering. He joined doctoral program in the Mechanical Engineering Department in 1993. He completed his PhD degree in 1999, while conducting research in the area of feature based modular fixture design. Since 2002, he is assistant professor at the Mechanical Engineering Department of the Uludag University. His principal research areas are computer aided design, finite element method, fixture design, genetic algorithms and shape optimization.