Declining Neighborhood Simulated Annealing Algorithm to Robust Fixture Synthesis in a Point-Set Domain

Taqiaden Alshameri

Hefei University of Technology <https://orcid.org/0000-0002-9958-8064>

Yude Dong (✉ yddong@hfut.edu.cn)

Hefei University of Technology

Abdullah Alqadhi

Hefei University of Technology

Research Article

Keywords: Fixture synthesis, Simulated annealing, Robust, Shannon entropy, Computer-aided design (CAD)

Posted Date: July 14th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-705119/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on January 28th, 2022. See the published version at <https://doi.org/10.1007/s00170-021-08507-z>.

Declining Neighborhood Simulated Annealing Algorithm to Robust Fixture Synthesis in a Point-Set Domain

Taqiaden Alshameri · Yude Dong · Abdullah ALqadhi

Received: date / Accepted: date

Abstract Fixture synthesis addresses the problem of fixture-elements placement on the workpiece surfaces. This article presents a novel variant of the Simulated Annealing (SA) algorithm called Declining Neighborhood Simulated Annealing (DNSA) specifically developed for the problem of fixture synthesis. The objective is to minimize measurement errors in the machined features induced by the misalignment at workpiece-locator contact points. The algorithm systematically evaluates different fixture layouts to reach a sufficient approximation of the global optimum robust layout. For each iteration, a set of previously accepted candidates are exploited to predict the next move. Throughout the progress of the algorithm, the search space is reduced and the new candidates are designated according to a declining Probability Density Function (PDF). To assure best performance, the DNSA parameters are configured using the Technique for Order Preference by Similarity to Ideal Solution (TOPOSIS). Moreover, the parameters are set to auto-adapt the complexity of a given input based on a Shanon entropy index. The optimization process is carried out automatically in the Computer-Aided Design (CAD) environment NX; a computer code was developed for this purpose using the Application Programming Interface (API) NX-Open. Benchmark examples from industrial partner and literature demonstrate satisfactory results.

Keywords Fixture synthesis · Simulated annealing · Robust · Shanon entropy · Computer-aided design (CAD)

1 Introduction

Fixture refers to the device used to immobilize and localize the workpiece during machining. Elements of fixtures include many components such as locator pins, locator pads, supports, and clamps. Locators are critical components that used to maintain the proper position of the workpiece. Errors in Locators-workpiece contact points (known as localization source error) can add to the formation of workpiece positional errors. Positional errors are measurement errors in the machined features (features of strict tolerances requirements are of main concern). To minimize these errors, Locators should be robustly positioned on the workpiece surfaces. Generally speaking, the exterior surfaces of the workpiece is assumed to be a vast collection of discrete points. These points represent all available positions for locator placement. In fixture synthesis, we deal with the allocation of a given number of locators on these available coordinates.

The problem of fixture synthesis is of combinatorial complexity [40]. Thus, using a simple combinatorial iterative test for all possible candidates is impractical even for a small number of locators. Fixture synthesizing problems have been conquered in the literature using traditional methods such as linear and nonlinear programming [40, 5, 43]. However, these methods do not guarantee a global or near-global solution. Additionally, it is highly dependent on the selection of the initial solution. On the other hand, nontraditional methods such as evolutionary algorithms are ca-

Taqiaden Alshameri^{1,2}: taqiaden@mail.hfut.edu.cn

Yude Dong^{1,*}: yddong@hfut.edu.cn

Abdullah ALqadhi³: Abu@mail.hfut.edu.cn

*Correspondance

1.School of Mechanical Engineering, Hefei University of Technology, Hefei 230009, China

2.School of Mechanical Engineering, Sana'a University, Sana'a 31220, Yemen

3.School of Management Science, Hefei University of Technology, Hefei 230009, China

Fig. 1 Study framework diagram

pable to surpass these drawbacks. Studies found in the literature regarding the fixture synthesis problem have been specifically focused on the use of Genetic Algorithm (GA) and Ant Colony Algorithm (ACA) [37, 24, 23, 21, 36, 20, 38, 18].

This paper proposes a new approach for robust fixture synthesis in a point-set domain implemented in the Computer-Aided Design (CAD) environment [39]. The optimization is carried out using a new algorithm called Declining Neighborhood Simulated Annealing (DNSA). The name Declining Neighborhood reflects the generating mechanism of the next candidates which will be discussed in detail. The framework of this study is illustrated in Fig. 1. A fixture model is drawn on kinematic analysis and mechanics of rigid bodies as in previous studies [7, 2, 29] but with a different objective function (cost function). The inputs to the DNSA are point-set domain and weighted reference points. Point-set domain is simply discrete points (extracted in the CAD software NX) of all possible coordinates on workpiece surfaces that are feasible for locator placement. Weighted Reference points are essential for the evaluation process which will be discussed in sections 3 and 4. The algorithm output is expected to be equal or sufficiently close to the global optimum.

Section 2 discusses some related works. The fixture model and the objective function are presented in section 3. Section 4 explains the discretization approach. Section 5 outlines the proposed DNSA algorithm. Setting the algorithm parameters is explained in section 6. The result and discussion of benchmark problems are presented in Section 7. Finally, section 8 concludes this paper and suggests possible future research direction.

2 Related works

The automatic configuration of machining fixtures includes two kinds of problems: fixture analysis and fixture synthesis [10]. Fixture analysis emphasizes the performance evaluation of a certain fixture schema under a set of fixturing requirements such as deterministic locating and total fixturing [8]. Fixture analysis theories are built on the classical screw theory [10], or geometric perturbation techniques [6]. On the other hand, fixture synthesis is concerned with the design of fixture layout to fulfil given requirements. Randy and Kenneth [5] Presented one of the early works in synthesizing the arrangement of three locators and one clamp for polygonal parts. Phoomboplab and Ceglarek [27] propose a design synthesis framework based on a new hybrid Design Structure Matrix and Task Flow Chain for dimensional management in multistage assembly. The design tasks are Tolerance Optimization, Multi-Fixture Layout Design, and Part-to-part Joint Design. Yu Zheng and Wen-Han Qian [44] investigate the use of 3-D modular fixtures for complex objects. They developed an algorithm to determine the optimal fixture element locations based on localization accuracy and immobilization capability criteria. On the whole, a review paper reported that the number of studies in the scope of fixture synthesis is fewer compared to fixture analysis [38].

In the context of fixture model, three workpiece models are commonly used for fixture layout optimization: Rigid body model, fixture contact point elastic model, and workpiece elastic model. Fuyong Yang et. al. [42] have developed a rigid body model to compensate for datum errors during machining with general fixture layouts. Yun-Hui Liu et. al [12] proposed an algorithm based on a ray-shooting technique to search for optimum form-closure grasps of hard fingers on a surface of a 3-D object. Satyanarayana and Melkote [33] have employed workpiece elastic deformation and reaction forces to predict machining surface error and stability of fixture-workpiece system. Anand Raghu and Melkote [30] demonstrated the effect of clamping sequence on workpiece location error using a workpiece-fixture elastic contact point model. James [3] combined contact elasticity with finite element methods for pressure distribution and contact load prediction at the contact points in a workpiece-fixture system. The aim is to determine the required clamping forces to keep the workpiece in position during machining. Krishnakumar and Melkote [20] and Kaya [18] optimized the fixture layout using simple GA and a workpiece elastic model. The objective is to minimize the workpiece elastic deformation under static loading conditions.

Traditional methods for fixture layout optimization have been reported by many studies. Wang and Pelinescu [40] present an iterative optimization method based on sequential optimal deletion in the point-set domain for fixture synthesis. The objective is to meet two performance criteria: the norm and dispersion of the locator contact forces and the workpiece localization accuracy. Brost and Goldberg [5] Synthesize the Modular Fixtures using an iterative algorithm. Their algorithm grows with an order of $O(n^5)$ where n represents the number of edges multiplied by the maximum diameter. Yu Zheng and Chee-Meng Chew [43] proposed a graphical approach based on the simplex distance to hold an object in a form closure while enhancing immobilization capability. The aforementioned fixture design methods have many drawbacks. for instance, they may lead to a local minimum solution. Moreover, some of these methods apply an exhaustive iteration to find the best solution which is not suitable for a vast range of candidates.

The heuristic approach found many applications in fixture synthesizing and analyzing problems. Vallapuzha [37] compared the performance of four fixture layout optimization algorithms, discrete GA, continuous GA, discrete SQP, and continuous SQP. Prabhakaran et al. [28] applied ACA and GA algorithms separately to the fixture optimization problem. The model used finite element analysis to minimize the dimensional and form errors. The results report faster and more accurate solutions for ACA. Padmanaban et al. [24, 23] used ACA to optimize the fixture layout design, the objective is to reduce elastic deformation. The algorithm was tested in both continuous and discrete domains with results prove the superiority of the former.

As far we know this work is the first to implement a new variant of Simulated Annealing (SA) for the discussed problem. SA holds many advantages which makes it suitable for fixture synthesis. For instance, the initial set of candidates has no significant impact on the final solution. Additionally, it has been theoretically proven that the SA is approaching a probability of one to reach the global optimum as long as the annealing schedule is extended [15]. The proposed algorithm is incorporated as a subroutine in the CAD software (NX) which contributes to the CAD-based methods for automated fixture design.

3 Fixture model

Before the construction of the optimization method, the objective function is formulated based on a rigid body fixture model. The fixture model is used to map the relationship between the localization source error

and the workpiece positional error. Localization error is assumed to result from three sources of variation: variation due to locator setup errors, variation due to manufacturing default of the locators, and variation due to manufacturing default of the workpiece [29]. In this work, locators are assumed to be sufficiently represented as points on a surface. For simplicity, the minimum number of locators is used to constrain a 2-D workpiece in a deterministic locating schema i.e., three locators are used [2].

The source variation vector for 3 locators is denoted as $\delta \mathbf{R} = [\delta x_1 \ \delta y_1 \ \delta x_2 \ \delta y_2 \ \delta x_3 \ \delta y_3]^T$ where δx_i and δy_i represents the source variation of a locating point at the coordinate (x_i, y_i) . The positional error ($\delta \mathbf{q}_{ref} = [\delta x_{ref} \ \delta y_{ref} \ \delta \theta_{ref}]^T$) at any reference point on the workpiece includes two parts, transitional δx_{ref} , δy_{ref} , and rotational $\delta \theta_{ref}$ with ref denotes the reference point. The value of $\delta \mathbf{q}$ error follows directly the variation in $\delta \mathbf{R}$. A transformation matrix $\mathbf{B}$ is used to correlate between the two variables such that $\delta \mathbf{q} = \mathbf{B}(\delta \mathbf{R})$. Call from the kinematic analysis of previous studies [29, 8] the matrix $\mathbf{B}$ is found to be dependent on the unit normal vectors and the coordinates at locator-workpiece contact points:

$$\mathbf{B} = -\mathbf{J}^{-1} \Phi_{\mathbf{R}} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,6} \\ \vdots & \ddots & \vdots \\ a_{3,1} & \cdots & a_{3,6} \end{bmatrix} \quad (1)$$

Where $\Phi_{\mathbf{R}}$ and $\mathbf{J}$,

$$\Phi_{\mathbf{R}} = \begin{bmatrix} \mathbf{n}_1^T & 0 & 0 \\ 0 & \mathbf{n}_2^T & 0 \\ 0 & 0 & \mathbf{n}_3^T \end{bmatrix} \quad (2)$$

$$\mathbf{J}_i = [-\mathbf{n}_i^T \ n_{iy}x_i - n_{ix}y_i] \quad (3)$$

$$\mathbf{n}_i = [n_{ix} \ n_{iy}]^T \quad (4)$$

The notations n_{ix} , n_{iy} , $i = 1, 2, 3$ are the unit normal vector components at the locating point (x_i, y_i) . For a 2-D workpiece if the Jacobian matrix ($\mathbf{J}$) rank is less than 3 then the matrix is uninvertible which means the workpiece is not deterministically located [2].

To assure a robust fixture configuration, locators are arranged properly to minimize the variation in positional errors induced by the localization source errors. In this study, a typical localization source error is assumed to follow an independent normal distribution, therefore, the variation of positional error is:

$$var(\delta \mathbf{q}_{ref}) = \sigma^2 \sum_{j=1}^{j=6} [a_{1,j}^2 \ a_{2,j}^2 \ a_{3,j}^2]^T \quad (5)$$

Eq. 5 is used to quantify the variance for both transitional and rotational error at a specified reference point.

The focus in this context is on the transitional error and not the rotational error. Hence, the matrix element $a_{3,j}^2$, which reflects the rotational error, is dropped. In practice, we can either count for resultant transitional error in the whole part or exclusively draw focus to critical feature(s) in the workpiece. Thus, instead of computing for one reference point, a set of reference points that fully depict these features is computed for transitional error, and the objective function is set to be the summation of matrix elements ($a_{i,j}$) at all reference points:

$$F(\mathbf{X}) = \sum_{ref=1}^{ref=N_{ref}} \left[\sum_{i=1,j=1}^{i=2,j=6} a_{i,j}^2 \right]_{ref} \quad (6)$$

Reference points, also known as key product characteristics points (KPCs) in other related studies, are automatically generated and individually assigned a specific weight w_{ref} . The generating mechanism and the calculation of these weights will be described in section 4. In brief, weights are indicators of how close a reference point to the feature of interest. The new objective function is then:

$$F(\mathbf{X}) = \sum_{ref=1}^{ref=N_{ref}} w_{ref} \times \left[\sum_{i=1,j=1}^{i=2,j=6} a_{i,j}^2 \right]_{ref} \quad (7)$$

The kinematic analysis used to derive the objective function 7 takes the mobility of bodies in contact as a function of surface normal, which implies a disregard for the local motion at the contact points. This treatment is sufficient for workpieces of plane surfaces. However, considering that positioning a locator at a curved region of the workpiece introduces more error. Hence, employing the 2nd-order mobility index (curvature) becomes essential [31]. Hereafter, we adopt using a correction value for curvature as proposed by [8]. The correction value is defined by Eq. 8 where $\kappa_i, i = 1, 2, 3$ refers to the Gaussian curvature at locating points and R is the mean dimension of the workpiece. These corrections are applied to the objective function, specifically, to the source variation vector $\delta \mathbf{R} = [C_1 \delta x_1 \ C_1 \delta y_1 \ C_2 \delta x_2 \ C_2 \delta y_2 \ C_3 \delta x_3 \ C_3 \delta y_3]^T$. A correction C_i at a locating point can be interpreted as an error amplification coefficient; when a locator is placed at a linear surface the curvature is zero and $C_i = 1$, while for a curved surface the correction value is more than one which results in a higher positional error. The new objective function with curvature correction is given by Eq. 9.

$$C_i = 1 + \kappa_i R \quad (8)$$

Fig. 2 Uniform mesh discretization of CAD face entity

$$F(\mathbf{X}) = \sum_{ref=1}^{ref=N_{ref}} w_{ref} \times \left[\sum_{i=1,j=1}^{i=2,j=3} C_j^2 (a_{i,(2j-1)}^2 + a_{i,(2j)}^2) \right]_{ref} \quad (9)$$

4 Discretization

The fixture synthesis problem deals with the placement of locators at the workpiece surfaces. For 2-D workpiece geometry, the positioning is at the boundary edges of the face entity of the workpiece CAD model. The boundary edges are converted to a set of discrete points which are indexed and maintained in a string to be referred as the point-set domain. Discretization is achieved by firstly converting the face entity into a set of uniformly spaced points and then isolating the boundary path as shown in Fig. 2. The uniform mesh parameter is the constant α which denotes the step incrementation. This method of uniform mesh discretization is simple to perform and it allows workpiece of arbitrary shape to be discretized with minimal complexity [26]. The discretization method is built with the following assumptions: 1) Workpiece boundary is smooth; 2) Distance between nodes (α) is so small so it does not significantly affect the accuracy of the solution.

The calculation of mathematical attributes (first and second derivatives, unit norm vectors, and curvature) for all points in the set domain can be explained by taking one instance point. Fig. 3 shows an instance point in the point-set domain with inline adjacent points. two hypothetical adjacent points are the base for approximating attributes. These two points are located on the two forward and backward fitting lines which are determined by interpolating adjacent points in both sides of the instance point (four points on both sides in this study). The distance between each hypothetical point and the instance point, denoted by β ,

Fig. 3 Instance point with adjacent points in the point-set domain of the boundary

Fig. 4 Apply the smoothing by bin for curvature values of an eclipse shape

is the mean distance between all pairs of two adjacent points in the boundary. The first and second derivatives at an instance point are approximated using the symmetric form of Newton's difference quotient. The curvature κ is computed according to the implicit formula of Gaussian curvature [14]. And the unit normal vector is simply determined as shown in Fig. 3.

The obtained attributes from the previous computation contain much noise. Counting on the assumptions of smooth boundary, we further apply a smoothing by bin for all attribute sets. In smoothing by bin, each attribute value is substituted by the mean value of a bin. A bin composed of the attribute values of the instance point, the previous 3 points, and the next 3 points. Fig. 4 shows curvature values for an eclipse shape before and after the smoothing by bin.

Generating of reference points: Reference points are representative points for a given feature and are used to estimate the resultant transitional error in the neighbor region. For a 2-D workpiece, a feature is recognized and bounded by a rectangular as shown in Fig. 5. The se-

Fig. 5 Automatic generating mechanism of reference points

lection of reference points for any feature is performed in two rounds, in each round, reference points on the feature boundary are chosen according to the following rules:

Round-one: four points located at the feature boundary having the minimum distance from the four corners of the bounding rectangular are selected.

Round-two: if any edge of the bounding rectangular does not contain at least one point that is selected from the previous round, then pick one point at the intersection of the edge and the feature boundary.

If round one yields at least one point located at each edge of the boundary rectangular then the new round produces no extra points, as in rectangular shape. However, if no point of the first round is placed at any edge of the bounding rectangular, then the new round will yield four points, as in eclipse shape. Other cases lie between these two cases.

Reference points weighting mechanism: The weight of any reference point is an indicator of its significance to function as a delegate for the feature(s). To estimate the values of these weights, the feature boundary edges are discretized in the same manner shown before (Fig. 2). Then, the contribution of each non-reference point is measured based on the closeness of this point to the reference point. weight of a reference point is the summation of all points contributions given by Eq. 10, where d_{ij} denotes the distance between a reference point $i = 1, 2, \dots, n$, and a point $j = 1, 2, \dots, m$ in the boundary edge of the target feature.

$$w_{ref} = \sum_{j=1}^m \frac{1}{|d_{ij}| \times \sum_{i=1}^n \frac{1}{|d_{ij}|}} \quad (10)$$

In practice, the workpiece can have many features of strict accuracy requirements. These features must be the base for computing the cost function. Accordingly, reference points are selected for all features separately. Then, for convenience, all weight values are normalized by dividing each weight by the sum of all weights.

5 Optimization algorithm

Simulated annealing is a probabilistic technique that was firstly introduced by Kirkpatrick et al. [19] and Černý [9] separately. It belongs to the class of Monte Carlo methods [22]. SA is built on the simple greedy search algorithm known as hill-climbing. In hill-climbing, the algorithm starts with an arbitrary value, then it moves toward an increased value to search for a better solution. The algorithm terminates when no further improvement is found. The main drawback of hill climbing is that it may get stuck in a local optimum. On the other hand, SA employs a probabilistic technique to overcome this drawback.

Comparison of both algorithms can be demonstrated by the simple example of a rabbit hopping around in an attempt to find the lowest point in a valley:

Hill-climbing algorithm: The rabbit successively jumps to a place lower than its current level until it reaches the lowest point not far away. This point is not necessarily the lowest in the valley.

Simulated annealing algorithm: The rabbit is drunk; It starts jumping randomly and then gradually becomes sober and jumps towards a lower direction until it reaches the lowest point or near the lowest point in the valley.

The concept of SA is drawn from a well-known physical phenomenon called annealing. In thermodynamics, annealing refers to the process of slow cooling where an object gradually cools down from an initial state of high temperature aiming to increase the size of crystals and reduce defects. During the slow cooling, the microstructure of the material is gradually altered so that molecules can have enough time to settle. This process eventually reaches a crystallization form which corresponds to the lowest energy state of the object. Reversely, if the process cools too fast (also known as "quenching"), this results in an amorphous form of the material which is not the lowest energy state. In short, the lower the cooling rate, the lower the final energy state of the object.

SA algorithm mainly applies the theory of thermodynamics annealing to statistics. Every point in the search space can be visualized as a molecule in the air with a specific free energy value. Search space points, like the air molecule, also have "energy" to indicate the suitability of the point to the problem. The algorithm starts searching for an arbitrary point in the space. In each step, a "neighbor" is selected first, and then the probability of reaching the "neighbor" from the existing location is calculated. If the probability is greater

than a given threshold, jump to the "neighbor"; if not, stay in the original position.

In minimization problems like in fixture synthesis, if a new solution is accepted, it is called a downhill move. If the new solution does not satisfy the acceptance condition, then it is subject to a probability function to decide whether to accept the new candidate. Following this strategy, SA shows efficiency in escaping a local minimum. The probability of accepting a new candidate is given by Eq. 11. ΔE denotes the change in the objective value. In the hill-climbing algorithm the probability of accepting a new candidate when $\Delta E > 0$ is zero, however in SA it is equal to the metropolis criterion $\exp(-\Delta E/k_b T)$ [22]. Where T is the current temperature and k_b is the Boltzmann constant which is calculated in every iteration by setting the metropolis criterion equals to one at $T = T_{max}$ and ΔE equals the minimum non-zero value obtained so far. Through a detailed examination of the acceptance probability equation, we can notice two conditions. First, as the solid cools down, the probability of accepting a worse candidate gets smaller. Second, if the change in energy (cost value) ΔE is high then the probability of acceptance is accordingly low.

$$p_a = \begin{cases} 1 & \text{if } \Delta E < 0 \\ \exp(-\frac{\Delta E}{k_b T}) & \text{otherwise} \end{cases} \quad (11)$$

So far, we have discussed the general principles of SA. In this study, we propose the DNSA algorithm which is a new variant of the original algorithm, referred as Standard Simulated Annealing (SSA). The new algorithm is developed specifically for the current problem of fixture synthesis. The proposed algorithm differs from the SSA in two aspects: First, DNSA exploits a set of previously accepted candidates to predict the next move. Second, the generation mechanism of new candidates is governed by a declining probability density function (PDF). To put it differently, a group that keeps records of the last accepted candidates is employed to direct the search area, more elaboration is in section 5.3.

5.1 The proposed algorithm

DNSA algorithm mainly contains two loops; The outer loop in which both temperature and the PDF domain size parameters are updated for each iteration. The temperature starts from an initial maximum temperature T_{max} and decreases gradually with the progress of the algorithm according to a cooling schedule. The PDF domain size parameters control the range of possible inputs. The range begins with the full string size of the point-set domain and decreases gradually. The inner

loop is the epoch loop where several iterations are repeated at the same temperature and PDF domain size. Evaluating and selection of candidates are carried out in this loop as follows: First, a new candidate is generated (see section 5.3). Second, the objective value of the selected candidate is computed using Eq. 7 when curvature effect is disregarded, or Eq. 9 when considering a correction for curvature. Then, the candidate is appended to the group of last accepted candidates based on the acceptance criteria of Eq. 11. The group capacity is set to a specified maximum number of candidates (have been set equal to 10, see section 6.1). When the group reaches its maximum limit, the oldest candidate is dropped to make space for the new one. If the group members are altered, then the PDF mean values are recomputed based on the new values. Finally, termination criteria are checked at the end of each iteration (see section 5.4).

5.2 Cooling schedule

The cooling schedule plays a critical role in DNSA. fast cooling rate may result in a local optimum. Conversely, a slow cooling rate causes the optimization process to last longer. The cooling rate depends on a set of parameters, specifically, Initial temperature, final temperature, and the number of iterations denoted as T_{max} , T_{min} and d respectively. Many cooling rate formulas are presented in the literature. In this paper, three cooling rate formulas were tested in contiguity with other parameters (see section 6.1). The selection is made to the exponential multiplicative cooling rate formula given by Eq. 12.

$$T_i = T_{max} e^{-k_c d}, \text{ and } k_c = \frac{\ln\left(\frac{T_{max}}{T_{min}}\right)}{d_{final}} \quad (12)$$

The cooling rate coefficient denoted by k_c is obtained by setting $T_i = T_{min}$ at $d = d_{final}$ where d_{final} is the last iteration. The used formula permits longer execution time at the lower temperature stage (final stage) compared to the higher temperature stage (early stage). This assures intensive search at the end of execution to further refine the solution.

5.3 Generation mechanism

The generating of new candidates is mainly controlled by a PDF function most of the execution time. A separate PDF is assigned for each locator. To simplify, the PDF is applied over the point-set domain to assign a selection probability for each index in the string. A set

Fig. 6 PDF declining mechanism in harmony with the drop in temperature

of control parameters are used to control the PDF behavior. Among them there is the mean parameter. The mean is used to direct the search area and its value is estimated based on the last accepted candidates. The higher the number of candidates used to compute the mean, the more accurate is the prediction of the searching zone. At the early stage of the algorithm execution, the group holds a minimum number of candidates. therefore, the prediction of the searching zone is not accurate due to high uncertainty in the estimated mean value. To solve this, a preliminary phase (first phase) precedes the main phase of PDF generating mechanism (second phase) as in Fig. 6. The first phase generates new candidates according to uniformly distributed probability. This phase continues until the number of lastly stored accepted candidates (group size) reaches the maximum limit. At an early stage of the second phase, the selection space covers a wider range in the point-set domain, and the searching mechanism is characterized as diversification. On the other hand, last stage iterations are characterized as intensification and the focus is drawn to a narrower zone. During this narrowing transition of the searching space, the solution gets closer to the global optimum. The PDF is designed to harmonize this transition between the two stages as shown in Fig. 6. In this context, we have proposed a set of piecewise Eq. 13 to mathematically depict the declining transition:

$$P(x_i) = \begin{cases} z, & 0 \leq x_i \leq c \\ ze^{l(c-x_i)}, & c < x_i < a \end{cases} \quad (13)$$

Algorithm 1 DNSA**Input:** Point-set domain and weighted reference points.**Output:** Optimum fixture layout.

```

1: for each iteration do
2:   Update temperature; ▷ Eq. 12
3:   Update PDF domain size parameters;
4:   for each epoch do
5:     if group of last accepted candidates is not full then
6:       Generate new candidate randomly;
7:     else
8:       Generate new candidate according to the PDF;
9:     Compute new cost; ▷ Eqs. 7 or 9
10:    Update acceptance probability; ▷ Eq. 11
11:    if acceptance probability  $\zeta$  randomize (0,1) then
12:      if current candidate is better than the optimum then
13:        Update the optimum candidate;
14:        Add current candidate to the group of last accepted candidates;
15:        if group size of last accepted candidates exceeds an upper limit then
16:          Erase the oldest element in the group;
17:        Update PDF mean parameters;
18:        Check termination criteria;

```

This set of equation are for one half of the PDF profile and the other half is simply the inverse as shown in Fig. 7. The x_i value here is the index of the point to be selected as the new candidate. The parameters z , c , a , and l are characteristic constants that control the shape and size of the PDF. There exist an infinite number of alternatives to adjust these parameters. In this study, we intuitively set the parameter values without extra complexity as given in Table 1. To generate a random point according to the probability defined by the given PDF, we first integrate the previous piecewise equation to get the cumulative probability distribution:

$$y = \begin{cases} zx_i, & 0 \leq x_i \leq c \\ zc + \frac{z}{l}(1 - e^{l(c-x_i)}), & c < x_i < a \end{cases} \quad (14)$$

or, in reverse form,

$$x_i = \begin{cases} \frac{y}{c}, & 0 \leq y \leq zc \\ c - \frac{1}{l} \ln(1 - \frac{y-zc}{z}), & zc < y < zc + \frac{z}{l}(1 - e^{l(c-a)}) \end{cases} \quad (15)$$

To obtain the value of x_i . First, the variable y is generated randomly following a uniform distribution within the range $0 < y < zc + \frac{z}{l}(1 - e^{l(c-a)})$. Then, substituting y in Eq. 15 yields the random x_i that follows the specified PDF.

In general, the PDF function is controlled using only the parameters m , a , and c (m refers to the mean value, whereas a and c are the domain size control parameters) while parameters z and l are maintained constant. Mean associated with the PDF of the $loc.^{th}$ locator is estimated using the Inverse Distance Weighted (IDW) interpolation (Eq. 16) applied on the group of last accepted candidates [35]. Weights w_{ji} are the reciprocal

Fig. 7 Proposed PDF function plot

of objective values for the group candidates and x_i is the corresponding index of each candidate as mentioned earlier.

$$\hat{m}_{loc.} = \frac{\sum_{j=1}^m x_i w_{ji}}{\sum_{j=1}^m w_{ji}} \quad (16)$$

Fig. 8 provides more elaboration for the PDF generating mechanism phase. Vertical arrows of different lengths symbolize the objective value of each candidate in the group. The points in the domain bounded by the dashed rectangular area are of potentiality to be selected in the next round. Indexes close to the middle of the dashed rectangular are more likely to be selected and this likelihood decreases as we move away toward the two extreme ends.

5.4 terminating criteria

The algorithm has two terminating conditions. The first condition is met when the number of iterations reaches

Table 1 Values of PDF paramters

Parameter	Value	Notes
a_{max}	$(pointset_domain_size - 1)/2$	The value of a at the initiation of the algorithm. This will cover the entire string of the point-set domain.
a_{min}	6	The PDF is applied over a range of six indexed values of the point-set domain at the last iteration.
$c_{reduction_rate}$	k_c^2	The reduction rate of the upper portion of the PDF (see Fig. 7) is square the cooling rate.
$a_{reduction_rate}$	k_c	The reduction rate of the lower portion of the PDF (see Fig. 7) is identical to the cooling rate.
z	100	Roughly estimated.
l	$\log(\frac{1}{z})/(c - a)$	Computed so that the lower ends of the equation plot will have a separation distance equal to 1 from the x axis.

Fig. 8 Generating next candidate according to PDF **Fig. 9** Terminating criteria, a) empirical data of termination point, b) theoretical termination graph

the upper limit of iterations i.e., the temperature minimum value T_{min} . However, in most cases, the best solution is found halfway to this limit. Therefore, A second termination condition is set which is triggered when no improvement is observed in a row of iterations; A counter for the number of successive rejected candidates is used to hit the termination criterion when reaching a predetermined maximum value. Fig. 9 shows empirical data of the number of iterations before termination for different point-set domain sizes. As the domain gets larger the termination occurs at a later stage of the execution. For large domain size, the algorithm may not terminate according to the second condition and the improvement will continue to appear but with insignificant change in the objective value, thus, the algorithm is forced to terminate when it reaches the maximum number of iterations (first termination condition).

6 Setting up the algorithm parameters

SA performance is highly dependent on the control parameters such as maximum temperature, number of iterations, number of epochs, and cooling rate [25]. However, as long as an SA-based algorithm runs for a long time, the setting of these parameters may not be a critical issue [1]. Adopting parameters that lead to a long execution time may yield an accurate solution, though it is unpractical to go through all possible combinations of the design variables. Hence, a trade-off has been made between accuracy and execution time in favor of accuracy. First, a systematic method is applied to set up the algorithm parameters for one specified input. then generalization has been made to adapt any type of input.

6.1 Initial configuration of parameters

In this study, the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is adopted to set up the DNSA parameters [13]. The TOPSIS was firstly proposed by Hwang and Yoon [16] and modified later by Hwang, Lai, and Liu [17]. It is categorized under the Multiple Attribute Decision-making techniques (MADM).

Before applying the TOPSIS method we perform a pre-evaluation phase to determine the best range of values for the control parameters at which DNSA appears to have a good performance. This is done by testing all parameters individually while maintaining an average estimation for the other parameters. The parameter alternatives are then arranged in all possible combinations and tabulated as in Table 2. Cooling rate indexes (1,2 and 3) refer to three tested formulas: Exponential multiplicative cooling, Logarithmical multiplicative cooling, and Linear multiplicative cooling respectively. Group size is the maximum number of last accepted candidates that are maintained to direct the search region.

The performance of DNSA is evaluated using three criteria:

1. The quality of the final solution, i.e., how close is the final solution to the global minimum.
2. The algorithm execution time.
3. The stability, i.e., measurement consistency of the previous two criteria.

For each alternative in Table 2, the algorithm was made to run 30 times to acquire some measurements for the evaluation process. Four Measurement metrics are taken during these runs to quantify the three criteria. These metrics are the average and deviation in measured executing times as well as the average and deviation in measured objective values denoted as μ_t , σ_t , μ_F , and σ_F respectively. The two average metrics quantify the first two criteria. Whereas stability is quantified by the two metrics of standard deviation. Comparing different alternatives using standard deviation and average values is acceptable since all test runs are subjected to the same input.

After estimating the four metrics, the metric's normalized score for each alternative is computed using vector normalization. The alternatives and the normalized performance indicators are known as the evaluation matrix. Each normalized performance indicator is then multiplied by a specific weight that indicates the significance of the criterion. Weights are estimated via Analytic Hierarchy Process (AHP) [32]. The importance of each performance indicator is quantified on a scale from 1 to 9.

The outcome of TOPSIS is ranking values of all alternatives based on their distance to the positive ideal solution (PIS) V_j^+ . The ideal solution is a hypothetical solution that hits the best score in each criterion. On the other hand, negative ideal solution (NIS) V_j^- refers to the distance to the worst solution. The distance of each alternative to the NIS and PIS is computed by Eqs. 17 and 18. Finally, the alternatives are ranked according to their closeness C_i using Eq. 19.

$$S_i^+ = \left[\sum_{j=1}^m (V_{ij} - V_j^+)^2 \right]^{0.5} \quad (17)$$

$$S_i^- = \left[\sum_{j=1}^m (V_{ij} - V_j^-)^2 \right]^{0.5} \quad (18)$$

$$C_i = \frac{S_i^-}{S_i^+ + S_i^-} \quad (19)$$

The evaluation process using TOPSIS yields an optimum configuration as seen at the row associated with

Fig. 10 (a) Inputs of different first derivative entropies vs dispersion in the final solution (CV), (b) Change in dispersion at different values of the correction factor

the rank value equal to 1 in Table 2. So far, all alternatives have been evaluated under the same input. Thus, the obtained configuration might not be preferable for other types of inputs [25].

6.2 Adaptation to input complexity

Intuitively, Inputs of complex shapes require a relatively longer execution time to equal the accuracy level for those of simple shapes. To study the algorithm performance under different inputs. First, we specify a measurement metric to indicate the complexity of an input. Then, the algorithm is left to run 30 times for each input to observe the dispersion in the final solution. The complexity is designated by the Shannon entropy metric to capture diversity in some raw information i.e., first derivative values. Eq. 20 is used to calculate the entropy in first derivative values for each input where P_i is the occurrence rate of a value in the set of all values at a specific measurement resolution [34]. The measurement resolution is set equal to 0.01. Since the first derivative values lie in the range $[-1, 1]$ then maximum entropy under the specified resolution is, therefore, equal to $\log_2(1 - (-1)/0.01) = 7.644$. Inputs of different entropies are compared to another metric for dispersion; Coefficient of Variation (CV) is sufficient for this purpose considering that we are comparing different inputs.

$$H = \sum_{i=1}^n P_i \log_2 \left(\frac{1}{P_i} \right) \quad (20)$$

Empirical results prove the linear proportionality between the complexity metric (H) and the result variation metric (CV) as illustrated in Fig. 10 (a). For sake of stable performance over different entries, the number of iterations is made to auto-adapt the complexity of a given input. In other words, the number of iterations is increased by a factor so

Table 2 Performance and TOPOSIS based ranking found for each alternative

#T steps	Iterations per Epoch	T_{max}	Group size	Cooling formula	μ_t	σ_t	μ_F	σ_F	C_i	Rank
300	30	1000	10	0	0.0299	0.0203	0.1631	0.039	0.9682	1
300	30	1000	10	1	0.1048	0.0615	0.164	0.067	0.8603	10
300	30	1000	10	2	0.1075	0.0913	0.1628	0.0272	0.8697	9
300	30	1000	20	0	0.0519	0.0331	0.1632	0.0351	0.9442	2
300	30	1000	20	1	0.0975	0.0651	0.1634	0.0415	0.8829	8
300	30	1000	20	2	0.1411	0.116	0.1655	0.0965	0.7883	16
300	30	100	10	0	0.1418	0.0799	0.1646	0.0934	0.7978	14
300	30	100	10	1	0.0778	0.0854	0.163	0.0491	0.8956	7
300	30	100	10	2	0.057	0.0428	0.1644	0.078	0.8969	6
300	30	100	20	0	0.0542	0.0405	0.1634	0.0541	0.9266	3
300	30	100	20	1	0.076	0.0476	0.164	0.0429	0.91	5
300	30	100	20	2	0.0557	0.0472	0.1638	0.0651	0.9125	4
30	300	1000	10	0	0.0408	0.0501	0.1712	0.2544	0.6782	24
30	300	1000	10	1	0.0186	0.0177	0.1751	0.3593	0.5921	33
30	300	1000	10	2	0.0313	0.0173	0.168	0.1265	0.8445	12
30	300	1000	20	0	0.0581	0.0779	0.1715	0.1498	0.7946	15
30	300	1000	20	1	0.011	0.0131	0.177	0.1955	0.757	19
30	300	1000	20	2	0.0198	0.0109	0.1695	0.1972	0.7549	21
30	300	100	10	0	0.0202	0.014	0.178	0.3357	0.6106	31
30	300	100	10	1	0.0424	0.0541	0.172	0.2566	0.6746	25
30	300	100	10	2	0.028	0.0164	0.1722	0.4081	0.5547	35
30	300	100	20	0	0.0585	0.064	0.1731	0.2847	0.6377	29
30	300	100	20	1	0.0514	0.0411	0.1735	0.3318	0.5999	32
30	300	100	20	2	0.0421	0.0484	0.178	0.2951	0.6353	30
300	300	1000	10	0	0.1809	0.1688	0.163	0.0606	0.7608	18
300	300	1000	10	1	0.1667	0.0772	0.1625	0.0286	0.8077	13
300	300	1000	10	2	0.1756	0.1328	0.1625	0.0359	0.7856	17
300	300	1000	20	0	0.3079	0.2091	0.1627	0.037	0.6449	28
300	300	1000	20	1	0.2132	0.2087	0.1627	0.0269	0.7327	22
300	300	1000	20	2	0.1236	0.0892	0.1626	0.037	0.8503	11
300	300	100	10	0	0.2876	0.1716	0.1625	0.0407	0.6683	26
300	300	100	10	1	0.2046	0.1286	0.1629	0.0415	0.7549	20
300	300	100	10	2	0.293	0.2553	0.1626	0.0384	0.6463	27
300	300	100	20	0	0.499	0.7413	0.1624	0.0286	0.4383	36
300	300	100	20	1	0.3571	0.3075	0.1636	0.0581	0.5738	34
300	300	100	20	2	0.2273	0.1532	0.163	0.0617	0.7181	23

that the new number of iterations equals $d(1 + k_h)$. The coefficient k_h is proportional to the entropy value and is determined empirically by testing the performance of one input over a range of k_h values. Fig. 10 shows empirical data for input of entropy value $H=4.6$. In this study, CV values less than 1 are accepted which corresponds to a coefficient value equal to 0.8. Extrapolating for other entropy values, the coefficient is approximated as $k_h = (0.8/4.6) \times H$. The number of iterations results from the TOPSIS method is 300, accordingly, for the input of maximum entropy value the number of iterations is then equal to $(k_{h_{max}} + 1) \times 300 = ((0.8/4.6 \times 7.644) + 1) \times 300 = 698$.

7 Results and Discussion

7.1 Examples

7.1.1 Part one

The examples in Table 3 are CAD models of mechanical parts to be machined. These parts contain features of high measurement accuracy demands (highlighted in blue). The optimization process is carried out for each part to obtain the optimum fixture layout that most satisfies the requirement of robustness. During machining, undesirable infinitesimal movement pivoting on the locating points can distort the relative distancing of these features. Placing the locators at some favorable locations can assist in diminishing the impact of this source of error. Accordingly, reference points are chosen to represent these features which will direct the computation of the objective value.

Fig. 11 2-D workpiece benchmark examples

Based on the complexity index of each CAD input, the algorithm number of iterations is adjusted accordingly to assure satisfactory accuracy of the obtained result as explained in section 6.2. The attained reduction in the size of the combinatorial problem is expressed as the ratio of the total number of alternatives to the number of candidates examined by the algorithm. For a point-set domain of size equal to 374 points, the number of alternatives is $\binom{374}{3} = 8649124$, and the number of examined alternatives is $337(\text{iterations}) \times 30(\text{epochs}) = 10110$, thus the reduction in the problem size is $10110 \times 100/8649124 = 0.10\%$.

Throughout the examples, the proposed method has efficiently generated the optimum fixture layouts. Different runs may result in different layout alternatives though all are acceptable selection since their objective values are close to each other. Execution time for each input seems to be dependent on three factors (entropy index, number of reference points, and a third factor relevant to the machine used to run the algorithm). More discussion is conducted in the second part of the examples where more focus is drawn to the quality of the obtained result.

7.1.2 Part two

For comparison, A set of benchmark examples (Fig. 11) drawn from the state of the art [5, 20, 39, 8, 41] are presented in the subsequent lines. Numerical values using our method are provided for workpieces (a) and (b), while (b) and (d) are left for qualitative comparison. Reference points for these examples are chosen to represent the entire workpiece and no specific inner features are considered. In the rectangular workpiece example (Fig. 11. a), the obtained fixture schema is similar to that in [20] but differs slightly in the coordination of each locator. In our model, the workpiece

is assumed to be a rigid body. Running the optimization using this model results in positioning the locators $L1$ and $L2$ close to the corner. On the other hand, the aforementioned study employed finite element analysis which counts for inner deformation. As a result, the locators $L1$ and $L2$ in the former mentioned study tend to be positioned far from the corners. This can be interpreted by the fact that corners are more brittle and vulnerable to fracture therefore not suitable for locator placement. Adopting models that use FEA may result in a more accurate solution for workpiece with significance degree of inner deformation. Nevertheless, Employing FEA holds high computational cost, and for each candidate, we need to generate a new mesh for the evaluation. On the contrary, rigid body models are fast to compute and suitable to implement as a subroutine task in a CAD environment. In the hexagon workpiece (Fig. 11. d), the result of our study is almost identical to that found in [5] and [39] since all are built on the same concepts of kinematic analysis of rigid bodies but with different objective functions. The aero-foil cross-section example (Fig. 11. c) is brought from the study [41] which concluded with three preferable fixture layouts. The fixture layout results from the objective function 7 is similar to the 4th fixture layout found in the aforementioned study. A better result is obtained by using the objective function 9 which count for curvature. The obtained fixture layout is observed to have the locators ($L1'$, $L2'$, and $L3'$) drifted away from regions of high curvature as expected. In the eclipse workpiece (Fig. 11. b), the result obtained from the objective function 7 as well as the objective function 9 is similar to what is found in [8] (referred in the study as infinitesimal error analysis and small error analysis respectively). In the context of the optimization method, the former study has specified an initial position and optimization direction for each locator before carrying the optimization process. In our study, however, no starting points nor searching direction is required though it is possible to constrain the placement of each locator to a specific region. The results of DNSA using the two objective functions are shown in Table 4 and Table 5. It is observed that objective values result from function 9 are higher compared to the objective values result from function 7 due to the extra cost added when considering the effect of curvature error.

7.2 Algorithm general aspects

7.2.1 Exploration and exploitation

Exploration and exploitation are essential characteristics for any heuristic algorithm. While exploration indi-

Table 3 Fixture layout optimization of different mechanical parts

Workpiece
Point-set domain size	374	434	390	418
Reference points
Complexity index (Shanon entropy)	1.165	2.143	3.119	4.971
Maximum number of iterations	337	412	463	560
Reduction in problem size	0.10%	0.09%	0.14%	0.14%
Average execution time (sec.)	4.1	4.5	2.8	10.9
Optimum alternative 1				
$F(\mathbf{X})$	2.15	2.20	2.35	2.48
Fixture layout
Optimum alternative 2				
$F(\mathbf{X})$	2.15	2.22	2.36	2.45
Fixture layout
Optimum alternative 3				
$F(\mathbf{X})$	-	2.19	2.29	2.51
Fixture layout	-

Table 4 Fixture layout for eclipse workpiece – curvature is ignored

	L1		L2		L3		F(X)
	X	Y	X	Y	X	Y	
Run1	63	309	45	126	654	366	2.68
Run 2	676	358	68	312	34	142	2.62
Run 3	41	303	29	126	638	366	2.73
Run 4	41	303	638	366	29	126	2.73

Table 5 Fixture layout for eclipse workpiece – curvature is considered

	L1		L2		L3		F(X)
	X	Y	X	Y	X	Y	
Run1	168	369	87	93	609	381	6.55
Run 2	609	361	87	73	168	349	6.55
Run 3	593	381	71	93	152	369	6.63
Run 4	593	361	152	349	71	73	6.63

cates the discovery of new regions of the search space, exploitation is carried within a narrower region close to the previously visited points. A trade-off between exploration and exploitation is necessary for any successful algorithm [11]. This has been implicitly achieved in our algorithm through the use of TOPSIS method. A transition from a large neighborhood search (exploration stage) to a small neighborhood search (exploitation stage) is the main characteristic of the DNSA algorithm.

7.2.2 Memory usage search

A very powerful element in metaheuristic algorithms is the usage of memory. In other words, taking advantage of previous records of accepted candidates to guide the search for new candidates [4]. In SSA, the generation of the next candidate is performed according to the Markov chains rule. It exclusively utilizes the information of the current state to resolve the transition to the next state. Therefore, it is a memory-less algorithm. In our method, however, records of the last accepted candidates are exploited to guide the search space.

7.2.3 Characteristic graphs

The convergence of the algorithm can be visualized by the set of characteristic graphs shown in Fig. 12. These graphs were taken for the previous eclipse example. Chart (a) shows the reduction in the acceptance probability (Eq. 11) through the execution progress. Chart (b) shows the changes in the objective value, the final optimum solution is likely to be found at a later stage of the algorithm execution but it could also occur at an early stage. Chart (c) shows the changes in the PDF mean for each locator. At the early stage of the algorithm execution, the fluctuation of the mean values is

Fig. 12 Characteristic graphs of the DNSA algorithm

very high then it gradually stabilizes. Chart (d) shows the change in temperature value every 9 successive iterations.

7.3 Deterministic locating

As stated previously, a satisfactory deterministic locating arrangement is associated with a Jacobian matrix determinant not equal to zero. A problem that arises under the approximated approach of discretization is that some indeterministic arrangement of locators are being quantified with a determinant not equal but close to zero. In fact, this bug has no impact on the optimization process since an arrangement with low Jacobian value will result in a high objective value and, consequently, will be excluded. This is true as far as the input shape permits a deterministic arrangement of locators, Some inputs cannot accommodate a deterministic set of locators due to boundary restriction as in circle [10].

To enhance algorithm robustness, we propose the use of a threshold value to reject candidates having an absolute determinant value less than a threshold value. This value is roughly determined empirically using an input of a circular workpiece. A circle shape is very useful to estimate the threshold value; any arrangement of locators on the circle perimeter must have, by analytical meaning, a zero-determinant value. This is obvious since a circle rotational degree of freedom cannot be constrained under any set of locators. Applying our approach for a circle shape input must also work in the same way and any non-zero determinant value must be regarded as zero. To estimate the threshold value, we run the algorithm on a circle of a given diameter while recording the absolute determinant values for all candidates. The result shows that these values are having an upper bound (threshold) equal to $(d + 1)/2$. For

non-circular shapes, the threshold is equivalent to the threshold of the minimum diameter circle that bound the entire shape.

7.4 Accuracy of the obtained result

The result obtained in this study is not necessarily the global optimum but an adequate approximation. The source of errors that occurred during the implementation of our method can be confined to three bounded error values. The first bounded error is owing to the validity of adopting the rigid body fixture model. In this model, the deformation of the workpiece is neglected which is a fair assumption to model relative stiff components. However, in some cases the deformation is influential and it becomes essential to embed its effect into the model. Therefore, other models such as workpiece–fixture elastic contact model and workpiece elastic model might be better alternatives to represent the problem. The second bounded error is owing to the approximated representation of the workpiece surface points. In this study, a uniform mesh is used to automate the discretization process for any given 2-D workpiece. The accuracy of this representation is affected by the mesh parameter α , the smaller α the higher the accuracy. The third bounded error is owing to the result accuracy of the optimization algorithm. Different runs of the DNSA algorithm often yield different results. In setting up the algorithm parameters, the deviation from the global optimum is made as minimal as possible. Despite these bounding errors, the DNSA algorithm provides sufficient result in view of accuracy and performance as shown in the previous examples.

8 Conclusion and future work

Robust arrangement of fixture layout contributes positively to the measurement accuracy of the workpiece during machining. This paper presents a novel algorithm (DNSA) for fixture layout optimization. The aim is to find the robust fixture layout from a large space of candidates that best minimize the impact of locator localization errors on the accuracy of the machined features. The selection of candidates is subjected to a probability function over a searching space that declines gradually with the progress of execution. The main characteristic of the DNSA is its transition from an explorational state to exploitation. To reach the best algorithm performance, the algorithm control parameters were systematically configured using the TOPOSIS method taking one specific input, yet applying the algorithm for different types of inputs shows variation

in performance. It is found that for any input, the algorithm performance metric (coefficient of variance of the final objective values) is proportional to the input complexity metric (based on Shannon entropy). Therefore, a stable performance was maintained for different inputs by increasing the number of iterations in proportion to the complexity metric. The proposed algorithm is proven to be an efficient technique and computable to other algorithms used in the field i.e., GA and ACA.

This work contributes further to CAD-based methods for automated fixture design. The implementation of our approach from the discretization to the computing of the objective value was carried out using the application programming interface NXOpen and the user interface NX. To conduct the optimization for a given 2-D workpiece, the user needs only to specify the face entity in the CAD model and the features of critical measurement accuracy.

As part of future study, integrating two types of models, specifically the rigid body workpiece model and the elastic workpiece model, might be more beneficial. The result obtained from our model (Rigid body-based model) is closed to that of the elastic workpiece model found in the literature. Moreover, the evaluation process is fast to compute and can produce sufficient result accuracy when deformation is not influential. However, the use of the elastic workpiece model becomes essential when inner deformation is significant as in high-duty machining, but since the evaluation of this model requires the usage of Finite Element Analysis, their computational cost is very high and it becomes impractical to evaluate every possible candidate for large search space. Therefore, we suggest further investigation in integrating both models so that the rigid body model is used in a preliminary phase to filter the searching space to a small number of best-fitting candidates. Then the elastic workpiece model is used in a secondary phase to select the optimum solution while counting for inner deformation.

Acknowledgements We would like to acknowledge all of, Mr. Zhu Yiheng who helped to assure coherency and sufficiency of the content through face-to-face discussion session. And Mr. Ahmed Magram for his effort in the linguistic and logical revision of the final draft.

Declarations

Funding

The authors did not receive support from any organization for the submitted work.

Conflicts of interest

The authors declare that they have no conflict of interest.

Availability of data and material

Not applicable

Code availability

Upon request, the authors are pleased to offer to provide the C++ code and the NX template for the result recalculation of the given examples.

Authors' contributions

Taqiaden Alshameri: Conceptualization, Methodology, Software, Formal analysis, Writing Original draft Yude Dong: Supervision, Resources, Validation Abdullah Alqadhi: Visualization, Writing Reviewing and Editing.

Ethics approval

All procedures performed in the study were in accordance with the ethical standards of the institutional and national research committee and with the COPE guidelines.

Consent to participate

Informed consent was obtained from all individual participants included in the study.

Consent for publication

The participant has consented to the submission of the case report to the journal.

References

1. Anily S, Federgruen A (1987) Simulated annealing methods with general acceptance probabilities. *Journal of Applied Probability* 24(3):657–667, DOI 10.2307/3214097
2. Asada H, By A (1985) Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixtures. *IEEE Journal on Robotics and Automation* 1(2):86–94, DOI 10.1109/JRA.1985.1087007
3. Asante JN (2008) A combined contact elasticity and finite element-based model for contact load and pressure distribution calculation in a frictional workpiece-fixture system. *The International Journal of Advanced Manufacturing Technology* 39(5-6):578–588, DOI 10.1007/s00170-007-1187-5
4. Blum C, Roli A (2003) Metaheuristics in combinatorial optimization. *ACM Computing Surveys* 35(3):268–308, DOI 10.1145/937503.937505
5. Brost R, Goldberg K (1994) A complete algorithm for synthesizing modular fixtures for polygonal parts. In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, IEEE Comput. Soc. Press, pp 535–542, DOI 10.1109/ROBOT.1994.351243
6. Brost R, Goldberg K (1996) A complete algorithm for designing planar fixtures using modular components. *IEEE Transactions on Robotics and Automation* 12(1):31–46, DOI 10.1109/70.481749
7. Cai W (2006) Robust pin layout design for sheet-panel locating. *The International Journal of Advanced Manufacturing Technology* 28(5-6):486–494, DOI 10.1007/s00170-004-2402-2
8. Cai W, Hu SJ, Yuan JX (1997) A Variational Method of Robust Fixture Configuration Design for 3-D Workpieces. *Journal of Manufacturing Science and Engineering* 119(4A):593–602, DOI 10.1115/1.2831192
9. Černý V (1985) Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications* 45(1):41–51, DOI 10.1007/BF00940812
10. Chou YC, Chandru V, Barash MM (1989) A mathematical approach to automatic configuration of machining fixtures: Analysis and synthesis. *Journal of Manufacturing Science and Engineering, Transactions of the ASME* 111(4):299–306, DOI 10.1115/1.3188764
11. Črepinšek M, Liu SH, Mernik M (2013) Exploration and exploitation in evolutionary algorithms. *ACM Computing Surveys* 45(3):1–33, DOI 10.1145/2480741.2480752
12. Dan Ding, Guoliang Xiang, Yun-Hui Liu, Yu Wang M (2002) Fixture layout design for curved workpieces. In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, IEEE, vol 3, pp 2906–2911, DOI 10.1109/ROBOT.2002.1013673
13. Fotuhi F (2011) Optimal determination of simulated annealing parameters using TOPSIS. In: *2011 IEEE International Conference on Industrial Engineering and Engineering Management*, IEEE, pp 46–50, DOI 10.1109/IEEM.2011.6117876

14. Goldman R (2005) Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design* 22(7 SPEC. ISS.):632–658, DOI 10.1016/j.cagd.2005.06.005
15. Granville V, Rasson JP, Krivánek M (1994) Simulated Annealing: A Proof of Convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(6):652–656, DOI 10.1109/34.295910
16. Hwang CL, Yoon K (1981) Methods for Multiple Attribute Decision Making. pp 58–191, DOI doi.org/10.1007/978-3-642-48318-9_3
17. Hwang CL, Lai YJ, Liu TY (1993) A new approach for multiple objective decision making. *Computers and Operations Research* 20(8):889–899, DOI 10.1016/0305-0548(93)90109-V
18. Kaya N (2006) Machining fixture locating and clamping position optimization using genetic algorithms. *Computers in Industry* 57(2):112–120, DOI 10.1016/j.compind.2005.05.001
19. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by Simulated Annealing. *Science* 220(4598):671–680, DOI 10.1126/science.220.4598.671
20. Krishnakumar K, Melkote SN (2000) Machining fixture layout optimization using the genetic algorithm. *International Journal of Machine Tools and Manufacture* 40(4):579–598, DOI 10.1016/S0890-6955(99)00072-3
21. Liao YG (2003) A genetic algorithm-based fixture locating positions and clamping schemes optimization. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 217(8):1075–1083, DOI 10.1177/095440540321700805
22. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics* 21(6):1087–1092, DOI 10.1063/1.1699114
23. Padmanaban KP, Prabhakaran G (2008) Dynamic analysis on optimal placement of fixturing elements using evolutionary techniques. *International Journal of Production Research* 46(15):4177–4214, DOI 10.1080/00207540601147297
24. Padmanaban KP, Arulshri KP, Prabhakaran G (2009) Machining fixture layout design using ant colony algorithm based continuous optimization method. *The International Journal of Advanced Manufacturing Technology* 45(9-10):922–934, DOI 10.1007/s00170-009-2035-6
25. Park Mw, Kim YD (1998) A systematic procedure for setting parameters in simulated annealing algorithms. *Computers and Operations Research* 25(3):207–217, DOI 10.1016/S0305-0548(97)00054-3
26. Penev K, Requicha A (1997) Automatic fixture synthesis in 3D. In: *Proceedings of International Conference on Robotics and Automation, IEEE*, vol 2, pp 1713–1718, DOI 10.1109/ROBOT.1997.614393
27. Phoomboplab T, Ceglarek D (2007) Design Synthesis Framework for Dimensional Management in Multistage Assembly System. *CIRP Annals* 56(1):153–158, DOI 10.1016/j.cirp.2007.05.037
28. Prabhakaran G, Padmanaban KP, Krishnakumar R (2007) Machining fixture layout optimization using FEM and evolutionary techniques. *International Journal of Advanced Manufacturing Technology* 32(11-12):1090–1103, DOI 10.1007/s00170-006-0441-6
29. Qin G, Zhang W, Wan M, Qin G (2006) A mathematical approach to analysis and optimal design of a fixture locating scheme. *The International Journal of Advanced Manufacturing Technology* 29(3-4):349–359, DOI 10.1007/s00170-005-2509-0
30. Raghu A, Melkote SN (2004) Analysis of the effects of fixture clamping sequence on part location errors. *International Journal of Machine Tools and Manufacture* 44(4):373–382, DOI 10.1016/j.ijmachtools.2003.10.015
31. Rimon E, Burdick J (1998) Mobility of bodies in contact. I. A 2nd-order mobility index for multiple-finger grasps. *IEEE Transactions on Robotics and Automation* 14(5):696–708, DOI 10.1109/70.720346
32. Saaty TL (2002) Decision making with the Analytic Hierarchy Process. *Scientia Iranica* 9(3):215–229, DOI 10.1504/ijssci.2008.017590
33. Satyanarayana S, Melkote S (2004) Finite element modeling of fixture-workpiece contacts: single contact modeling and experimental verification. *International Journal of Machine Tools and Manufacture* 44(9):903–913, DOI 10.1016/j.ijmachtools.2004.02.010
34. Shannon CE WW (1998) *The mathematical theory of communication*. University of Illinois press
35. Shepard D (1968) A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings of the 1968 23rd ACM national conference on -*, ACM Press, New York, New York, USA, pp 517–524, DOI 10.1145/800186.810616
36. Vallapuzha S, De Meter EC, Choudhuri S, Khetan RP (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* 42(2):265–275, DOI 10.1016/S0890-6955(01)00113-

4

37. Vallapuzha S, De Meter EC, Choudhuri S, Khetan RP (2002) An investigation of the effectiveness of fixture layout optimization methods. *International Journal of Machine Tools and Manufacture* 42(2):251–263, DOI 10.1016/S0890-6955(01)00114-6
38. Vasundara M, Padmanaban K (2014) Optimization of Fixture Layout and Artificial Neural Network (ANN) Weights of ANN-Finite Element Analysis Based Fixture Layout Model Using Genetic Algorithm. *Journal of Engineering and Technology* 4(2):103, DOI 10.4103/0976-8580.141180
39. Wang M (2000) An optimum design for 3-D fixture synthesis in a point set domain. *IEEE Transactions on Robotics and Automation* 16(6):839–846, DOI 10.1109/70.897795
40. Wang MY, Pelinescu DM (2001) Optimizing fixture layout in a point-set domain. *IEEE Transactions on Robotics and Automation* 17(3):312–323, DOI 10.1109/70.938388
41. Wang Y, Chen X, Liu Q, Gindy N (2006) Optimisation of machining fixture layout under multi-constraints. *International Journal of Machine Tools and Manufacture* 46(12-13):1291–1300, DOI 10.1016/j.ijmachtools.2005.10.014
42. Yang F, Xing Y, Li X (2020) A comprehensive error compensation strategy for machining process with general fixture layouts. *The International Journal of Advanced Manufacturing Technology* 107(5-6):2707–2717, DOI 10.1007/s00170-020-05148-6
43. Zheng Y, Chew CM (2010) A geometric approach to automated fixture layout design. *Computer-Aided Design* 42(3):202–212, DOI 10.1016/j.cad.2009.10.002
44. Zheng Y, Qian Wh (2008) A 3-D modular fixture with enhanced localization accuracy and immobilization capability. *International Journal of Machine Tools and Manufacture* 48(6):677–687, DOI 10.1016/j.ijmachtools.2007.10.022