Workspace Synthesis for Flexible Fixturing of Stampings

Jinkoo Lee

S. Jack Hu

Allen C. Ward

Dept. of Mechanical Engineering and Applied Mechanics,
The University of Michigan,
Ann Arbor, Michigan 48109

Fixtures are used to position and hold parts for a series of assembly operations. In automotive body assembly, these fixtures conventionally have been dedicated, therefore they must be replaced whenever there are model changes in an auto body assembly plant. In recent years, however, the automotive industry has been changing from high volume to small-to-medium volume production per model with an increasing number of models because customer tastes are diversifying. To cope with this change, auto companies need to be capable of producing a variety of models in small-to-medium volume, and they rely on flexible assembly lines and flexible fixtures. These flexible fixtures use robots as programmable fixture elements so that they can be reprogrammed for different stamped sheet metal parts. When designing flexible fixtures, fixture designers need to be concerned with fixture workspaces for a set of different stampings. However, existing fixture design methods address the fixturing of one stamping only. This paper presents a system that fixture designers can use to synthesize flexible fixture workspaces for a set of different stampings. Based on circular workspaces for flexible fixture robots, this system finds optimal workspace sizes and centers on a fixture base plate with a graphical display for visual checking. This system is simple to use and produces results quickly.

1 Introduction

Fixtures are used to position and hold parts for assembly operations. In automotive body assembly, these fixtures conventionally have been dedicated, producing each model in high volume. In recent years, however, the automobile industry has been changing from high volume to small-to-medium volume production per model with an increasing number of models because customer tastes are diversifying. In other words, auto companies need to cope with fast market changes. As a result, they increasingly rely on flexible assembly lines that can manufacture a variety of vehicles in small-to-medium volume, unlike dedicated assembly lines that can manufacture only one vehicle type. These flexible assembly lines require flexible fixtures. These fixtures use robots as their programmable fixture elements (fixels) so that different stampings can be located in a flexible fixture by reprogramming rather than rebuilding. Another benefit of using flexible fixtures is that auto companies can reduce cost and time for fixtures because it requires high cost and long lead time to rebuild the dedicated fixtures whenever there are new model changes. For example, many auto companies replace current models every 2 to 5 years, and building new dedicated fixture costs millions of dollars and takes many months.

When designing flexible fixtures, fixture designers must consider all the models simultaneously. That is, they must be concerned with not only determining the locators for one stamping but also the fixture workspaces required for a set of different stampings. Research results have been published on fixture design methods for determining the locators for one stamping; these results are reviewed in section 2. However, there is no available method that fixture designers can use to synthesize flexible fixture workspaces for a set of different stampings. Thus, a new workspace synthesis system is presented in section 3. An example of synthesizing flexible fixture workspaces is illustrated in section 4, with conclusions and limitations given in section 5.

2 Existing Fixture Design Methods

There are two main issues in designing a dedicated fixture: the number of locators (and supports), and their locations. For a rigid part, six locators ("3-2-1") are used to restrain the 6 degrees of freedom of the workpiece. If we take the example of a cube, we can put 3 locators on the bottom surface, 2 locators on the left surface, and 1 locator on the back surface. However, a large stamping part requires more than 6 locators because the stamping can be deformed due to its own weight or welding forces. The following two subsections review literature relevant to the two main questions.

2.1 Number of Locators. Since stampings often require more than six locators to prevent deformation, determining the number of locators for a given stamping design is important. For example, as shown in Fig. 1, a front door inner panel is located by four NC blocks/clamps in the y direction, one four-way pin for the x and z directions, and one two-way pin (or a round pin in combination with a slot) for the z direction. This door inner panel has 4 locators in the y direction because the panel is flexible and can be easily deformed in this direction due to manufacturing process load or its own weight. In other words, this door inner panel has a "4-2-1" locating scheme. In general, a stamping may have an "N-2-1" locating scheme, where $N \geq 3$ (Cai, Hu and Yuan, 1996).

Rearick, Hu and Wu (1993) presented the first published work that addresses the problem of determining the number of locators optimally for sheet metal stampings. They formulated an optimization problem using an objective function combining the number of locators and the root mean squared deflections of the nodes, where the nodes were defined in the finite element model of a stamping. They also presented an algorithm to determine the number of locators for a given stamping design. If the number of locators increases, the sum of the root mean squared deflections decreases, but fixture cost increases. On the other hand, if the number of locators decreases, the cost of the fixture decreases, but deflection increases.

2.2 Placement of Locators. After determining the number of locators for one given stamping design, the next task is to find the positions of the locators on the given stamping design. When determining the positions of the locators, one major issue is to minimize the deflection of the stamping due to its own weight or external forces such as welding forces.

There are several published articles that address the problem of placing the locators on a deformable part such as a stamping.

Contributed by the Manufacturing Engineering Division for publication in the JOURNAL OF MANUFACTURING SCIENCE AND ENGINEERING. Manuscript received July 1996; revised March 1998. Associate Technical Editor: T. C. Woo.

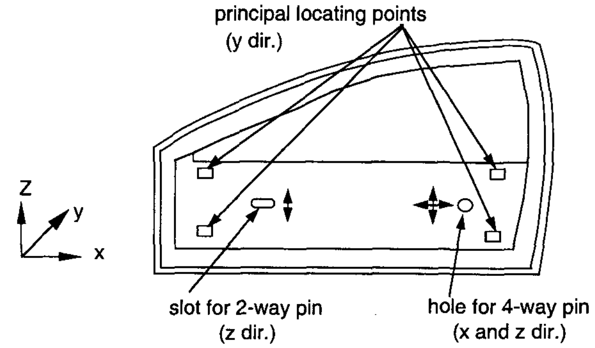


Fig. 1 Locators for a front door inner panel

Menassa and DeVries (1991) presented an optimal fixture design approach that minimizes the sum of deflections of a deformable part due to external loads. They assume that a deformable part is located by a "3-2-1" locating scheme, and search for the optimal positions of the 3 locators in the 3-2-1 locating scheme. Cai, Hu and Yuan (1996) improved the work of Rearick et al. (1993) in the areas of fixturing principles for stampings and remeshing problems in structural optimization, and presented another optimal fixture design approach to minimizing the sum squares of deflections of a stamping due to welding forces. Unlike Menassa and DeVries' work, they assume that a stamping is located by an "N-2-1" locating scheme. Thus, their approach includes an algorithm for determining "N" in the N-2-1 locating scheme.

All the methods reviewed above are for one given stamping design. Here we will present our initial work in designing fixtures for a set of stampings. The next section presents a flexible fixture workspace synthesis method and its computational implementation, which fixture designers can use to synthesize flexible fixture workspaces for a set of stampings.

3 System for Synthesizing Flexible Fixture Workspace

This section argues for the set-based approach in designing fixtures, describes the representation of fixture robot workspaces and locator regions, and presents a method to optimally synthesize flexible fixture workspaces for a set of different stampings. Stampings are manufactured through a series of press operations such as drawing, trimming, piercing, and flanging, and then these stampings are assembled into auto body parts, typically through spot welding operations. For example, as Fig. 2 shows, a door outer panel involves 3 press operations, and a door inner panel involves 5 press operations. After the door inner panel is assembled into a door inner assembly by spot welding operations, the door inner panel assembly and the door outer panel become a door assembly through a hemming process, and then the door assembly is assembled into a car body through a bolting

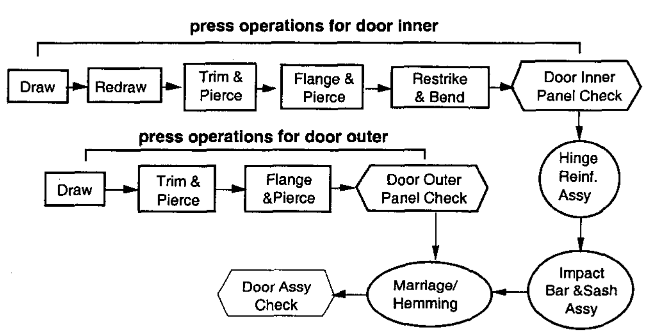


Fig. 2 A door manufacturing process

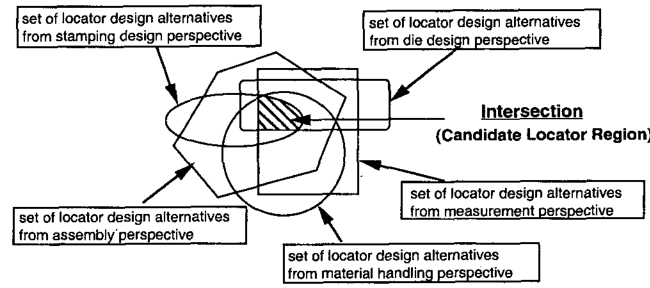


Fig. 3 Candidate locator region from multiple design perspectives

process. As the door manufacturing process involves several press operations and assembly operations, fixture designers need to work together with other team members for other perspectives as shown in Fig. 3. Thus, we recommend that fixture designers follow the set-based concurrent engineering approach (Lee, 1996).

Here, because of the set-based stamping design approach, we assume that there is a candidate locator region for each locator of a stamping (or a set of stamping design alternatives) rather than a fixed locating point. We can consider this candidate locator region as the intersection of many preferred regions by different team members, as shown in Fig. 3. For example, if a stamping is going to be located by *4* surface points (NC blocks) and *2* pins (for a hole and a slot) shown in Fig. 1, we assume that this stamping has 6 candidate locator regions rather than 6 fixed points in a stamping design stage. If we use a flexible fixture for this stamping, we need 6 fixture robots. A fixture robot is a robot manipulator with its end effector being a locating pin or block. Further, if a fixture robot is selected to cover the candidate locator region for a locator, any point in the candidate locator region can be chosen as the final locator. That is, a flexible fixture can be designed and built even before the final selection of the locator position. We also assume that the set of stampings to be handled by the same flexible fixture have the same N-2-1 locating scheme, requiring N blocks and 2 pins for a total of $N + 2$ fixturing robots.

With this background, we address how to synthesize the flexible fixture robot workspaces for M different stampings of the same locating scheme. If each stamping is assumed to have $N + 2$ candidate locator regions for the $N + 2$ locators and we position M stampings together on a flexible fixture base plate, we will have $(N + 2)*M$ candidate locator regions for M stampings. If we combine M candidate locator regions into one combined region for each locator, we will have $N + 2$ combined regions. For these $N + 2$ combined regions, we want to use a flexible fixture that has $N + 2$ fixture robots. In this setting, there are four problems in synthesizing flexible fixture workspaces: 1) how to represent robot workspaces and candidate locator regions, 2) how to find a stamping arrangement that enables us to use the smallest possible robots without overlapping between robot workspaces, 3) what size of robots to use, and 4) where to place the robots on a fixture base plate. The first problem is solved in section 3.1, and the remaining three problems are solved in section 3.2.

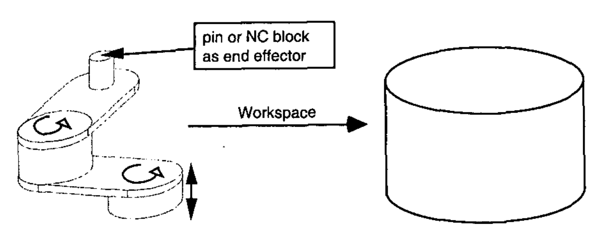


Fig. 4 Robot (PRR Joints) and its workspace

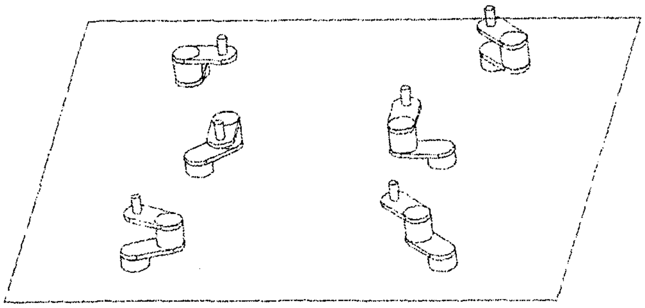


Fig. 5 Robot placements on a flexible fixture base plate

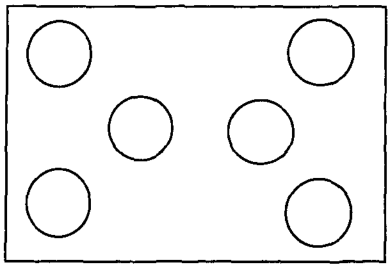


Fig. 6 Representation of robot workspaces as circles

3.1 Representation of Robot Workspace and Candidate Locator Region. First, we address the representation of the robot workspaces of a flexible fixture. Here we assume that all the robots have the Prismatic Revolute Revolute (PRR) joint configurations (FANUC, 1994), so that their workspaces are represented as filled cylinders as shown in Fig. 4. All the robots have either a pin (for a hole or a slot) or an NC block (for a surface point) as their end effectors. There can be other configurations such as PPP (Prismatic Prismatic Prismatic) joints: the workspace of such a robot is a cube. For the cylindrical workspace of the PRR robot, we are concerned with the radius and the center of the cylinder because of possible collisions. Thus, we consider only the center and the radius of the circle obtained by projecting the cylindrical workspace on a 2D plane. For example, Fig. 5 shows the placements of the 6 robots of a flexible fixture, and Fig. 6 shows 6 circles as our representation of 6 robot workspaces. In short, we represent a robot workspace as a circle on a 2D plane.

Second, we address the representation of candidate locator regions. For example, there are three door inner panels of different sizes as shown in Figs. 7, 8, and 9, and these panels have the same locating scheme. Each panel has 6 (4 surface points + 1 hole + 1 slot) candidate locator regions and each candidate locator region is defined by a set of 3D points (or the vertices of a polyhedron representing a candidate locator region). We assume that there is a flexible fixture base plate and we map all the points of candidate locator regions for all the panels onto a plane of the base plate. In short, we represent a candidate locator region as a polygon in 2D, which is again represented by its vertices.

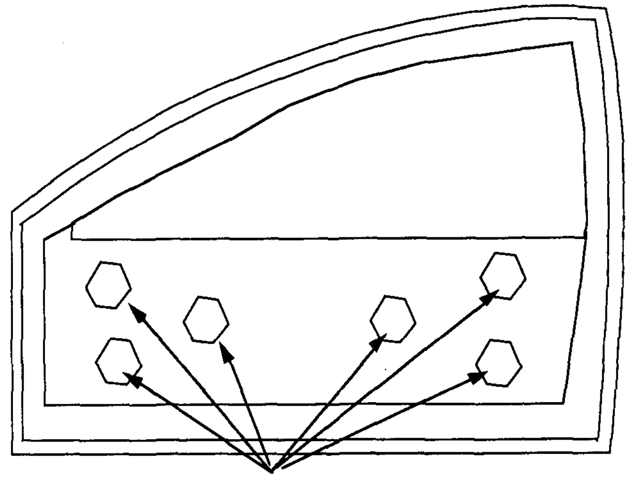


Fig. 8 Candidate locator regions for a door inner panel of Model B

3.2 Algorithm and Its Computational Implementation. This section discusses how to find a stamping arrangement that enables the use of the smallest possible robots without collisions among them. This stamping arrangement simultaneously solves the problems of what size robots to use and where to place the robots on a flexible fixture base plate.

We assume that M stampings have $M*(N + 2)$ polygons for candidate locator regions and these polygons form $N + 2$ combined regions to be covered by $N + 2$ circles representing fixture robots. The $N + 2$ combined regions can overlap with each other or become large depending on stamping arrangement. For example, let's assume that there are three stampings (A, B, and C) and each stamping has 6 polygons for 6 candidate locator regions (1 to 6) as shown in Fig. 10. If we put these three stampings together on a flexible fixture base plate, we will have 18 polygons, and three polygons (e.g., $A1$, $B1$, and $C1$) form one combined region that needs to be within one circle (e.g., circle 1) for one robot. This example has 6 combined regions for 6 robots as shown in Fig. 10. If we move stamping A downward, the 6 combined regions formed by the 18 polygons of three stampings will become larger. If we compute one minimum circle that encloses all the vertices in a combined region, we have 6 circles, and these 6 circles may or may not overlap depending on the arrangement of stampings. In other words, if we cleverly arrange stampings, the circles can be small and without overlap so that we can use small size robots without worrying about collisions between robots. In this setting, the problem we are interested in is how to find the optimal stamping arrangement that minimizes the circle sizes without overlap between circles. Algorithm Scheme 1 solves this problem.

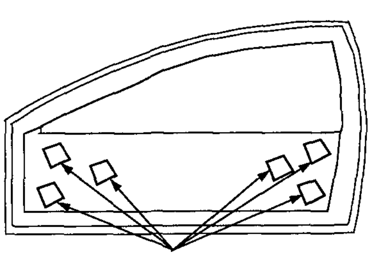


Fig. 7 Candidate locator regions for a door inner panel of Model A

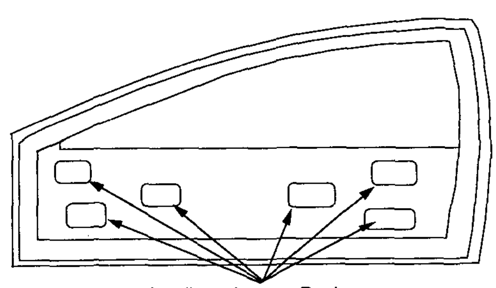


Fig. 9 Candidate locator regions for a door inner panel of Model C

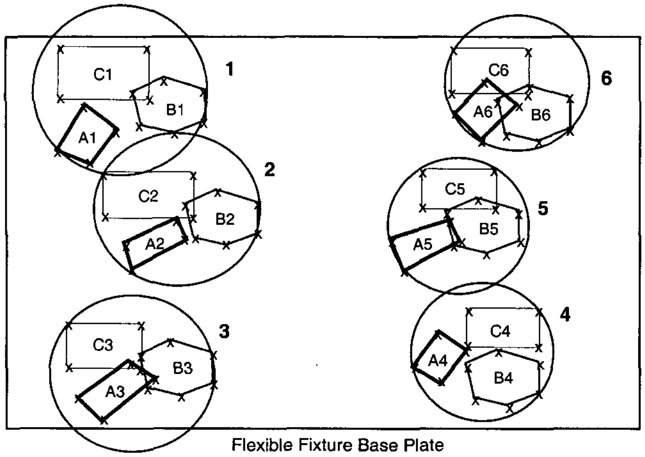


Fig. 10 Robot workspaces and candidate locator regions for three stampings

Algorithm Scheme 1: Finding optimal stamping arrangement, robot size, and robot center

(1) Designate one stamping as stationary (arbitrary selection) Repeat (Probabilistic Optimal Search Loop: Steps 2–5)

(2) Use a probabilistic optimal search technique to determine the values for the movement variables for the remaining ($M - 1$) stampings in 2D within a specified domain (2 translations & 1 rotation per stamping, thus $3*(M - 1)$ movement variables)

(3) Compute the new coordinates of the vertices of $(N + 2)*(M - 1)$ polygons for candidate locator regions (The stationary stamping and $M - 1$ stampings form $N + 2$ combined regions. Each combined region is a set of points.)

(4) For each combined region of $N + 2$ combined regions, compute the minimum circle enclosing M candidate locator regions of the combined region

(5) Calculate the objective function (a function of circle size and overlap) Until the minimum value of the objective function is found or search time is up

(6) Display the circles graphically on a computer screen, and print

—the stamping movements for stamping arrangement
—the circle diameters for robot sizes
—the circle centers for robot placements

In this research, we employed a probabilistic optimal search computer program, a genetic algorithm based computer program (Grefenstette, 1991). This programs determines the values for the movement variables and this corresponds to Step 2 of Algorithm Scheme 1. The details of the genetic algorithms are referred to in Goldberg (1989) and Davis (1991). Example input data for Grefenstette's program is described in Fig. 16. Other probabilistic optimal search programs, such as simulated annealing (van Laarhoven et al., 1987; Press, 1992), can also be used for our purposes.

Algorithm Scheme 1 is conceptually similar to the optimal blank nesting problem presented by Jain et al. (1992). In Jain's work, the computer reads in part geometry and metal strip information as an input, and displays as an output the optimal part layout that gives the minimal scrap without part overlapping. They employ simulated annealing for the optimal search process.

The following four subsections are allocated for the details of Steps 3 to 6 of Algorithm Scheme 1: the first for Step 3, the second for Step 4, and the third for Steps 5 and 6.

3.2.1 Computation of a Set of Points after Moving Stampings. This subsection describes the details of Step 3 of Algorithm Scheme 1. Here the points are the vertices of polygons.

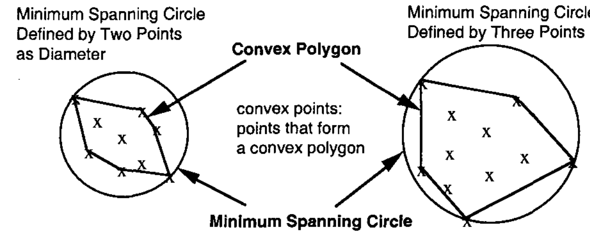


Fig. 11 Minimum spanning circle of a set of points

First, we show how we can compute the new point coordinates of a candidate locator region after moving a stamping in two translations and one rotation. For example, from Fig. 10 we define the points (x in this figure) of candidate locator region 1 of stamping A as P_{A1}, the points of region 2 as $P_{A2}, \ldots$, and the points of region 6 as P_{A6}. In a similar way, we define P_{B1}, $P_{B2}, \ldots, P_{B6}, P_{C1}, P_{C2}, \ldots, P_{C6}$ for stampings B and C. If move stamping A by h_A, v_A, and θ_A, we can compute a transformation matrix T_A using Eq. (1). The values of h_A, v_A, and θ_A are determined by the genetic algorithm. Here, we use homogeneous coordinates because they allow the translations and rotations to be represented by a single matrix. As an example of the homogeneous coordinate representation of a point, Eq. (2) shows the representation of the i-th point of candidate locator region 1 of stamping A. A new point (np) of stamping A after its movement can be computed using Eq. (3). Similar equations can be derived for stamping B, whose transformation matrix is composed of h_B, v_B, and θ_B. If we define NP_{A1} (or the collection of np_{A1i}) as the new points of candidate locator region 1 of stamping A, then $NP_{A1} = T_A \times P_{A1}$. The points of the other candidate locator regions can be computed in a similar way.

$$T_A = \begin{bmatrix} \cos\theta_A & -\sin\theta_A & h_A\cos\theta_A - v_A\sin\theta_A \\ \sin\theta_A & \cos\theta_A & h_A\sin\theta_A + v_A\cos\theta_A \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$p_{A1i} = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (2)$$

$$np_{A1i} = T_A \times p_{A1i} \quad (3)$$

After computing the new coordinates of all the points for moved stampings, we combine all the points of the candidate locator regions of the same locator number into one set because these points must be covered by one robot. For example, the new points (NP_{A1}) from $A1$, the new points (NP_{B1}) from $B1$, and the points (P_{C1}) of $C1$ are combined into one set of points. Here P_{C1} stays the same because stamping C was designated as stationary. In total, there are six sets of points for six robots. In general, M stampings will have $N + 2$ sets of points for N

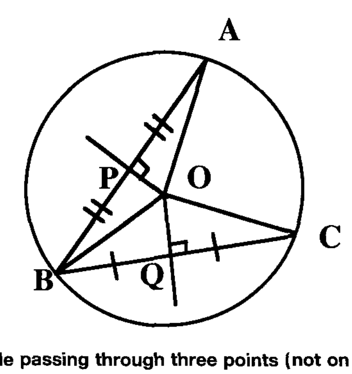


Fig. 12 Circle passing through three points (not on the same line)

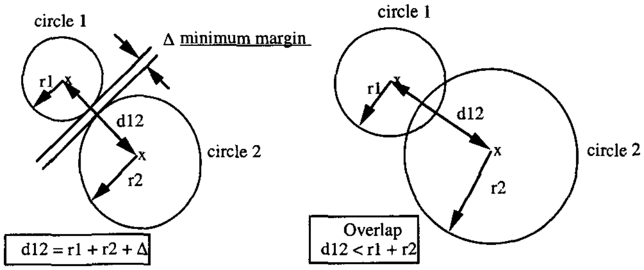


Fig. 13 Overlap between two circles

+ 2 robots if each stamping has N surface points, one hole, and one slot for locators.

Now, we move to Step 4 of Algorithm Scheme 1, finding the smallest circle that encloses all the points of the set. This smallest circle is commonly called the minimum spanning circle. The following subsection describes this step in detail.

3.2.2 Minimum Circle Enclosing All the Points of a Set. There are many published research results on the minimum spanning circle for a given set of points. These results are thoroughly reviewed by Preparata and Shamos (1988). We briefly review three of them here. First, Rademacher and Toeplitz (1957) present an algorithm that runs in $O(N^4)$. They compute all the circles defined by either two points or three points and choose the smallest circle that encloses all the points. Second, Shamos and Hoey (1975) present an algorithm that runs in $O(N \log N)$. They construct the farthest point Voronoi diagram in $O(N \log N)$ and find the two diametrical point circle or the three circumference point circle in $O(N)$. Third, Megiddo (1983) presents an algorithm that runs in $O(N)$. He transforms a quadratic minimization problem (the distance equation between the center and a circumference point is quadratic) into a linear problem and solves the linearized minimization problem employing the simplex method.

In this research, we take a variation of Rademacher and Toeplitz's algorithm because the number of points in our problem setting is small (tens of points) and it is very easy to program this algorithm. We use three steps in finding the minimum spanning circle. First, we find the convex points (Graham, 1972; O'Rourke, 1993) for a given set of points because the minimum spanning circle goes through convex points as shown in Fig. 11. In other words, we can reduce the number of search points by considering convex points only. Second, we find a circle formed by the two points as the circle diameter that are farthest among the convex polygon points. Third, we check whether or not all the convex polygon points are enclosed by this circle. If enclosed, we have found the minimum spanning circle. If not enclosed, we compute all the circles that can be defined by three points out of the convex polygon points, and choose the smallest circle that encloses all the convex polygon points.

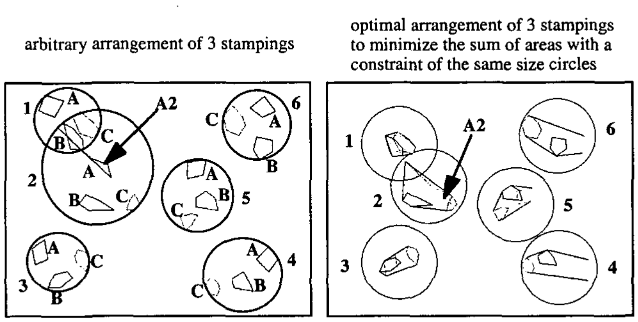


Fig. 14 Example with no feasible stamping arrangement

Data for Candidate Locator Regions
(6 fixture robots, 3 stampings)

	A		B		C	
	x	y	x	y	x	y
1	-7.8	4.6	-6.2	3.8	-5.6	4.2
	-6.8	4.2	-5.8	2.2	-5.2	3.2
	-6.2	5.2	-5.0	2.2	-4.3	2.9
	-7.4	5.6	-5.0	2.8	-4.0	3.4
			-5.6	3.5	-4.4	3.9
					-5.2	4.6
2	-4.6	0.6	-5.2	-1.2	-2.2	-1.6
	-3.2	0.4	-4.6	-1.6	-1.8	-1.6
	-3.4	1.2	-3.0	-1.6	-1.4	-1.2
	-4.4	1.6	-4.0	-1.0	-1.5	-0.9
			-4.8	-0.6	-1.7	-0.6
					-2.0	-1.0
3	-8.2	-4.0	-7.2	-6.4	-5.5	-4.4
	-8.0	-5.0	-6.4	-6.4	-5.4	-4.8
	-7.2	-4.6	-5.6	-5.8	-4.8	-5.4
	-7.4	-3.4	-6.0	-5.4	-4.6	-4.6
			-6.8	-5.6	-4.8	-4.2
					-5.3	-4.0
4	6.0	-4.6	4.6	-6.4	2.8	-6.2
	6.8	-5.4	5.8	-6.6	3.1	-6.8
	7.4	-4.6	5.6	-6.0	3.5	-6.9
	6.8	-3.8	5.2	-5.4	3.7	-6.5
			4.5	-5.8	3.6	-6.1
					3.2	-5.7
5	2.6	0.6	2.2	-1.4	1.5	-2.8
	1.6	0.4	3.6	-1.6	1.9	-2.5
	1.6	1.4	3.4	-0.8	2.1	-2.2
	2.8	1.6	2.8	-0.4	2.3	-1.9
			2.2	-0.8	1.9	-1.8
					1.2	-2.0
6	5.6	4.8	5.8	2.2	4.1	3.4
	6.4	4.2	6.2	1.6	4.9	3.1
	7.2	4.6	7.0	2.1	5.3	3.7
	6.5	5.5	7.0	2.8	5.1	4.3
			6.0	3.0	4.5	4.9
					3.9	4.3

Search Domain for
Movement Variables
(used by Grefenstette's Program)

genes: 6

gene 0
min: -5, max 5
values: 1024
format: %7.2f

gene 1
min: -5, max 5
values: 1024
format: %7.2f

gene 2
min: 0, max 6.28
values: 1024
format: %7.2f

gene 3
min: -5, max 5
values: 1024
format: %7.2f

gene 4
min: -5, max 5
values: 1024
format: %7.2f

gene 5
min: 0, max 6.28
values: 1024
format: %7.2f

Data for
Grefenstette's Program

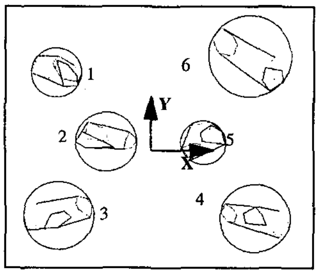
Experiment = 1
Total Trials = 5000
Population Size = 50
Structure Length = 60
Crossover Rate = 0.6
Mutation Rate = 0.001
Generation Gap = 1.0
Scaling Window = 5
Report Interval = 200
Structures Saved = 10
Max Gens w/o Eval = 2
Dump Interval = 0
Dumps Saved = 0
Options = acefL
Random Seed = 123456789
Rank Min = 0.75

Fig. 15 Example data for the flexible fixture workspace synthesis system

In our approach to finding the minimum spanning circle, it is simple to construct a circle with two points as the circle diameter. However, the construction of a circle that goes through three points requires several preliminary steps. We describe how to construct this circle using Fig. 12, where A, B, and C are the three points. First, construct a perpendicular line (PO) through the mid-point (P) of line segment AB. Second, construct a perpendicular line (QO) through the mid-point (Q) of line segment BC. Third, find the intersection point (O) between lines PO and QO. Last, construct a circle that has the center at O and the radius as the distance between point O and point A. Then, the circle also goes through points B and C. This can be easily proved from four right angle triangles OPA, OPB, OQB, and OQC. By the Pythagorean theorem, $OP^2 + PA^2 = OA^2 = OP^2 + PB^2 = OB^2 = OQ^2 + QB^2 = OQ^2 + QC^2 = OC^2$. Therefore, $OA = OB = OC$.

3.2.3 Optimal Search Formulations and Search Result. In this research we consider three cases when synthesizing flexible fixture workspaces. The first case is that we want to use the smallest possible robots and we can choose any size. The second case is that we want to use the smallest possible robots with all the robots the same size. The third case is that we want to use the smallest possible robots and we prefer a particular robot to be as small as possible. This case is for situations in which we do not have much room for a particular robot due to neighboring components on the base plate. These three cases can be interpreted as minimizing: (1) the sum of circle areas, (2) the sum of circle areas with a constraint of the same circle size, and (3) the sum of circle areas with a preference for a particular circle to be as small as possible. For all the three cases, there is a constraint of no overlap between the circles.

For Step 5 of Algorithm Scheme 1, we define three different objective functions for the three cases. First, let's represent the



- Stamping Arrangement
  B: $\Delta x = 0.46$, $\Delta y = 1.50$, $\Delta\theta = 0.06$ (rad)
  C: $\Delta x = 0.72$, $\Delta y = 1.23$, $\Delta\theta = 6.14$ (rad)
- Circle Diameter (Robot Size)
  1: D = 2.98, 2: D = 3.58, 3: D = 4.11
  4: D = 4.12, 5: D = 2.66, 6: D = 4.88
- Circle Centers
  1: xc = -6.31, yc = 4.60
  2: xc = -3.35, yc = 0.41
  3: xc = -6.15, yc = -4.11
  4: xc = 5.37, yc = -4.24
  5: xc = 2.34, yc = 0.35
  6: xc = 5.04, yc = 5.58

Fig. 16 Output of flexible fixture workspace: total workspace small

constraint of no overlap with some margin between circles as shown in Fig. 13. If we represent A_i and ri as the area and radius of the i-th circle respectively, dij as the distance between the centers of the i-th and j-th circle, and Δ as the minimally allowable margin between two circles, then we can define the overlap variable α as Eq. (4). We use α for representing the no overlap constraint as a penalty term in three objective functions: Eq. (5) for the first case, Eq. (6) for the second case, and Eq. (7) for the third case. We formulated these objective functions to use with genetic algorithms. In the optimization, a constraint is treated as a penalty term in an objective function as Eqs. (5), (6), and (7).

$$\left.\begin{array}{ll} \text{if } (ri + rj + \Delta) > dij & \text{(Overlap)} \\ \alpha = (ri + rj + \Delta) - dij & \\ \text{else} & \text{(No Overlap)} \\ \alpha = 0 & \end{array}\right\} \quad (4)$$

Case 1: Objective Function

$$= \sum_{i=1}^{N+2} A_i + \sum_{i,j=1,\, i\neq j}^{N+2} \alpha^*(A_i^* A_j) \quad (5)$$

Case 2: Objective Function

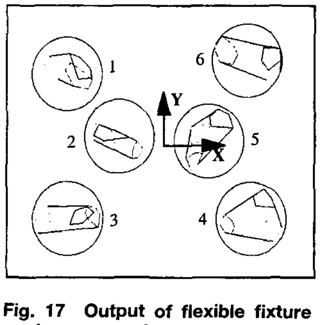
$$= (N+2)^* A_{\max} + \sum_{i,j=1,\, i\neq j}^{N+2} \alpha^*(A_i^* A_j) \quad (6)$$

Case 3: Objective Function

$$= \sum_{i=1,\, i\neq k}^{N+2} A_i + BN^* A_k + \sum_{i,j=1,\, i\neq j}^{N+2} \alpha^*(A_i^* A_j) \quad (7)$$

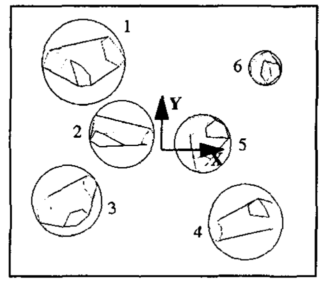
(BN indicates a Big Number to magnify the effect of k-th circle on the function.)

The genetic algorithm will give us the optimal values for stamping movement variables, circle diameters, and circle center coordinates. For example, if there are M stampings and each stamping has $N + 2$ candidate locator regions, we will have $3^*(M - 1)$ optimal values for arranging $M - 1$ stampings, $N + 2$ circle diameters, and $2^*(N + 2)$ coordinates as $N + 2$ circle centers. The optimal search result corresponds to Step 6 of Algorithm Scheme 1.



- Stamping Arrangement
  B: $\Delta x = 0.07$, $\Delta y = 1.75$, $\Delta\theta = 0.03$ (rad)
  C: $\Delta x = -0.08$, $\Delta y = 1.86$, $\Delta\theta = 0.15$ (rad)
- Circle Diameter (Robot Size)
  1to 6: D = 4.29
- Circle Centers
  1: xc = -6.13, yc = 4.73
  2: xc = -3.02, yc = 0.80
  3: xc = -6.17, yc = -4.08
  4: xc = 5.31, yc = -4.18
  5: xc = 2.78, yc = 0.66
  6: xc = 5.27, yc = 5.53

Fig. 17 Output of flexible fixture workspace synthesis system: total workspace small and all fixture robots having the same size



- Stamping Arrangement
  B: $\Delta x = 0.46$, $\Delta y = 1.50$, $\Delta\theta = 0.06$ (rad)
  C: $\Delta x = 0.72$, $\Delta y = 1.23$, $\Delta\theta = 6.14$ (rad)
- Circle Diameter (Robot Size)
  1: D = 5.23, 2: D = 4.13, 3: D = 4.42
  4: D = 4.70, 5: D = 3.57, 6: D = 2.08
- Circle Centers
  1: xc = -5.19, yc = 4.83
  2: xc = -2.72, yc = 0.40
  3: xc = -6.23, yc = -3.68
  4: xc = 5.09, yc = -5.01
  5: xc = 2.39, yc = -0.14
  6: xc = 6.29, yc = 4.48

Fig. 18 Output of flexible fixture workspace synthesis system: total workspace is small but with one particular robot workspace (#6) to be as small as possible

In some design situations, we may have to conclude that the current candidate locator regions do not allow any non-overlapping stamping arrangement. That is, there is no feasible arrangement of stampings. For example, Fig. 14 shows that there is an overlap between circle 1 and circle 2. Here the sketch on the left hand side is an arbitrary arrangement of three stampings (A, B, and C), and the sketch on the right hand side is the optimal arrangement found by a computer using Eq. (6). In this example, we can conclude that there is no feasible stamping arrangement, and we need to refine (or narrow) some candidate locator regions such as $A2$. If we use Eq. (5) instead of Eq. (6), we may be able to find a feasible arrangement. Some cases, we may have to go through several cycles of refining candidate locator regions and running a search mechanism before we find a satisfactory arrangement of stampings.

4 Example of Synthesizing Flexible Fixture Workspaces

This section illustrates how to synthesize flexible fixture workspaces with an example. There are three steps in synthesizing flexible fixture workspaces for a set of stampings using the method presented in section 3. First, we need to prepare 3 data files: (1) a file for 2D point coordinates of the candidate locator regions of all the stampings, (2) a file for the search domain as ranges of values for stamping movement variables (a gene for a variable), and (3) a file for genetic algorithms. Figure 15 shows an example of three data files in a tabular format.

This example has three stampings (A, B, and C) and each stamping has 6 candidate locator regions per stamping, where each candidate locator region is represented by a set of 2D points for the vertices of a polygon. If we designate one stamping (A) as stationary, a computer can move the remaining stampings (B and C) within the search domain. Second, we run the workspace synthesis computer program. Third, we investigate computer outputs as shown in Figs. 16, 17, and 18. Genes 0 to 2 correspond to the three movement variables of stamping B, and genes 3 to 5 correspond to the movement variables of stamping C.

5 Conclusions

In this paper, we presented a system that fixture designers can use to synthesize flexible fixture workspaces for a set of different stampings. In particular, a fixture robot workspace is represented by a circle and a candidate locator region is represented by the vertices of a polygon. An algorithm scheme is developed to find the optimal arrangement of stampings and workspace sizes and centers, and this algorithm scheme is computationally implemented by employing Grefenstette's program. The workspace synthesis system is tested with lab data and appears to be effective. Currently the system is limited to cylindrical workspaces. It looks feasible to use this system in a real industry environment. In the future, we want to generalize robot workspaces for different configurations such as PPP (Prismatic Prismatic Prismatic).

References

Abe, K., Hayashi, A., and Sakaino, M., 1995, "An Update on Nissan Intelligent Body Assembly System," *Body Assembly & Manufacturing,* Vol. 15, International Body Engineering Conference, Oct. 31–Nov. 2, Detroit, Michigan, M. Nasim Uddin, ed.

Bausch, J. J., and Youcef-Toumi, K., 1990a, "Kinematic Methods for Automated Fixture Reconfiguration Planning," *IEEE International Conference on Robotics & Automation,* pp. 1396–1401.

Cai, W. J., Hu, S. J., and Yuan, J., 1996, "Deformable Sheet Metal Fixturing: Principles, Algorithms and Simulations," ASME Journal of Manufacturing Science and Engineering, Vol. 118, pp. 318–324, August.

Campbell, P., 1994, *Basic Fixture Design,* Industrial Press Inc., New York, NY.

Davis, L., 1991, *Handbook of Genetic Algorithms,* New York, Van Nostrand Reinhold.

Drake, A., 1984, "Fixture Design: Working with Modules," *Manufacturing Engineering,* January, pp. 35–38.

Eary, F., and Reed, E., 1974, *Techniques of Pressworking Sheet Metal,* 2nd Ed., Prentice Hall.

Fanuc Robotics, 1995, "A Brochure of FLEXTOOL® Body Shop Manufacturing Solutions," FANUC Robotics North America, 2000 South Adams Road, Auburn Hills, MI 48326-2800.

Goldberg, David E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning,* Reading, Massachusetts, Addison-Wesley.

Grefenstette, J., 1991, *GENESIS,* TSP P.O. Box 991, Melrose, MA 02176 (Computer Program in C and User Manual—Public Domain for Research Purpose).

Jain, P., Fenyes, P., and Richter, R., 1992, "Optimal Blank Nesting Using Simulated Annealing," ASME *Journal of Mechanical Design,* Vol. 114, March 1992, pp. 160–165.

Lee, Jinkoo, 1996, *Set-based Design Systems for Stampings and Flexible Fixture Workspaces,* PhD Thesis, The University of Michigan, Ann Arbor, MI.

Menassa, R. J., and DeVries, W. R., 1991, "Optimization Methods Applied to Selecting Support Positions in Fixture Design," ASME Journal of Engineering for Industry, Vol. 113, pp. 412–418.

Nee, A. Y. C., Bhattacharyya, N., and Poo, A. N., 1987, "Applying AI in Jigs and Fixtures Design," *Robotics & Computer-Integrated Manufacturing,* Vol. 3, No. 2, pp. 195–200.

O'Connor, Leo, 1994, "Agile Manufacturing in a Responsive Factory," *Mechanical Engineering,* pp. 54–57, July.

O'Rourke, Joseph, 1995, *Computational Geometry in C,* Cambridge University Press.

Preparata, Franco P., and Shamos, Michael Ian, 1988, *Computational Geometry,* Springer-Verlag.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 1992, *Numerical Recipes in C,* 2nd Ed., Cambridge University Press.

Rearick, M. R., Hu, S. J., and Wu, S. M., 1993, "Optimal Fixture Design for Deformable Sheet Metal Workpieces," *Transactions of NAMRI/SME,* pp. 407–412.

Sekine, Y., Koyama, S., and Imazu, H., 1991, "Nissan's New Production System: Intelligent Body Assembly System," *SAE Papers 910816.*

van Laarhoven, P., and Aarts, E., 1987, *Simulated Annealing: Theory and Applications,* D. Reidel Publishing Co.

Ward, A., Liker, J., Sobek, D., and Cristiano, J., 1995, "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster," *Sloan Management Review,* Spring, Vol. 36, No. 3, pp. 43–61.

Youcef-Toumi, K., Bausch, J. J., and Blacker, S. J., 1989, "Automated Setup and Reconfiguration for Modular Fixturing," *Robotics & Computer-Integrated Manufacturing,* Vol. 5, No. 4, pp. 357–370.

Youcef-Toumi, K., Liu, W. S., and Asada, H., 1988, "Computer-Aided Analysis of Reconfigurable Fixtures and Sheet Metal Parts for Robotic Drilling," *Robotics & Computer-Integrated Manufacturing,* Vol. 4, No. 3/4, pp. 387–393.

Zhang, X., and Roy, U., 1993, "Criteria for Establishing Datums in Manufactured Parts," *Journal of Manufacturing Systems,* Vol. 12, No. 1, pp. 36–50.