CHANGEABILITY PLANNING OF MODULAR FIXTURES IN A ROBOTIC ASSEMBLY SYSTEM TO MANAGE PRODUCT VARIETY

A THESIS

SUBMITTED TO THE FACULTY OF

UNIVERSITY OF MINNESOTA

BY

Hossein Tohidi

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF SCIENCE IN ENGINEERING MANAGEMENT

Advisor: Dr. Tarek AlGeddawy

July 2016

Acknowledgements

I would like to take this opportunity to acknowledge all who have helped, assisted and supported me in the completion of this thesis.

I would like to primarily thank my adviser Dr. Tarek AlGeddawy for all his precious guidance and support in this research over the past two years. I would also like to thank my thesis examining committee, Dr. Hongyi Chen and Dr. Wenqing Zhang for their time and and insightful comments and recommendations.

I would like to acknowledge the Chancellor's Fellowship I received from the UMD Graduate School QMA fund at the University of Minnesota Duluth. I would also like to thank Dr. Tarek AlGeddawy and Dr. Hongyi Chen who have supported me in getting this fellowship.

I am very grateful to my parents for their endless support and love. I am also grateful to all my friends and fellow students in the MIE department for their friendship and support.

Also I would like to thank my wonderful wife, Shima Hosseinpour, and dedicate my thesis to her for all her support and love.

Abstract

Changeability in Manufacturing Systems has been implemented in various industries over the last forty years to meet the challenge of change in product demand and design. Changeability is an umbrella paradigm and consists of different manufacturing system characteristics, such as changeover-ability, reconfigurability, flexibility, transformability and agility. On the system level, flexibility is enabled particularly by implementing modular equipment designs such as modular fixtures that can hold a variety of product geometries. On the operational level, optimizing the changeability plan of those modular fixtures improves the performance of the manufacturing system.

This thesis considers a hole-pattern modular fixture to increase the changeability of an automated assembly system. In this assembly setting, a robot that is located on top of a conveyor belt loop places different parts on the modular fixture and secure them by inserting four pegs around each part. The more peg replacements are occurred, the longer fixture setup time is required. Hence, the researchers of this study have developed five different mathematical models to effectively determine the best parts and pegs locations on the fixture to minimize the number of pegs replacements considering different conditions and constraints. At this point, researchers presented a new fixture design that improves the fixture modularity to hold more products with different geometries. By making a few modifications, the five previous models have been extended to be used for this new fixture design.

Subsequently, researchers proposed three case studies with different parameters sizes to investigate the models' efficiency. The results show that the different models are able to significantly reduce fixture setup time.

Table of Contents:

List of Tables:

List of Figures:

Chapter 1. Introduction

In today's world, introducing more product variety gives a competitive advantage to manufacturers to meet different market segment demands and customer requirements. This competition is intensified by the emergence of new materials and technologies which enables manufacturers to produce new products and services or make new economically justifiable products. Those changes and requirements are usually associated with different and new product features which leads to a wide variety of product designs and part geometries [1].

However, the proliferation of product variants increases manufacturing costs, planning complexity, and managerial burdens. Therefore, managing product variety is challenging and requires attention to the whole production cycle. Designing and fabricating fixtures are among the main aspects that should be considered since they constitute about 10 to 20 percent of the total manufacturing costs [2]–[4].

Designing and fabricating different fixtures are among the main barriers for manufacturing systems to produce a product variety with different part geometries. This is due to the fact that new produced parts geometries may require designing a set of corresponding fixture to handle those parts. Designing and fabricating fixtures is usually performed using expensive and time consuming processes for tight tolerances, and manufacturers need storage space for the unused fixtures. To overcome the time and costs associated with the frequent changes in fixture design, modular fixtures have been developed. The changeability plan of these fixtures is very important for changeable manufacturing systems, especially in automated systems in which robots are in charge of placing and securing parts in their respective places. The focus of this thesis is to develop changeability plans for modular fixtures to improve their application and efficiency in robotic assembly and minimize their changeover time.

In this study, 10 x 10 modular fixtures are used in a mid-volume mid-variety production system in order to hold a variety of geometrically different parts in a robotic assembly system. As shown in Figure 1, cradles are installed on top of a set of pallets in a "Hold'n Go" conveyor-belt loop moving through an automated assembly system.

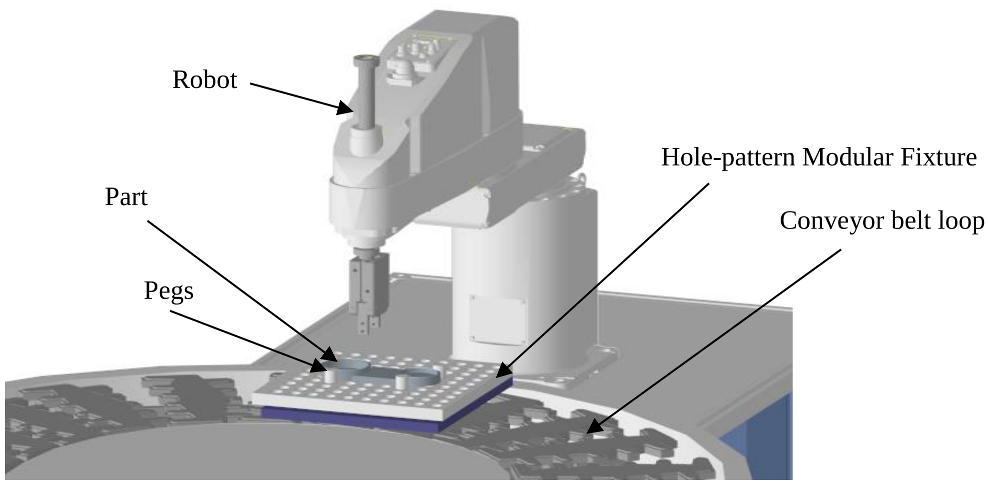


Figure 1. Robotic assembly station on Hold'n Go conveyor system robot

The modular cradle has a hole-pattern on its adapter plate for pegs to be inserted. The designed fixture is shown in Figure 2.

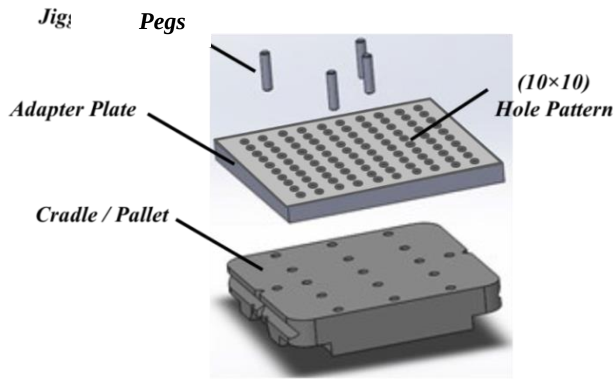


Figure 2. Proposed hole-pattern modular fixture

The proposed fixture brings universality to the assembly line because pegs can be easily re-arranged in order to fix a wide variety of part geometries. This provids sufficient support reactions for parts and prevent slippage and shaking during assembly processes. For example, a rectangular shape is fixed by inserting four pegs (Figure 3). The four pegs secure the part in place for the worst case impact force and moment conditions at any direction.

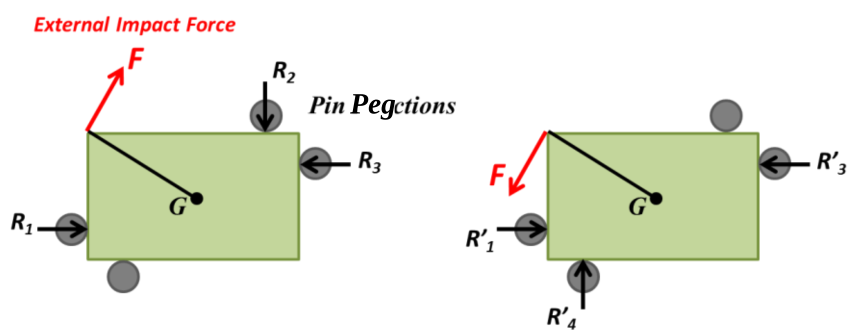


Figure 3. Rectangular part secured with 4 pegs

This is a universal system that can adapt to any part shape and geometry by rearranging the pegs. The external forces on an irregular shape are demonstrated in Figure 4.

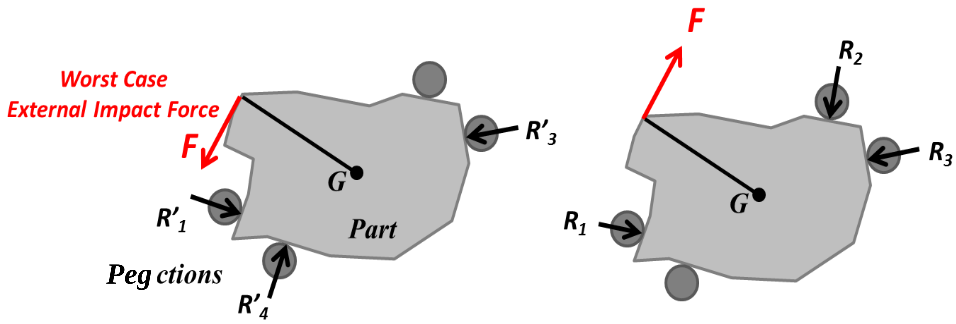


Figure 4. Irregular part secured with 4 pegs

However, a larger number of replaced pegs requires a longer setup time for fixtures. This negatively impacts manufacturing costs. A single peg replacement takes 1 second to pick up the peg, 1 second to reach the new location, and 3 seconds to insert the peg in its new location. Therefore, it takes 5 seconds for each peg to rearrange which means 20 seconds for each part to be fixed on the cradle. This setup is done in series with other assembly stations in an automated balanced line. In this thesis, it is assumed that a 1 second production interruption will cost 16.66 cents based on producing 30 mixed-geometry parts per hour which has a profit of $20 each. As a result, reducing fixture preparation time by 50% can save up to $1.666 per part.

In this research, five different mathematical models have been progressively developed to optimize the performance of the designed assembly system applying different objectives and constraints to reduce the fixture setup time. Then a new fixture design has been introduced, which requires consequent changes in the models. These models will be presented in the next section.

This thesis is organized into six chapters. The remainder of this thesis is as follows:

In Chapter 2, a literature survey is presented to highlight the best practices in changeable manufacturing systems and fixtures. The research methodology is discussed in Chapter 3. This section presents several mathematical models that optimize the overall performance of an automated changeable manufacturing system to assemble a variety of products. These models are developed progressively to enable the assembly system, considering different conditions and constraints. In Chapter 4, three different numerical examples are presented to show how these models work. These examples are randomly generated in small, medium and large sizes to investigate how effective the models are when the problem size increases. In Chapter 5, the results for running the proposed models are presented. Then several comparisons are made to assess the efficiency and effectiveness of the models. Finally, Chapter 6 contains concluding remarks and future research directions.

Chapter 2: Literature Survey

Nowadays companies face various challenges due to globalization. To penetrate new markets and adjust to rapid changes in customers' demands, manufacturing firms need to have the ability to produce a variety of products in an efficient manner. In the last few decades, reconfigurable manufacturing concept has been developed to respond to the needs for producing a variety of products [5]. As Maler et al. defined, reconfigurability is "the ability to repeatedly change capacity and functionality in a cost efficient way, in order to meet different demand situations in terms of variation in volume as well as in product characteristics" [6]. Customization, convertibility, scalability, modularity, integrability and diagnosability are the main reconfigurable strategies [6].

To achieve a reconfigurable system, reconfigurability should be applied at all different levels of manufacturing systems such as planning and control, system design, and machine design [7]. Chaube et al. classified different requirements of a reconfigurable manufacturing system into layout, machine tools, and planning [8].

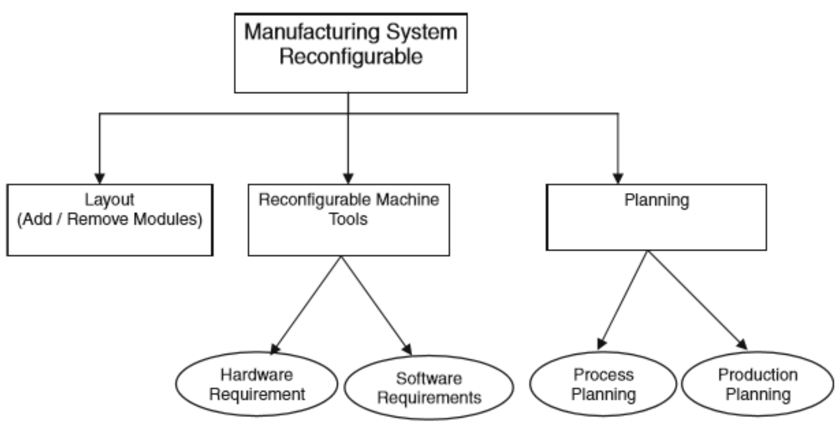


Figure 5. Hierarchical structure of reconfigurable manufacturing systems [8]

As shown in Figure 5, planning and designing reconfigurable machining tools are two main aspects of a reconfigurable system. Hence, this study is going to represent a new intelligent planning system that can improve the applicability of a flexible fixture in an assembly line. The main drawbacks of flexible manufacturing systems are their higher expenses and lower production rates compared to dedicated manufacturing systems [7]. Therefore the presented research has been defined to minimize system setup time and to consequently maximize the production rate.

Fixtures are fundamental components of almost all manufacturing operations. During different operations, fixtures hold parts from machining to assembly to avoid shaking or slippage during these processes. The design of fixtures determine the quality and costs of final products [4], [9]. Most of the fixtures which are widely used in different industries are designed for specific purposes and any changes in the predefined parameters such as part's geometries might not be properly handled by those fixtures [10]. As a result, firms need to design new fixtures that can hold new parts. Increasing the number of products or constantly changing in the parts' geometries increases the manufacturers' need for designing new fixtures. Designing, installing, and calibrating different fixtures costs a lot and also increase the production time significantly.

Regardless of the challenges that manufacturing different types of products may cause for a company, producing a single product to meet all the customers' needs is no longer possible. As a result, in the last few decades, there has been a steady growth in the number of companies that tended to produce variety of products [11]. Various customers' needs in different markets, rapid technology changeovers, price competition, and price discrimination are of the main reasons for companies to offer a range of products [11].

As mentioned earlier, designing and fabricating different fixtures in a high-variety production system constitute a significant portion of manufacturing costs. In addition, in today's competitive world, companies are looking for winning more market shares by providing a variety of products for their customers. Therefore, many studies have been conducted on designing flexible fixtures that can hold a variety of parts to improve the integrity, intelligence, and flexibility of manufacturing firms [12].

To reduce costs associated with fixture, many researchers work on designing flexible fixtures that can hold different parts' geometries, and as a result, reduce the required number of fixtures and increase sustainability of manufacturing systems [13]–[16]. Kang and Feng believe that designing flexible fixtures can reduce 80% of fixture costs[16].

Vasundara and Padmanaban categorized different types of flexible fixtures, namely two-dimensional, three-dimensional, V-blocks and peg-type fixture system[17]. Fixtures can also be classified as modular, reconfigurable, sensory-based, phase change, adaptable fixtures, and as programmable conformable clamps. Among these, modular fixtures are highly important ones, and they are widely used in different industries [15-18]. In Figure 6, two types of modular fixtures are depicted.

Vacuum fixtures are another type of flexible fixture that is widely used in different industries. As shown in Figure 7, Hu proposed a sensory-based intelligent vacuum fixture that can be used for machining operations and also facilitate the part loading process [12].

a) b)

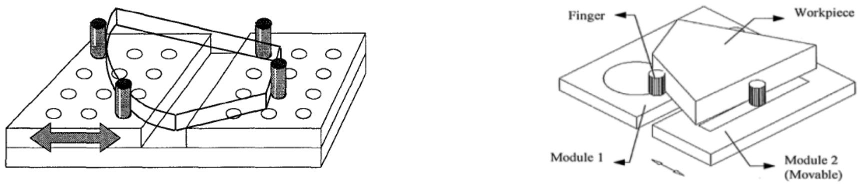


Figure 6. a) Hole pattern modular fixture with movable jaws [14] , b) Three-fingered modular fixture [18]

a) b)

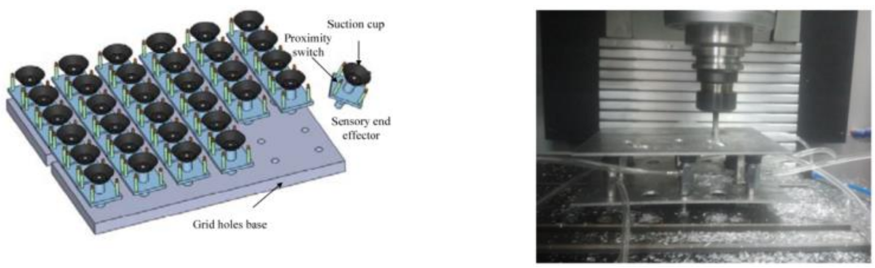


Figure 7. Sensory-based intelligent vacuum fixture [12]

Shirinzade introduced different flexible fixture strategies [19]. Among these are reconfigurable fixtures and programmable clamps as shown in Figure 8.

a) b)

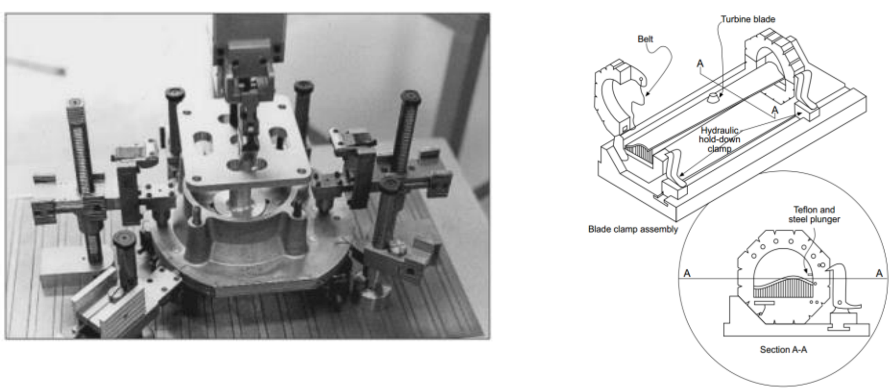
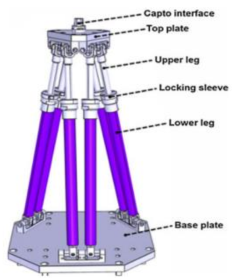


Figure 8. a) Reconfigurable fixture, b) Programmable clamps [19]

In another study, Jonsson and Ossbahr introduced a reconfigurable fixture (Figure 9.a). They also discuss different approaches to using this reconfigurable fixture [9]. Xiong et al. present a different reconfigurable fixture that is suitable for large sheets of metal (Figure 9.b). They present a Genetic Algorithm to determine the number of supports and their locations [20].

a)



b)

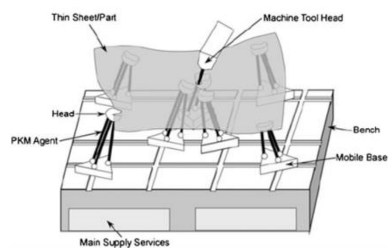


Figure 9. Reconfigurable fixtures

Modular fixtures provide production sites with higher changeability by enabling them to secure irregular shapes (see Figure 10).

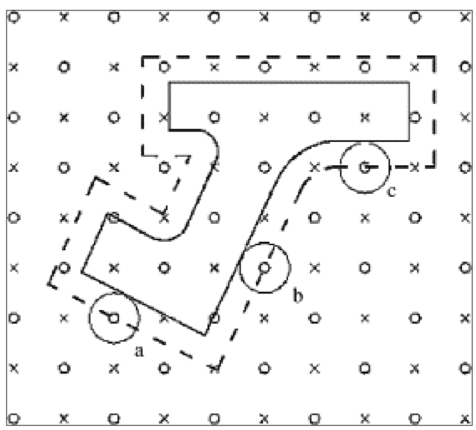


Figure 10. Irregular part securing with modular fixture using locators [21]

They also reduce production lead time by easily adapting to various parts with different shapes and sizes [16]. It should be noted that designing modular fixtures needs significant of knowledge and experience, and it is a very complex task which is usually performed by CAD-based tools [22]. This includes setup and fixture planning, fixture configuration design, and fixture design verification [23]. Enumerating all fixture plans is impossible, so a computer-based program is needed to find the optimal fixture plan [21].

Chapter 3: Methodology

To minimize the time and effort associated with designing and applying different fixtures in a make to order automated assembly system, a modular fixture is designed. A robot is placed on top of a conveyor loop to place different part on a cradle and fixing them by inserting pegs. The part and the pegs locations are determined by mathematical models which have been developed to minimize peg replacements.

The models have been developed progressively considering different constraints with the same objective of minimizing the fixture setup time. Hence, in this section, different mathematical formulations will be presented to illustrate how they can help the system to achieve ths goal.

3.1 Basic Model

As mentioned earlier, the problem is about finding the best possible places for the pegs to be inserted in a universal modular cradle in order to minimize the number of peg replacements. This concept is depicted in Figure 11 for more clarification.

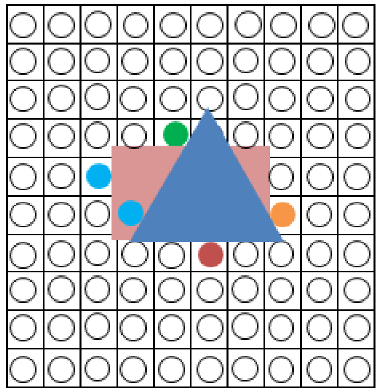


Figure 11. 10' x 10 hole pattern modular fixture securing two parts successively

As shown in Figure 11, two different parts with different parts are fixed one after another on a hole-pattern modular fixture. Each part is fixed using four pegs and as depicted, three pegs are not changed between these two successive parts. Note that it is assumed in this paper that the possible peg locations for each part are predetermined by checking the force closure equations. Therefore, finding the best locations among these possible locations is considered. The Mixed Integer Programming (MIP) model is given using the following notations:

Indices:

i	part number ($i = 1,2,\dots,I$)
w, w'	peg number ($w = 1,2,\dots,W$)
l	l'th location for a peg($l = 1,2,\dots,L$))

Parameters:

px_{iwl}	the first coordinate (X) of pegs' possible locations for part i, peg w and l'th place
py_{iwl}	the second coordinate (Y) of pegs' possible locations for part i, peg w and l'th place
M'	a big number(1000)

Variables:

M_{iwl} Binary variable. $M_{iwl} = 1$ only if the corresponding coordinates are selected. Elsewhere $M_{iwl} = 0$.

$\mathrm{n}_{\mathrm{iw}w'}$ Binary variable. $\mathrm{n}_{\mathrm{iw}w'} = 1$ only if the corresponding peg does not change for two different successive parts. elsewhere $\mathrm{n}_{\mathrm{iw}w'} = 0$

x_{iwl} the first coordinate (X) of the chosen location

y_{iwl} the second coordinate (Y) of the chosen location

x'_{iw} the first coordinate (X) of the chosen location for i'th part and w'th peg

y'_{iw} the second coordinate (Y) of the chosen location for i'th part and w'th peg

$\Delta\mathrm{x}_{\mathrm{iw}w'}$ horizontal distance between two peg locations for two successive parts, part i and (i +1).

$\Delta\mathrm{y}_{\mathrm{iw}w'}$ vertical distance between two peg locations for two successive parts, part i and (i +1).

To clarify different indexes, they are shown in the Figure 12 for a simple problem. In this example w and l are assumed to be 4 and 2 respectively. It means that 4 different places must be chosen for pegs to be inserted and each peg has 2 different possible locations.

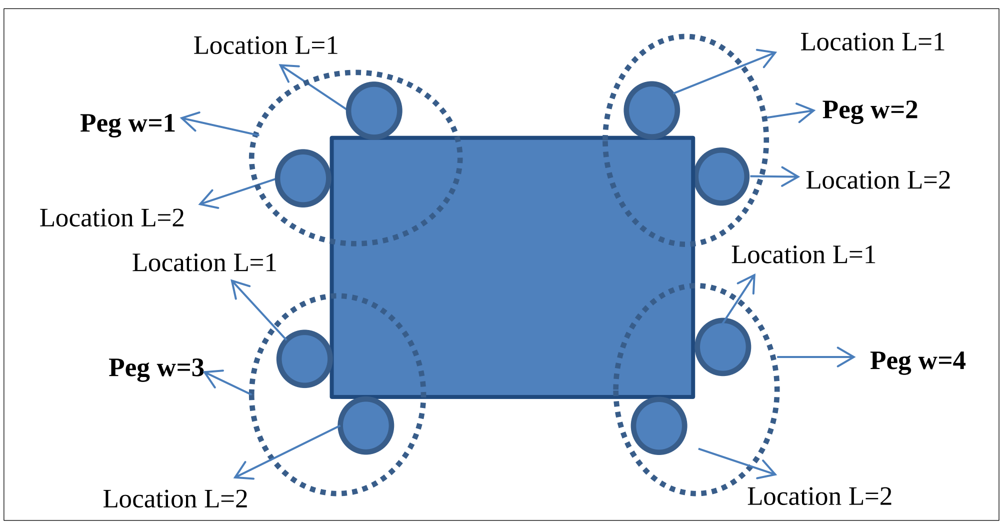


Figure 12. Indices

To show how big the problem is, the total number of combinations is computed. One of each possible locations (l) for each peg (w) must be chosen for inserting peg, $\binom{l}{1}^{w}$. To fix i different parts, the total number of possible locations is:

$$\binom{l}{1}^{\mathrm{wi}} \tag{3-1}$$

Based on equation 1, the number of possible combinations for a small size problem with 5 part, 4 pegs and 2 possible locations for each peg is:

$$\binom{l}{1}^{\mathrm{wi}} = 2^{20} = 1{,}048{,}576 \tag{3-2}$$

Note that the problem's size can be grown exponentially by increasing parameters' value. For instance the number of possible combinations for 10 different parts, with the same value of w and l will be equal to:

$$\binom{l}{1}^{\mathrm{wi}} = 2^{40} \sim 1{,}099{,}511{,}627{,}776 \tag{3-3}$$

Therefore, it is important for the system to notice the solution time to be reasonable and applicable for industries.

As mentioned before, in this study it is assumed that the possible sets for peg locations are predetermined in a way that provides sufficient contact with parts and prevent parts' slippage during assembling process. The possible set for each part is given in two separate matrixes px_{iwl} and py_{iwl} for X and Y coordinates.

3.1.1. Mathematical Formulation

In this section the developed mathematical model is presented:

$$obj = \max(\sum_{i}\sum_{w}\sum_{w'} \mathrm{n}_{\mathrm{i}\mathrm{w}w'}) \tag{3-4}$$

Subject to:

$$\sum_{l} M_{iwl} = 1 \qquad \forall\, i\,, w \tag{3-5}$$

$$\mathrm{x}_{iwl} = \mathrm{px}_{iwl} * M_{iwl} \qquad \forall\,\mathrm{i}\,,\mathrm{w},\mathrm{l} \tag{3-6}$$

$$\mathrm{y}_{iwl} = \mathrm{py}_{iwl} * M_{iwl} \qquad \forall\,\mathrm{i}\,,\mathrm{w},\mathrm{l} \tag{3-7}$$

$$x'_{iw} = \sum_{l} \mathrm{x}_{iwl} \qquad \forall\, i\,, w \tag{3-8}$$

$$y'_{iw} = \sum_{l} \mathrm{y}_{iwl} \qquad \forall\, i\,, w \tag{3-9}$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'} \geq x'_{iw} - x'_{(i+1)\,w'} \qquad \forall\, i, w, w' \tag{3-10}$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'} \geq x'_{(i+1)w'} - x'_{iw} \qquad \forall\, i, w, w' \tag{3-11}$$

$$\Delta\mathrm{y}_{\mathrm{iw}w'} \geq y'_{iw} - y'_{(i+1)\,w'} \qquad \forall\, i, w, w' \tag{3-12}$$

$$\Delta\mathrm{y}_{\mathrm{iw}w'} \geq y'_{(i+1)w'} - y'_{iw} \qquad \forall\, i, w, w' \tag{3-13}$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'} + \Delta\mathrm{y}_{\mathrm{iw}w'} \leq M' * (1 - \mathrm{n}_{\mathrm{iw}w'}) \qquad \forall\, i, w, w' \tag{3-14}$$

$$M_{iwl}, \mathrm{n}_{\mathrm{iw}w'} \in \{0,1\} \tag{3-15}$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'}, \Delta\mathrm{y}_{\mathrm{iw}w'} \geq 0 \tag{3-16}$$

$$\mathrm{x}_{iwl}, \mathrm{y}_{iwl}, x'_{iw}, x'_{iw} \in int \tag{3-17}$$

The first equation is the objective function which is maximizing the number of fixed pegs between any two successive parts. Note that in this paper instead of minimizing number of peg replacement, the number of fixed pegs are maximized.

Equation 3-5 states that the model must choose one of the possible locations for each peg. $M_{iwl} = 1$ if the corresponding location is chosen and $M_{iwl} = 0$ elsewhere. Equations 3-6 to 3-9 compute pegs' X and Y coordinates for each part (${x'}_{iw}$ and ${y'}_{iw}$) and equations 3-10 to 3-14 calculate number of fixed pegs between two successive parts.

Note that equations 3-10 and 3-11 are separated from the following constraint:

$$\Delta \mathrm{x}_{\mathrm{iw}w'} > \left| {x'}_{iw} - {x'}_{(i+1)\,w'} \right| \tag{3-18}$$

Because $\Delta \mathrm{x}_{\mathrm{iw}w'}$ is a positive variable, if two points ${x'}_{iw}$ and ${x'}_{(i+1)\,w'}$ don't match with each other, then $\Delta \mathrm{x}_{\mathrm{iw}w'} > 0$. Same for equation 3-12 and 3-13, if ${y'}_{iw}$ and ${y'}_{(i+1)\,w'}$ don't match with each other, $\Delta \mathrm{y}_{\mathrm{iw}w'} > 0$. If two points completely match with each other (same X and Y coordinates) then $\Delta \mathrm{x}_{\mathrm{iw}w'}$ and $\Delta \mathrm{y}_{\mathrm{iw}w'}$ could get the value 0 or more than 0. Equation 3-14 forces $\mathrm{n}_{\mathrm{iw}w'}$ to get value 0 if the distance between two points is more than 0. Put in other words, if X and Y coordinates for two points don't match with each other, variable $\mathrm{n}_{\mathrm{iw}w'}$ is forced to get a 0 value. Otherwise it can get value 0 or more than 0. Since $\mathrm{n}_{\mathrm{iw}w'}$ is in the objective function, $\mathrm{n}_{\mathrm{iw}w'}$ will get 1 when there is no force to get value 0.

3.2 NLRTE Model[1]

To improve the Basic Model and cut off the fixtures' setup time, a new model has been developed to enable the system to rotate and translate parts on the cradle. Hence, the system has more flexibility to position the parts on the cradle and fix it with inserting pegs. Therefore, the result is significantly improved. Rotation and translation of a part on the cradle is depicted in Figure 13.

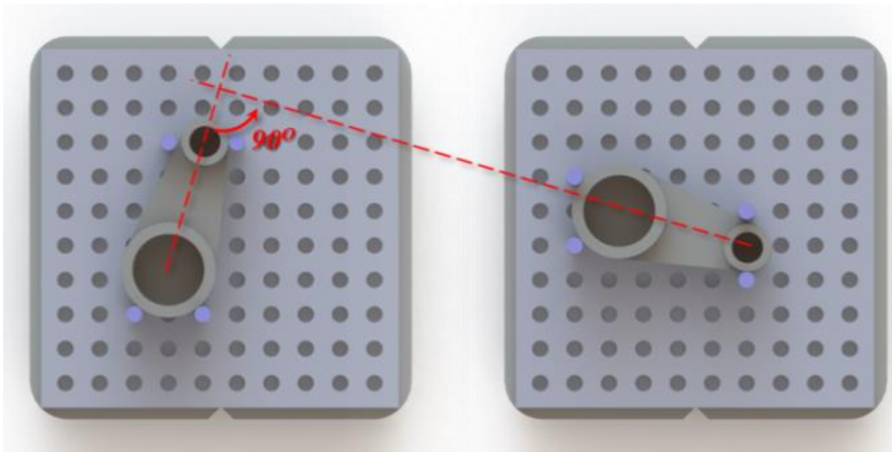


Figure 13. 90° part rotation

3.2.1. Mathematical Model

The mixed integer non-linear programming (MINLP) model is developed using the following notations:

Index:

i	part number $(i = 1,2, \ldots , I)$
w, w'	peg number $(w = 1,2, \ldots , W)$
l	l' location for a peg$(l = 1,2, \ldots , L)$

[1] Non-linear Rotation and translation enabled Model

Parameters:

px_{iwl}	the first coordinate (X) of pegs' possible locations for part i, peg w and l'th place
py_{iwl}	the second coordinate (Y) of pegs' possible locations for part i, peg w and l'th place
M'	a big number(1000)

Variables:

M_{iwl}	Binary variable. $M_{iwl} = 1$ only if the corresponding coordinates are selected. Elsewhere $M_{iwl} = 0$.
$\mathrm{n}_{\mathrm{i}ww'}$	Binary variable. $\mathrm{n}_{\mathrm{i}ww'} = 1$ only if the corresponding peg does not change for two different successive parts. elsewhere $\mathrm{n}_{\mathrm{i}ww'} = 0$
x_{iwl}	the first coordinate (X) of the chosen location
y_{iwl}	the second coordinate (Y) of the chosen location
x'_{iw}	the first coordinate (X) of the chosen location for i'th part and w'th peg
y'_{iw}	the second coordinate (Y) of the chosen location for i'th part and w'th peg
$\Delta\mathrm{x}_{\mathrm{i}ww'}$	horizontal distance between two peg locations for two successive parts, part i and (i +1).
$\Delta\mathrm{y}_{\mathrm{i}ww'}$	vertical distance between two peg locations for two successive parts, part i and (i +1).
nr_{i}	The translation of part i to right
nl_{i}	The translation of part i to left
nu_{i}	The translation of part i to up
nd_{i}	The translation of part i to down
θ_{i}	The rotation coefficient of part i around the origin point (0,0)

In this section the developed mathematical model is presented:

$$obj = \max\left(\sum_{i}\sum_{w}\sum_{w'} \mathrm{n}_{\mathrm{i}ww'}\right) \tag{3-19}$$

Subject to:

$$\sum_l M_{iwl} = 1 \qquad \forall\, i\,, w \qquad (3\text{-}20)$$

$$\mathrm{x}_{iwl} = \mathrm{px}_{iwl} * M_{iwl} \qquad \forall\, \mathrm{i}\,, \mathrm{w}, \mathrm{l} \qquad (3\text{-}21)$$

$$\mathrm{y}_{iwl} = \mathrm{py}_{iwl} * M_{iwl} \qquad \forall\, \mathrm{i}\,, \mathrm{w}, \mathrm{l} \qquad (3\text{-}22)$$

$$x'_{iw} = \sum_l \mathrm{x}_{iwl} * \cos\left(\theta_\mathrm{i} * \frac{\pi}{2}\right) - \sum_l \mathrm{y}_{iwl} * \sin(\left(\theta_\mathrm{i} * \frac{\pi}{2}\right) + \mathrm{nr_i} - \mathrm{nl_i} \qquad \forall\, i\,, w \qquad (3\text{-}23)$$

$$y'_{iw} = \sum_l \mathrm{x}_{iwl} * \sin\left(\theta_\mathrm{i} * \frac{\pi}{2}\right) + \sum_l \mathrm{y}_{iwl} * \cos(\left(\theta_\mathrm{i} * \frac{\pi}{2}\right) + \mathrm{nu_i} - \mathrm{nd_i} \qquad \forall\, i\,, w \qquad (3\text{-}24)$$

$$0 \le x'_{iw} \le 10 \qquad \forall\, i, w \qquad (3\text{-}25)$$

$$0 \le y'_{iw} \le 10 \qquad \forall\, i, w \qquad (3\text{-}26)$$

$$0 \le \theta_\mathrm{i} < 4 \qquad \forall\, i \qquad (3\text{-}27)$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'} \ge x'_{iw} - x'_{(i+1)\,w'} \qquad \forall\, i, w, w' \qquad (3\text{-}28)$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'} \ge x'_{(i+1)w'} - x'_{iw} \qquad \forall\, i, w, w' \qquad (3\text{-}29)$$

$$\Delta\mathrm{y}_{\mathrm{iw}w'} \ge y'_{iw} - y'_{(i+1)\,w'} \qquad \forall\, i, w, w' \qquad (3\text{-}30)$$

$$\Delta\mathrm{y}_{\mathrm{iw}w'} \ge y'_{(i+1)w'} - y'_{iw} \qquad \forall\, i, w, w' \qquad (3\text{-}31)$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'} + \Delta\mathrm{y}_{\mathrm{iw}w'} \le M' * (1 - \mathrm{n}_{\mathrm{iw}w'}) \qquad \forall\, i, w, w' \qquad (3\text{-}32)$$

$$M_{iwl}, \mathrm{n}_{\mathrm{iw}w'} \in \{0,1\} \qquad (3\text{-}33)$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'}, \Delta\mathrm{y}_{\mathrm{iw}w'} \ge 0 \qquad (3\text{-}34)$$

$$\mathrm{x}_{iwl}, \mathrm{y}_{iwl}, x'_{iw}, x'_{iw}, nl_i, nr_i, nd_i, nu_i, \theta_\mathrm{i} \in int \qquad (3\text{-}35)$$

Similar to the Basic Model, the objective function is defined as maximizing the number of fixed pegs between any two successive parts. Equation 3-20 states that the model must choose one of the possible locations for each peg. $\mathrm{M_{iwl}} = 1$ if the corresponding location is chosen and $\mathrm{M_{iwl}} = 0$ elsewhere. Equations 3-21 to 3-24 calculate the peg locations on

XY axis. It should be considered that equations 3-23 and 3-24 are the key equations which fulfil the parts' translation and rotation. Note that to rotate a point (x_1, y_1) around the origin with α degree, the following formula is used.

$$x_2 = x_1 \cos(\alpha) - y_1 \sin(\alpha) \qquad (3\text{-}36)$$

$$y_2 = x_1 \sin(\alpha) + y_1 \cos(\alpha) \qquad (3\text{-}37)$$

Where x_1 and y_1 represent the initial coordinates, and x_2 and y_2 are the new coordinates after α degree rotation. In this model, four different rotation angles $(0, \frac{\pi}{2}, \pi, \frac{3\pi}{2})$ are considered, because the other angles cause different challenges in the assembly line and also cause infeasible solutions. To clarify the rotation and translation equation, look at the following example.

A rectangle with 4 coordinates, (2,1), (2,3), (6,1) and (6,3) is assumed. In this case, the rectangle rotates $\alpha = \frac{\pi}{2}$ around the origin, translates 3 units up $(\mathrm{nu} = 3)$ and 8 points right $(\mathrm{nr} = 8)$. The results are shown in Figure 14. It should be noted that after the rotation and without any translation, the result is an infeasible solution because the rectangle is out of 10 x10 feasible regions.

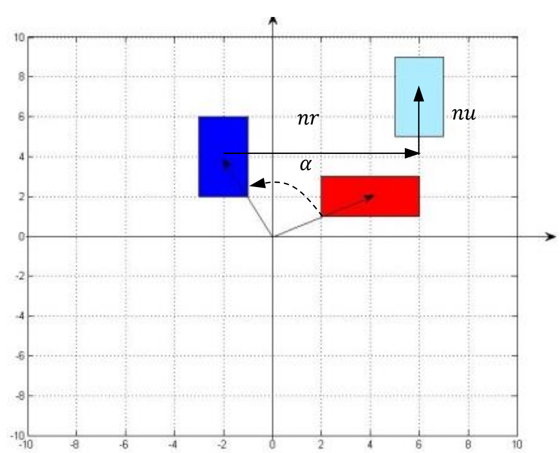


Figure 14. Part rotation and translation on the cradle

Equations 3-25 and 3-26 state that the final X and Y coordinate must be in the feasible region. Equation 3-27 states that the rotation coefficient must be chosen from this set: {0,1,2,3}. Equations 3-28 to 3-32 calculate number of fixed pegs between two successive parts. Equations 3-28 and 3-29 are separated from the following constraint:

$$\Delta x_{iww'} > \left| x'_{iw} - x'_{(i+1)\,w'} \right| \qquad (3\text{-}38)$$

Because $\Delta x_{iww'}$ is a positive variable, if two points x'_{iw} and $x'_{(i+1)\,w'}$ don't match with each other, then $\Delta x_{iww'} > 0$. The same is true for equation 3-30 and 3-31; if y'_{iw} and $y'_{(i+1)\,w'}$ don't match with each other, then $\Delta y_{iww'} > 0$. If two points completely match each other (same X and Y) then $\Delta x_{iww'}$ and $\Delta y_{iww'}$ could get the value 0 or more than 0. On the other hand, equation 3-32 forces $n_{iww'}$ to get value 0 if the distance between two points is more than 0. In other words, if x and y coordinates for two points don't match each other, variable $n_{iww'}$ is forced to get a 0 value. Otherwise it may get value 0 or more than 0. Because $n_{iww'}$ is in the objective function with the maximization sign, $n_{iww'}$ gets 1 when there is no force to get value 0.

3.3 LRTE Model[2]

The non-linearity nature of the NLRTE model, make the solution time to increase dramatically by increasing in the problem's parameters. Therefore, the model is not able to converge to optimal solution in a timely manner and it means that the model is not effectively applicable to a large size problem. To overcome to this problem, another model

[2] Linear rotation translation enabled model

is developed using GAMS and MATLAB to make the mathematical model linearized. This model is called LRTE model hereafter.

LRTE considers all the possible part translations and rotations on the cradle in order to determine the best part positions and the best location for inserting pegs. For instance, in Figure 15, three successive part are holding on a modular cradle one after another.

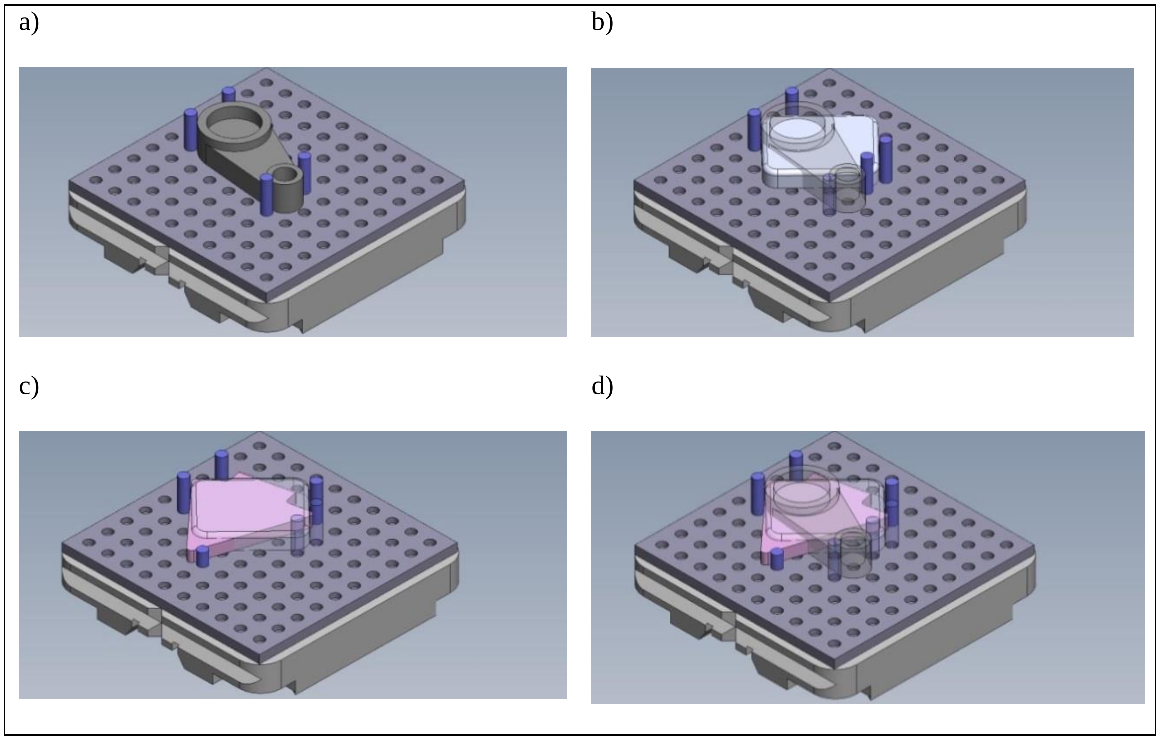


Figure 15. Hole-pattern modular fixture hold three parts successively

As shown in this figure, the first part was fixed on the fixture by choosing 4 locations among possible set of peg location. In section b, the second part was fixed on the cradle by replacing 1 peg. Note that the first part and the replaced pegs are shown in a transparent way to illustrate the replacement process. The same thing for part c, the third part was fixed by rearrangement of two pegs. As a result three part are fixing on cradle but only three pegs’ replacements occurred. The top view of the cradle is depicted in Figure 16.

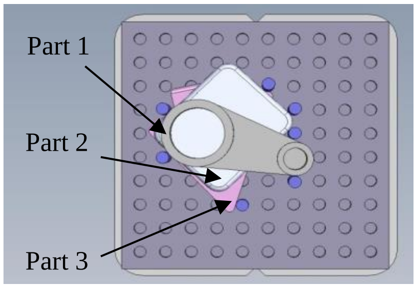


Figure 16. Fixture holding three parts - top view

As shown in Figure 15, three different parts with different geometries are fixed one after another on a hole-pattern modular fixture. Each part is fixed using four -pegs and as depicted, some pegs are not replaced between these successive parts. Therefore, 7 pegs were needed in total and 3 pegs’ replacement occurred to fix these different part and as mentioned before the less pegs’ rearrangements that are made, the less fixtures preparation time that is needed.

3.3.1. LRTE[3] Mathematical Model

As mentioned before, the NLRTE Model was non-linear. The nonlinearity nature of mathematical model presented in [24] increases the solution time and reduces its efficiency. To overcome this problem and improve the applicability of proposed modular fixture, a mixed integer linear programming model is developed. In this paper, the proposed model will be referred by LRTE model which stands for Linear Rotation and Translation Enabled Model. Before describing the model, different indices, parameters, and decision variables are introduced.

[3] Linear Rotation and Translation Enabled

Indices:

i	part's number ($i = 1,2, \dots, I$)
w, w'	peg's number ($w = 1,2, \dots, W$)
l	l'th location for a peg($l = 1,2, \dots, L$)
θ	θ'th rotation of a part (θ=1,2,..,4)

Parameters:

$\mathrm{px}_{iwl\theta}$	the first coordinate (X) of pegs' possible locations for part i, peg w and l'th place
$\mathrm{py}_{iwl\theta}$	the second coordinate (Y) of pegs' possible locations for part i, peg w and l'th place
M'	a big number(1000)

Variables:

M_{iwl}	Binary variable. $M_{iwl} = 1$ only if the corresponding coordinates are selected. Elsewhere $M_{iwl} = 0$.
$\mathrm{n}_{\mathrm{i}ww'}$	Binary variable. $\mathrm{n}_{\mathrm{i}ww'} = 1$ only if the corresponding peg does not change for two different successive parts. elsewhere $\mathrm{n}_{\mathrm{i}ww'} = 0$
$\mathrm{n'}_{\mathrm{i}\theta}$	Binary variable. $\mathrm{n'}_{\mathrm{i}\theta} = 1$ only if the θ'th rotation of part i is selected.
$\mathrm{x}_{iwl\theta}$	The first coordinate (X) of the chosen location
$\mathrm{y}_{iwl\theta}$	The second coordinate (Y) of the chosen location
x'_{iw}	The first coordinate (X) of the chosen location for i'th part and w'th peg
y'_{iw}	The second coordinate (Y) of the chosen location for i'th part and w'th peg
$\Delta\mathrm{x}_{\mathrm{i}ww'}$	Horizontal distance between two peg locations for two successive parts, part i and (i +1).
$\Delta\mathrm{y}_{\mathrm{i}ww'}$	Vertical distance between two peg locations for two successive parts, part i and (i +1).
$\mathrm{nr_i}$	The translation of part i to right
$\mathrm{nl_i}$	The translation of part i to left
$\mathrm{nu_i}$	The translation of part i to up
$\mathrm{nd_i}$	The translation of part i to down

According to the problem statement, objective function and constraints are formulated below:

$$\max Z = \sum_i \sum_w \sum_{w'} \mathrm{n}_{\mathrm{i}ww'} \tag{3-39}$$

$$\sum_l M_{iwl\theta} = \mathrm{n'}_{\mathrm{i}\theta} \qquad \forall\, i\,, w, \theta \tag{3-40}$$

$$\sum_w \sum_l M_{iwl\theta} = W * \mathrm{n'}_{\mathrm{i}\theta} \qquad \forall\, i\,, \theta \tag{3-41}$$

$$\sum_w \sum_l \sum_\theta M_{iwl\theta} = W \qquad \forall\, i \tag{3-42}$$

$$\mathrm{x}_{iwl\theta} = \mathrm{px}_{iwl\theta} * M_{iwl\theta} \qquad \forall\, \mathrm{i}\,, \mathrm{w}, \mathrm{l}, \theta \tag{3-43}$$

$$\mathrm{y}_{iwl\theta} = \mathrm{py}_{iwl\theta} * M_{iwl\theta} \qquad \forall\, \mathrm{i}\,, \mathrm{w}, \mathrm{l}, \theta \tag{3-44}$$

$$x'_{iw} = \sum_\theta \sum_l \mathrm{x}_{iwl\theta} + \mathrm{nr_i} - \mathrm{nl_i} \qquad \forall\, i\,, w \tag{3-45}$$

$$y'_{iw} = \sum_\theta \sum_l \mathrm{y}_{iwl\theta} + \mathrm{nu_i} - \mathrm{nd_i} \qquad \forall\, i\,, w \tag{3-46}$$

$$0 \le x'_{iw} \le 10 \qquad \forall\, i, w \tag{3-47}$$

$$0 \le y'_{iw} \le 10 \qquad \forall\, i, w \tag{3-48}$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'} \ge x'_{iw} - x'_{(i+1)\,w'} \qquad \forall\, i, w, w' \tag{3-49}$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'} \ge x'_{(i+1)w'} - x'_{iw} \qquad \forall\, i, w, w' \tag{3-50}$$

$$\Delta\mathrm{y}_{\mathrm{iw}w'} \ge y'_{iw} - y'_{(i+1)\,w'} \qquad \forall\, i, w, w' \tag{3-51}$$

$$\Delta\mathrm{y}_{\mathrm{iw}w'} \ge y'_{(i+1)w'} - y'_{iw} \qquad \forall\, i, w, w' \tag{3-52}$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'} + \Delta\mathrm{y}_{\mathrm{iw}w'} \le M' * (1 - \mathrm{n}_{\mathrm{iw}w'}) \qquad \forall\, i, w, w' \tag{3-53}$$

$$M_{iwl\theta}, \mathrm{n}_{\mathrm{iw}w'}, \mathrm{n'}_{\mathrm{i}\theta} \in \{0,1\} \tag{3-54}$$

$$\Delta\mathrm{x}_{\mathrm{iw}w'}, \Delta\mathrm{y}_{\mathrm{iw}w'} \ge 0 \tag{3-55}$$

$$\mathrm{x}_{iwl\theta}, \mathrm{y}_{iwl\theta}, x'_{iw}, y'_{iw}, nl_i, nr_i, nd_i, nu_i \in int \tag{3-56}$$

In comparison with the NLRTE model rotation angle is added as a new index to LRTE model. To calculate the new possible sets of peg locations after rotation a separate MATLAB program is developed. This program rotates different part and the peg locations with 4 different angles, $\theta = (0, \frac{\pi}{2}, \pi, \frac{3\pi}{2})$, separately and calculates a new larger set of peg location. To rotate the part with θ degree, the same formula which were presented in NLRTE model (Equation 3-36 and 3-37) are used. It should be noticed that the peg locations must be integer values between 1 and 10 representing the 10 x 10 hole-pattern fixture. Therefore, to avoid infeasible solutions, only 4 rotation angles $\left(0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right)$ are considered. On the other hand, after rotating a part, the pegs will be out of the 10 x 10 feasible region. Therefore a corrective function is developed in MATLAB to avoid infeasible solutions. For instance, in Figure 17, a triangular part (Triangle No. 1) is rotated 180 degrees around the origin (0,0). But, the result (Triangle No.2) is not a feasible solution because it is not within the 10 x 10 fixture's area. The corrective function translates the shape in order to bring it back into the feasible region (Triangle No.3).

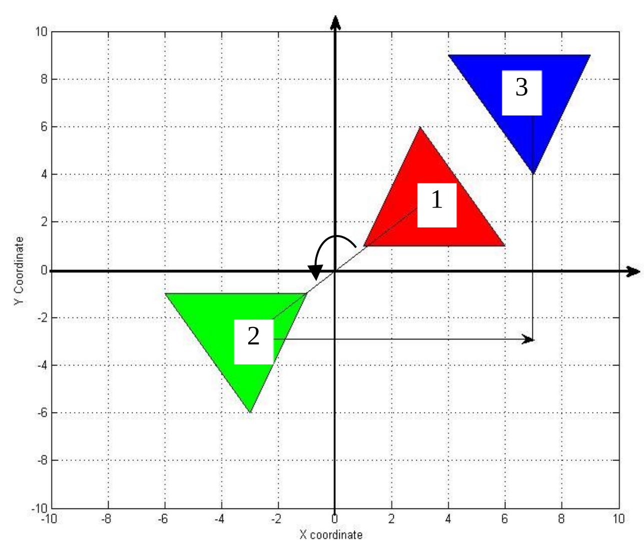


Figure 17. Corrective function after rotation

After finding the feasible set of peg location, they are imported to GAMS. Then LRTE model will select the best peg locations out of a set of candidate places in order to minimize the pegs' rearrangements during the planning horizon.

Equation 1 states the model's objective function which is maximizing total number of fixed pegs between any two successive parts. It should be noted that this objective is equivalent with minimizing the total number of pegs' rearrangements.

Equations 3-40 to 3-42 are key constraints, stating that the model must choose $W = 4$ pegs from possible sets of peg location, considering that all of the peg locations must belong to a specific rotation angle. For instance, the model cannot choose one location from possible peg locations with π degree rotation and another one with 0 degree rotation. Variable $\mathrm{n'}_{\mathrm{i}\theta}$ determines which rotation degree is chosen and $M_{iwl\theta}$ determines which pegs location are chosen within the possible set with that specific rotation angle. It should be mentioned that the equation 3-42 can also be stated as below:

$$\sum_{\theta} n'_{i\theta} = W \qquad \forall\, i \qquad (3\text{-}57)$$

Equations 3-43 to 3-46 calculate the X and Y coordinates for each peg based on $M_{iwl\theta}$. Note that equations 3-45 and 3-46 capable the model to translate the part on the cradle. Equation 3-47 and 3-48 define the 10 x 10 feasible regions and equations 3-49 to 3-53 determine if pegs rearrangement happened or not. Note that equations 3-49 and 3-50 are split from the following constraint:

$$\Delta \mathrm{x}_{\mathrm{i}ww'} > \left| x'_{iw} - x'_{(i+1)\,w'} \right| \qquad (3\text{-}58)$$

Because $\Delta \mathrm{x}_{\mathrm{i}ww'}$ is a positive variable, if two points x'_{iw} and $x'_{(i+1)\,w'}$ don't match each other, then $\Delta \mathrm{x}_{\mathrm{i}ww'} > 0$. The same is true for equations 3-51 and 3-52, if y'_{iw} and $y'_{(i+1)\,w'}$ don't match each other then $\Delta \mathrm{y}_{\mathrm{i}ww'} > 0$. If two points completely match (same X and Y) then $\Delta x_{iww'}$ and $\Delta y_{iww'}$ could get the value 0 or more than 0. On the other hand, equation 3-53 forces $n_{iww'}$ to get value 0 if the distance between two points is more than 0. In other words, if x and y coordinates for two points don't match each other, variable $n_{iww'}$ is forced to be 0. Otherwise it may get value greater than or equal to 0. Because $n_{iww'}$ is in the objective function with the maximization sign, $n_{iww'}$ gets 1 when there is no force to get value 0.

3.4 SLRTE Model[4]

LRTE model was so effective in finding the best parts' position on the cradle and determining the best places for pegs to be inserted. But in that model, it was assumed that the parts come in order. In other words the job orders are predetermined and the system does not have the flexibility to change it. However, in real situation it is not always the case and the system can change the job orders to optimize its performance.

Hence, a new model is developed to give the system the flexibility to change the job sequences in order to minimize the fixture setup time. It should be noticed that translation and rotation are part of this model as well.

3.4.1. Mathematical Model

Before describing the model, different indices, parameters, and decision variables are introduced.

[4] Scheduling enabled LRTE model

Indices:

i, i'	part number ($i = 1,2,\dots,I$)
w, w'	peg number ($w = 1,2,\dots,W$)
l	l'th location for a peg($l = 1,2,\dots,L$)
θ	θ'th rotation of a part (θ=1,2,..,4)

Parameters:

$\mathrm{px}_{iwl\theta}$	the first coordinate (X) of pegs' possible locations for part i, peg w and l'th place
$\mathrm{py}_{iwl\theta}$	the second coordinate (Y) of pegs' possible locations for part i, peg w and l'th place
M'	a big number(1000)

Variables:

$M_{ii'wl\theta}$	Binary variable. $M_{ii'wl\theta} = 1$ only if the corresponding coordinates are selected. Zero elsewhere.
$\mathrm{n}_{\mathrm{iw}w'}$	Binary variable. $\mathrm{n}_{\mathrm{iw}w'} = 1$ only if the corresponding peg does not change for two different successive parts. elsewhere $\mathrm{n}_{\mathrm{iw}w'} = 0$
$\mathrm{n'}_{\mathrm{ii}'\theta}$	Binary variable. $\mathrm{n'}_{\mathrm{ii}'\theta} = 1$ only if the job i is done $i'th$ in the jobs' order and θ'th degree of rotation is selected.
$\mathrm{x}_{iwl\theta}$	the first coordinate (X) of the chosen location
$\mathrm{y}_{iwl\theta}$	the second coordinate (Y) of the chosen location
x'_{iw}	the first coordinate (X) of the chosen location for i'th part and w'th peg
y'_{iw}	the second coordinate (Y) of the chosen location for i'th part and w'th peg
$\Delta\mathrm{x}_{\mathrm{iw}w'}$	horizontal distance between two peg locations for two successive parts, part i and (i +1).
$\Delta\mathrm{y}_{\mathrm{iw}w'}$	vertical distance between two peg locations for two successive parts, part i and (i +1).
$\mathrm{nr_i}$	The translation of part i to right
$\mathrm{nl_i}$	The translation of part i to left

nu_i	The translation of part i to up
nd_i	The translation of part i to down

According to the problem statement, objective function and constraints are formulated below:

$$\max Z = \sum_i \sum_w \sum_{w'} n_{iww'} \qquad (3\text{-}59)$$

$$n'_{ii'\theta} = \sum_w \sum_l M_{ii'wl\theta} /4 \qquad \forall\, i\,, i', \theta \qquad (3\text{-}60)$$

$$\sum_{i'} \sum_\theta n'_{ii'\theta} = 1 \qquad \forall\, i \qquad (3\text{-}61)$$

$$\sum_i \sum_\theta n'_{ii'\theta} = 1 \qquad \forall\, i' \qquad (3\text{-}62)$$

$$\sum_{i'} \sum_l \sum_\theta M_{ii'wl\theta} = 1 \qquad \forall\, i\,, w \qquad (3\text{-}63)$$

$$\sum_w \sum_\theta \sum_l \sum_{i'} M_{ii'wl\theta} = 4 \qquad \forall\, i \qquad (3\text{-}64)$$

$$x_{i'wl\theta} = \sum_i (px_{iwl\theta} * M_{ii'wl\theta}) \qquad \forall\, i', w, l, \theta \qquad (3\text{-}65)$$

$$y_{i'wl\theta} = \sum_i (py_{iwl\theta} * M_{ii'wl\theta}) \qquad \forall\, i', w, l, \theta \qquad (3\text{-}66)$$

$$x'_{iw} = \sum_\theta \sum_l x_{iwl\theta} + nr_i - nl_i \qquad \forall\, i\,, w \qquad (3\text{-}67)$$

$$y'_{iw} = \sum_\theta \sum_l y_{iwl\theta} + nu_i - nd_i \qquad \forall\, i\,, w \qquad (3\text{-}68)$$

$$1 \le x'_{iw} \le 10 \qquad \forall\, i, w \qquad (3\text{-}69)$$

$$1 \le y'_{iw} \le 10 \qquad \forall\, i, w \qquad (3\text{-}70)$$

$$\Delta x_{iww'} \ge x'_{iw} - x'_{(i+1)\,w'} \qquad \forall\, i, w, w' \qquad (3\text{-}71)$$

$$\Delta x_{iww'} \ge x'_{(i+1)w'} - x'_{iw} \qquad \forall\, i, w, w' \qquad (3\text{-}72)$$

$$\Delta y_{iww'} \ge y'_{iw} - y'_{(i+1)\,w'} \qquad \forall\, i, w, w' \qquad (3\text{-}73)$$

$$\Delta y_{iww'} \ge y'_{(i+1)w'} - y'_{iw} \qquad \forall\, i, w, w' \qquad (3\text{-}74)$$

$$\Delta x_{iww'} + \Delta y_{iww'} \le M' * (1 - n_{iww'}) \qquad \forall\, i, w, w' \qquad (3\text{-}75)$$

$$M_{ii'wl\theta}, n_{iww'}, n'_{ii'\theta} \in \{0,1\} \qquad (3\text{-}76)$$

$$\Delta x_{iww'}, \Delta y_{iww'} \ge 0 \qquad (3\text{-}77)$$

$$x_{i'wl\theta}, y_{i'wl\theta}, x'_{iw}, y'_{iw}, nl_i, nr_i, nd_i, nu_i \in int \qquad (3\text{-}78)$$

Similar to the other models that have been developed so far, the SLRTE Model's objective function is maximizing number of fixed pegs between any pair of parts. However in this model i' is added as a new index representing the job sequences. The binary variable $M_{ii'wl\theta}$ is a key variable here which get value of 1 when job i is done i'th in the jobs' sequence and for peg w, the lth location with θ degree rotation.

Equations 3-60 to 3-64 take the following logical constraints into consideration:

1) Each job has to be done only once.
2) Each position in job sequence must be filled out with only one job.
3) For each job, 4 pegs must be chosen from the possible set of pegs locations
4) All pegs for each job must be selected from one degree rotation.

Constraints 3-65 and 3-68 calculate the final peg locations by translating parts on the cradle. The next two constraints ensure that the final peg locations are within the 10 x 10 inch feasible region. Constraints 3-71 to 3-75 calculate the number of fixed pegs between any pair of jobs. These constraints are fully described in LRTE model explanation.

3.5 RTPO Model[5]

LRTE model enable the system to determine the best number of pegs' replacement by taking different part rotation and translation into consideration. In this section a mathematical model is developed to not only determine the best peg locations between two successive parts, but also figure out the optimized route that the robot should go through to replace the pegs. In other words, the problem is about minimizing the number of pegs that need to be replaced and the best travel path to replace pegs. These two objectives can translate into robot vertical and horizontal movement and the minimum movement that the robot need to make, the less energy it will consume. In order to formulate this problem, two proposed objectives are separated as a dynamic programing problem to reduce the complexity of the model and keep it efficient for real industrial cases.

Hence, two separate mathematical formulations are developed to find out the optimum vertical and horizontal movement plans accordingly. In the first step the LRTE model is used to determine the parts' position on the cradle as well as the best peg locations in order to minimize number of pegs that need to be replaced. Subsequently the results are used in the second model (RTPOM) to figure out the minimum travel distance between different peg location. The dynamic of the problem formulation is mapped in Figure 18.

[5] Robot Travel Path Optimization Model

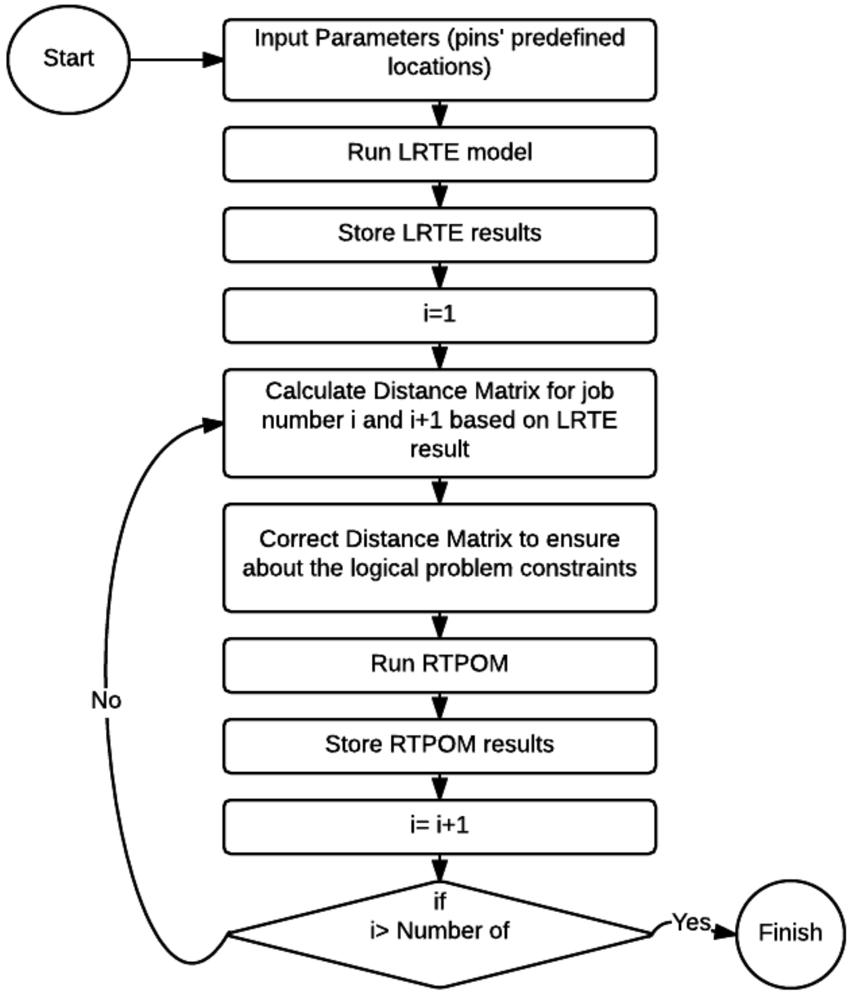


Figure 18. RTPOM approach – Schematic approach

3.5.1. Robot Travel Path Optimization Model (RTPOM)

In this section a set of x-y coordinates are given based on the previous step result. The problem is about finding the best path between pegs in order to minimize the total travel distance. To illustrate the problem, let's take a look at the following figure.

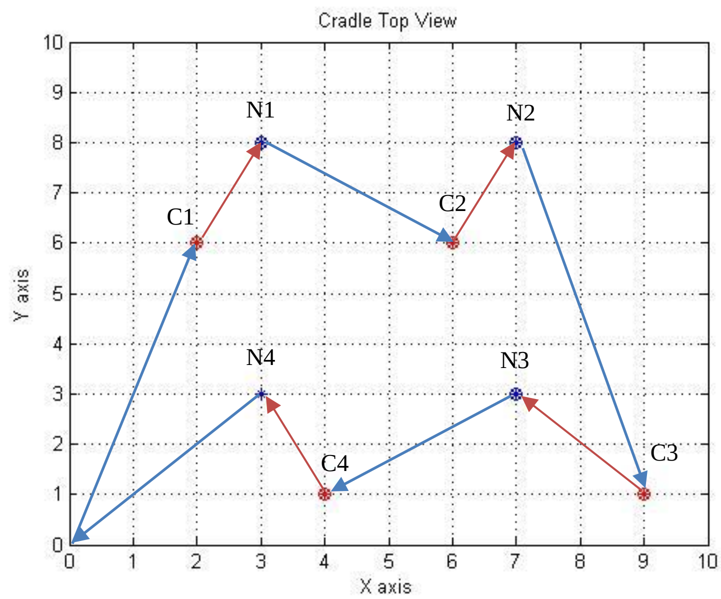


Figure 19. Possible travel path between current and future peg locations

In this figure, 8 different coordinates are given in two different sets, the current (C) and the next (N) locations. The feasible routes are the one which starts from the origin point (0,0), visit one of the current locations in order to pick up a peg, then visit one of the new peg locations in order to insert the peg and so on so forth to the last pegs to be replaced. Finally, the robot will go back to the origin points to complete its closed tour for first two part. As implied, the problem statement is a kind of the well-known problem, Travel Salesman Problem (TSP), which has been discussed in the literature for many years. However there is a difference between this problem and a common TSP model. In this problem, robot need to visit different sets of locations every other one. Because as described, the robot must take a peg out from a current location and insert it in its new location. Therefore the robot is not free to choose all locations among not-visited locations.

TSP models are usually defined by two set of information: the number of cities to visit, and distance between each pair of cities. In this paper, number of cities and their locations is pre-defined by LRTE model. Then, the results send to another program that is developed in MATLAB to calculate the distance matrix for any two successive part. To ensure the problem's logical constraints such as avoiding visiting two points from one set in a row, distance between them are defined as a big number, M, which is set to 1000 in RTPOM. As a result the distance matrix for set of 8 locations is defined as below:

<table>
<tr><th></th><th>O</th><th>S1</th><th>S2</th><th>S3</th><th>S4</th><th>F1</th><th>F2</th><th>F3</th><th>F4</th></tr>
<tr><td>O</td><td>M</td><td>D_{12}</td><td>D_{13}</td><td>D_{14}</td><td>D_{15}</td><td colspan="4">M</td></tr>
<tr><td>S1</td><td rowspan="4">M</td><td rowspan="4" colspan="4">M</td><td>D_{26}</td><td>D_{27}</td><td>D_{28}</td><td>D_{29}</td></tr>
<tr><td>S2</td><td>D_{36}</td><td>D_{37}</td><td>D_{38}</td><td>D_{49}</td></tr>
<tr><td>S3</td><td>D_{46}</td><td>D_{47}</td><td>D_{48}</td><td>D_{59}</td></tr>
<tr><td>S4</td><td>D_{56}</td><td>D_{57}</td><td>D_{58}</td><td>D_{69}</td></tr>
<tr><td>F1</td><td>D_{61}</td><td>D_{62}</td><td>D_{63}</td><td>D_{64}</td><td>D_{65}</td><td rowspan="4" colspan="4">M</td></tr>
<tr><td>F2</td><td>D_{71}</td><td>D_{72}</td><td>D_{73}</td><td>D_{74}</td><td>D_{75}</td></tr>
<tr><td>F3</td><td>D_{81}</td><td>D_{82}</td><td>D_{83}</td><td>D_{84}</td><td>D_{85}</td></tr>
<tr><td>F4</td><td>D_{91}</td><td>D_{92}</td><td>D_{93}</td><td>D_{94}</td><td>D_{95}</td></tr>
</table>

Figure 20. Distance matrix

where $D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. Note that the matrix is symmetrical meaning that the distance from point i to j is the same as distance from j to i, $D_{ij} = D_{ji}$. As mentioned before, half of the matrix is filled out with a big number, M to ensure the logical constraints of the problem. For instance after leaving the origin point (0,0), the robot must visit any points from the set of current peg locations in order to take the peg out. That is to say, visiting any point from the new set of pegs locations is logically wrong. Therefore the distance between origin point and the set of new peg locations are set as M. Also after picking up a peg from the current set of peg locations the robot must visit a location from the new set of locations to insert the peg. So it does not make sense to visit a location from the same set again. As a result the related matrix elements are all filled with M.

The matrix size is not fixed and it is a function of the number of pegs that should be replaced between each pair of part, $n_{ii'}$, and it is calculated with the following equation:

$$\text{Distance matrix size between part } i \text{ and } i' = 2 * n_{ii'} + 1 \qquad (3\text{-}79)$$

Traveling salesman problem can be solved as a mixed integer linear programming. In this section the proposed model by Miller-Tucker-Zemlin is used to solve the problem [25]. The mathematical formulation is given below:

$$Objective\ function: \sum_{i}\sum_{i'}(D_{ii'} * X_{ii'}) \qquad (3\text{-}80)$$

$$\sum_{i'} X_{ii'} = 1 \qquad \forall i \qquad (3\text{-}81)$$

$$\sum_{i} X_{ii'} = 1 \qquad \forall i' \qquad (3\text{-}82)$$

$$U_i \leq n \qquad \forall\, i \mid i \neq 1 \qquad (3\text{-}83)$$

$$U_i \geq 2 \qquad \forall\, i \mid i \neq 1 \qquad (3\text{-}84)$$

$$U_1 = 1 \qquad (3\text{-}85)$$

$$U_i - U_{i'} \leq (n-1) * (1 - X_{ii'}) \qquad \forall\, i, i' \mid i \neq 1, i' \neq 1 \qquad (3\text{-}86)$$

where the objective function is minimizing the robot total travel distance and $D_{ii'}$is the distance between two points i and i'. In this formulation, $X_{ii'}$ is a binary variable that is equal to one if the corresponding route between points i and i' is selected. Based on Miller et al. formulation, extra variable U_i is defined. This variable prevents subtours to be created. Figure 20 present an infeasible solution with subtours.

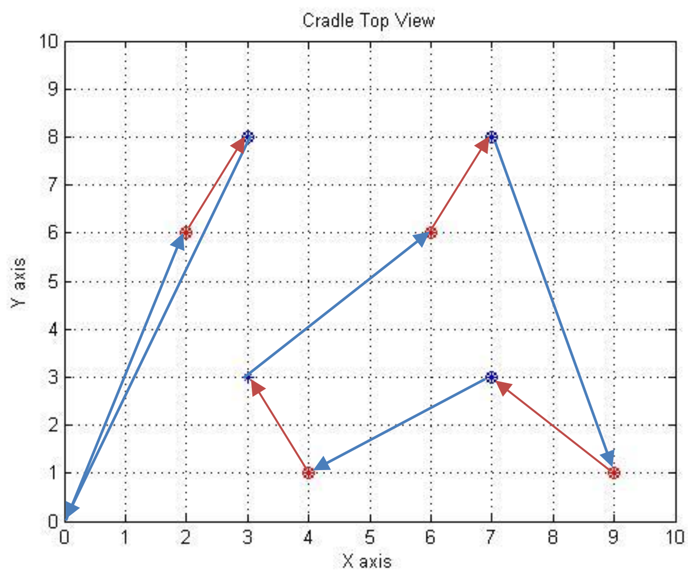


Figure 21. Sub-tours to be prevented

3.6 New Fixture Design

In this study, five different models have been presented so far that could increase the efficiency of the designed 10 x 10 hole-pattern fixture in an automated assembly system, considering different objectives and constraints. In this section, a new fixture design is introduced that can improve the system performance by being able to adapt to more parts' geometries. In previous design, fixture sometime was unable to hold the part securely as inserting pegs in any of holes cannot grab the part hard enough. This problem is shown in the following figure.

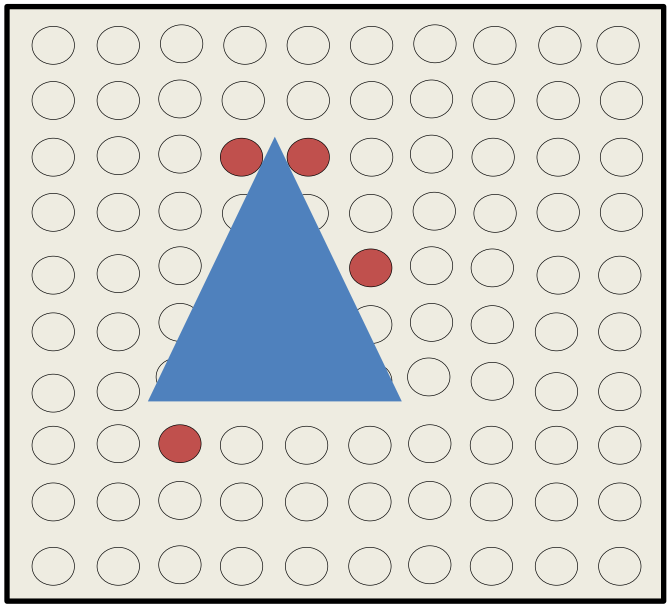


Figure 22. Fixture unable to secure the part securely

Due to the given part's geometries, the fixture is unable to keep the part securely. As depicted, two pegs could not touch the part and it means the part will move freely by importing any forces in those directions. To overcome to that problem and increase the changeability of the proposed assembly system, a new fixture is designed. In this fixture, number of holes is increased in the given 10 x 10 surface area with the same size of pegs' diameter. The isometric and top view of the designed fixture is given in Figure 23.

a) b)

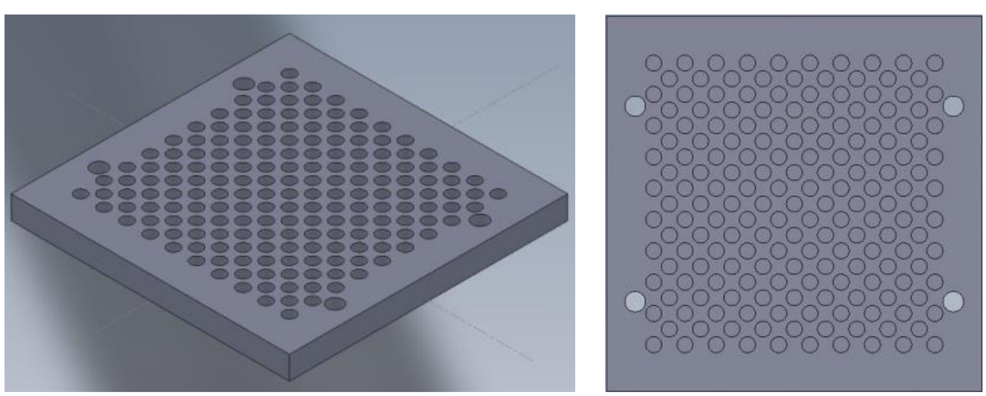


Figure 23. Newly designed fixture

Having holes within previous rows and columns, this fixture can increase the system's capability to hold more variety of products in the assembly system.

Planning the robot on the new fixture design is the next step in this study. The mathematical models that have been developed so far is based on previous design and to be able to apply them for the new fixture design, some constraints need to be modified and one constraint should be added.

Let's take a look at Figure 24. As depicted, imaginary holes (circles with dash line) are added to the basic design in order to make it similar to the previous fixture. The number of rows and columns in this figure can be calculated using following formula:

$$n' = 2 * n - 1 \qquad (3\text{-}87)$$

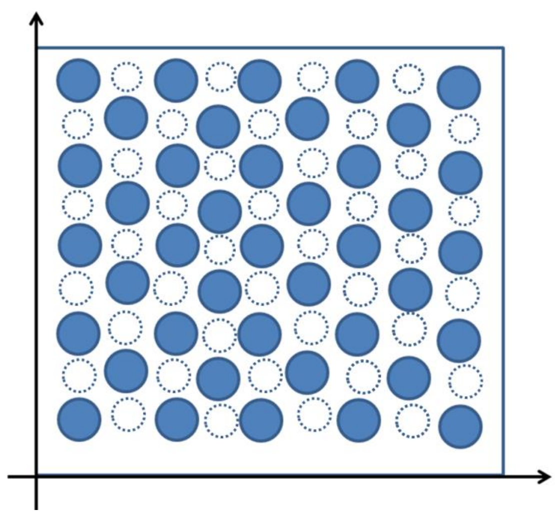


Figure 24. New fixture design with adding imaginary holes

Where n and n' are number of rows in the previous and current fixture design respectively. Based on the formula, in a 5X5 fixture which is shown in Figure 24, 9 rows and columns exist. Using the same formula, in a 10 x 10 fixture that was used in previous sections, 19 rows and columns exist. This is the first set of constraints that needs to be modified in the mathematical formulation. For instance, in the SLRTE model, equation 3-69 and 3-70 need to be modified as below:

$$1 \leq x'_{iw} \leq 19 \qquad \forall\, i, w \qquad (3\text{-}88)$$

$$1 \leq y'_{iw} \leq 19 \qquad \forall\, i, w \qquad (3\text{-}89)$$

On the other hand, a new constraint is needed to prevent the model from selecting the imaginary holes. This constraint is given in equation (3-90).

$$\text{Row No.} + \text{Column No.} = 2 * n$$

where n is a positive integer variable. By using the proposed approach all five mathematical models that have been developed can be extended to use for the new fixture design.

Chapter 4. Numerical Examples

To illustrate the application of the developed models in real case situations and to compare their efficiency in terms of the optimum solution and solution time, three different numerical examples have been generated and solved separately. These examples were randomly generated in different sizes to investigate the efficiency of the proposed models by increasing the problem size. It should be noted that the proposed models are run based on the basic 10 x 10 fixture design.

4.1 Case Study 1

The first example is about a small-size problem with 5 part to be planned on a 10 x 10 fixture. In this example, among the pre-determined possible set of pegs locations 4 locations must be selected. The problem's parameters are given in Tables 1 to 3.

Table 1. Indices for Case 1

$\boldsymbol{i} = \{\mathbf{1}, \mathbf{2}, \dots, \mathbf{5}\}$	$\boldsymbol{w} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}\}$	$\boldsymbol{l} = \{\mathbf{1}, \mathbf{2}\}$

Table 2. X-coordinate for Case 1- possible sets of peg locations

px_{iwl}	$w = 1$		$w = 2$		$w = 3$		$w = 4$	
	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$
$i = 1$	2	3	4	5	6	7	7	8
$i = 2$	3	4	8	7	6	7	4	3
$i = 3$	1	2	4	4	4	3	1	1
$i = 4$	2	3	4	5	6	7	7	8
$i = 5$	4	5	9	8	7	8	6	6

Table 3. Y-coordinate for Case 1- possible sets of peg locations

py_{iwl}	$w = 1$		$w = 2$		$w = 3$		$w = 4$	
	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$
$i = 1$	2	2	4	5	4	3	2	2
$i = 2$	8	8	8	8	4	6	4	6
$i = 3$	2	2	3	4	5	5	4	3
$i = 4$	2	2	4	5	4	3	2	2
$i = 5$	6	6	6	6	3	4	3	4

To clarify the given information, the possible peg locations for the first two parts are depicted in Figure 25. As shown in this figure, point (5,5) is the only common point between the two sets of possible peg locations.

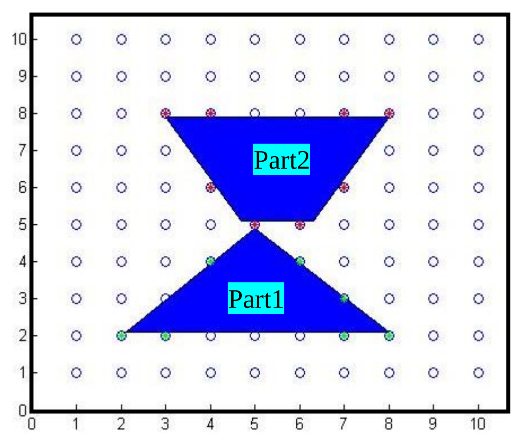


Figure 25. Possible peg locations for fixing first two parts

4.2 Case Study 2

This example is a medium-size case, and plan 10 different parts on the assembly system. The different models should determine the best locations for those part on the fixture as well as pegs to be inserted that can minimize the robot movements. The parameters are given in the following tables.

Table 4. Indices for Case 2

$\boldsymbol{i} = \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{10}\}$	$\boldsymbol{w} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}\}$	$\boldsymbol{l} = \{\mathbf{1}, \mathbf{2}\}$

Table 5. X-coordinate for Case 2- possible sets of peg locations

px_{iwl}	$w = 1$		$w = 2$		$w = 3$		$w = 4$	
	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$
$i = 1$	2	3	4	5	6	7	7	8
$i = 2$	3	4	8	7	6	7	5	4
$i = 3$	1	2	4	4	4	3	1	1
$i = 4$	3	4	5	6	7	8	8	9
$i = 5$	4	5	9	8	7	8	6	5
$i = 6$	2	3	5	5	5	4	2	2
$i = 7$	1	2	3	4	5	6	6	7
$i = 8$	2	3	7	6	5	6	4	3
$i = 9$	0	1	3	3	3	2	0	0
$i = 10$	2	3	4	5	6	7	7	8

Table 6. Y-coordinate for Case 2- possible sets of peg locations

py_{iwl}	$w = 1$		$w = 2$		$w = 3$		$w = 4$	
	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$
$i = 1$	2	2	4	5	4	3	2	2
$i = 2$	8	8	8	8	5	6	5	6
$i = 3$	2	2	3	4	5	5	4	3
$i = 4$	2	2	4	5	4	3	2	2
$i = 5$	8	8	8	8	5	6	5	6
$i = 6$	2	2	3	4	5	5	4	3
$i = 7$	2	2	4	5	4	3	2	2
$i = 8$	8	8	8	8	5	6	5	6
$i = 9$	2	2	3	4	5	5	4	3
$i = 10$	3	3	5	6	5	4	3	3

4.3 Case Study 3:

This example is a large-size problem and plans 50 parts on the fixture. This problem is used to investigate the efficiency of the proposed models in terms of solution time and quality. The problem's parameters are given in the Tables7 to 9.

Table 7. Indices for Case 3

$\boldsymbol{i} = \{\mathbf{1}, \mathbf{2}, \dots, \mathbf{50}\}$	$\boldsymbol{w} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}\}$	$\boldsymbol{l} = \{\mathbf{1}, \mathbf{2}\}$

Table 8. X-coordinate for Case 3- possible sets of peg locations

px_{iwl}	$w = 1$		$w = 2$		$w = 3$		$w = 4$	
	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$
$i = 1$	2	3	4	5	6	7	7	8
$i = 2$	3	4	8	7	6	7	5	4
$i = 3$	1	2	4	4	4	3	1	1
$i = 4$	3	4	5	6	7	8	8	9
$i = 5$	4	5	9	8	7	8	6	5
$i = 6$	2	3	5	5	5	4	2	2
$i = 7$	1	2	3	4	5	6	6	7
$i = 8$	2	3	7	6	5	6	4	3
$i = 9$	0	1	3	3	3	2	0	0
$i = 10$	2	3	4	5	6	7	7	8
$i = 11$	3	4	8	7	6	7	5	4
$i = 12$	1	2	4	4	4	3	1	1
$i = 13$	2	3	4	5	6	7	7	8
$i = 14$	3	4	8	7	6	7	5	4
$i = 15$	1	2	4	4	4	3	1	1
$i = 16$	3	4	5	6	7	8	8	9
$i = 17$	4	5	9	8	7	8	6	5
$i = 18$	2	3	5	5	5	4	2	2
$i = 19$	3	4	5	6	7	8	8	9

$i = 20$	4	5	9	8	7	8	6	5
$i = 21$	2	3	5	5	5	4	2	2
$i = 22$	1	2	3	4	5	6	6	7
$i = 23$	2	3	7	6	5	6	4	3
$i = 24$	0	1	3	3	3	2	0	0
$i = 25$	1	2	3	4	5	6	6	7
$i = 26$	2	3	7	6	5	6	4	3
$i = 27$	0	1	3	3	3	2	0	0
$i = 28$	2	2	4	5	4	3	2	2
$i = 29$	8	8	8	8	5	6	5	6
$i = 30$	2	2	3	4	5	5	4	3
$i = 31$	2	2	4	5	4	3	2	2
$i = 32$	8	8	8	8	5	6	5	6
$i = 33$	2	2	3	4	5	5	4	3
$i = 34$	2	2	4	5	4	3	2	2
$i = 35$	8	8	8	8	5	6	5	6
$i = 36$	2	2	3	4	5	5	4	3
$i = 37$	3	3	5	6	5	4	3	3
$i = 38$	9	9	9	9	6	7	6	7
$i = 39$	3	3	4	5	6	6	5	4
$i = 40$	1	1	3	4	3	2	1	1
$i = 41$	7	7	7	7	4	5	4	5
$i = 42$	1	1	2	3	4	4	3	2
$i = 43$	3	3	5	6	5	4	3	3
$i = 44$	9	9	9	9	6	7	6	7
$i = 45$	3	3	4	5	6	6	5	4
$i = 46$	1	1	3	4	3	2	1	1
$i = 47$	7	7	7	7	4	5	4	5
$i = 48$	1	1	2	3	4	4	3	2
$i = 49$	3	3	5	6	5	4	3	3
$i = 50$	9	9	9	9	6	7	6	7

Table 9. Y-coordinate for Case 3- possible sets of peg locations

px_{iwl}	$w = 1$		$w = 2$		$w = 3$		$w = 4$	
	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$	$l = 1$	$l = 2$
$i = 1$	2	2	4	5	4	3	2	2
$i = 2$	8	8	8	8	5	6	5	6
$i = 3$	2	2	3	4	5	5	4	3
$i = 4$	2	2	4	5	4	3	2	2
$i = 5$	8	8	8	8	5	6	5	6
$i = 6$	2	2	3	4	5	5	4	3
$i = 7$	2	2	4	5	4	3	2	2
$i = 8$	8	8	8	8	5	6	5	6
$i = 9$	2	2	3	4	5	5	4	3
$i = 10$	3	3	5	6	5	4	3	3
$i = 11$	9	9	9	9	6	7	6	7
$i = 12$	3	3	4	5	6	6	5	4
$i = 13$	1	1	3	4	3	2	1	1
$i = 14$	7	7	7	7	4	5	4	5
$i = 15$	1	1	2	3	4	4	3	2
$i = 16$	3	3	5	6	5	4	3	3
$i = 17$	9	9	9	9	6	7	6	7
$i = 18$	3	3	4	5	6	6	5	4
$i = 19$	1	1	3	4	3	2	1	1
$i = 20$	7	7	7	7	4	5	4	5
$i = 21$	1	1	2	3	4	4	3	2
$i = 22$	3	3	5	6	5	4	3	3
$i = 23$	9	9	9	9	6	7	6	7
$i = 24$	3	3	4	5	6	6	5	4
$i = 25$	1	1	3	4	3	2	1	1
$i = 26$	7	7	7	7	4	5	4	5
$i = 27$	1	1	2	3	4	4	3	2
$i = 28$	2	3	4	5	6	7	7	8
$i = 29$	3	4	8	7	6	7	5	4
$i = 30$	1	2	4	4	4	3	1	1

$i = 31$	3	4	5	6	7	8	8	9
$i = 32$	4	5	9	8	7	8	6	5
$i = 33$	2	3	5	5	5	4	2	2
$i = 34$	1	2	3	4	5	6	6	7
$i = 35$	2	3	7	6	5	6	4	3
$i = 36$	0	1	3	3	3	2	0	0
$i = 37$	2	3	4	5	6	7	7	8
$i = 38$	3	4	8	7	6	7	5	4
$i = 39$	1	2	4	4	4	3	1	1
$i = 40$	2	3	4	5	6	7	7	8
$i = 41$	3	4	8	7	6	7	5	4
$i = 42$	1	2	4	4	4	3	1	1
$i = 43$	3	4	5	6	7	8	8	9
$i = 44$	4	5	9	8	7	8	6	5
$i = 45$	2	3	5	5	5	4	2	2
$i = 46$	3	4	5	6	7	8	8	9
$i = 47$	4	5	9	8	7	8	6	5
$i = 48$	2	3	5	5	5	4	2	2
$i = 49$	1	2	3	4	5	6	6	7
$i = 50$	2	3	7	6	5	6	4	3

As mentioned before, this example will be used to investigate the efficiency of the different models in terms of solution quality and time. This is due to the fact that a large size problem might increase the problem's complexity drastically. Recalling the formula from Chapter 3, the number of feasible solutions based on the Basic Model can be calculated by:

$$Number\ of\ Feasible\ Solution = \binom{l}{1}^{\mathrm{wi}}$$

Therefore, while the first case has $2^{20} = 1{,}048{,}576$ different feasible solutions, the feasible region for the third case is significantly larger and it will be $2^{200} = 1.606938\mathrm{e} + 60$. Adding more flexibility such as enabling the model to translate and rotate the parts on the cradle, or determining the job sequence, to the models, the feasible region will expand dramatically. Therefore, it is highly important to investigate the efficiency of the proposed models in terms of time and quality of the optimum solution.

Chapter 5. Results and Discussion

To solve the proposed numerical examples with different mathematical models, models are formulated using GAMS and MATLAB. All models except the second (NLRTE) can be classified as mixed integer linear programming models. Therefore, they can be solved using CPLEX solver. Due to its non-linear nature, the NLRTE is solved using BAROON solver. This section presents the results of running the different models for the proposed numerical examples. As mentioned before, applicability of the models in real industries highly depends on their effectiveness regarding solution time and quality. Therefore, a comparison in different results is made to investigate the models' effectiveness, especially for large-size problems. Note that all models are solved on a PC with a Core i7 3.4 GHz processer and 16 GB RAM.

5.1 Basic Model Results

In this model, parts are fixed on the cradle, and the model should determine the best place for pegs to be inserted among the predetermined possible peg locations. The results of running the basic model for 1000 seconds are given below:

Table 10. Summary of Basic Model results

	Number of parts	**Number of fixed pegs**
Case 1	5	2
Case 2	10	7
Case3	50	29

5.2 NLRTE Model Results

The non-linear rotation and translation-enabled model is the second mathematical model that has been developed to enable the system to take all the part rotations and translations into consideration. As shown in Table 11, the higher flexibility of the NLRTE model improves the results significantly. Note that non-linearity nature of this model increase the solution time drastically. The solutions are generated running the model for 150,000 seconds.

Table 11. Summary of NLRTE model results

	Number of parts	**Number of fixed pegs**
Case 1	5	12
Case 2	10	23
Case3	50	100

5.3 LRTE Model Results

The LRTE model has been developed by linearizing the NLRTE model to decline the solution time. The results for running the LRTE model for 1000 seconds are presented in Table 12:

Table 12. Summary of LRTE model results

	Number of parts	**Number of fixed pegs**
Case 1	5	12
Case 2	10	23
Case3	50	110

For instance, the final peg locations for the first problem are summarized in Table 13.

Table 13. Optimal peg locations for the first problem

Part No.	W1	W2	W3	W4	Number of fixed pegs				
					Part1	Part2	Part3	Part4	Part5
Part1	(4,7)	(6,6)	(5,3)	(4,3)					
Part2	(3,6)	(6,6)	(5,3)	(4,3)	3				
Part3	(2,5)	(4,3)	(5,3)	(3,6)		3			
Part4	(5,3)	(2,5)	(3,6)	(5,7)			3		
Part5	(5,7)	(5,3)	(3,3)	(3,6)				3	
					Total No. of fixed pegs				12

The result shows that 12 pegs could be fixed between any two successive parts. As shown in Figure 26, the first, third and fifth parts were rotated by 270 degree, and the second part was rotated by 90 degree in order to optimize the objective function. This also shows the initial and the new part locations in yellow and blue and labelled 1 and 2, respectively. Figure 27 shows how different parts overlapped after running the model.

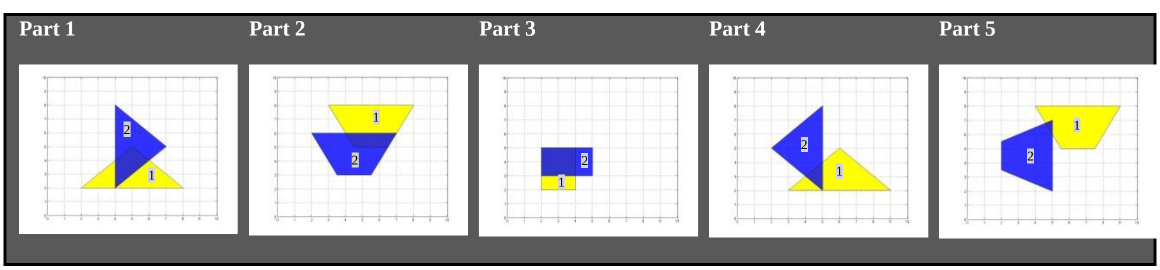


Figure 26. Parts positions on cradle before (blue) and after planning (yellow)

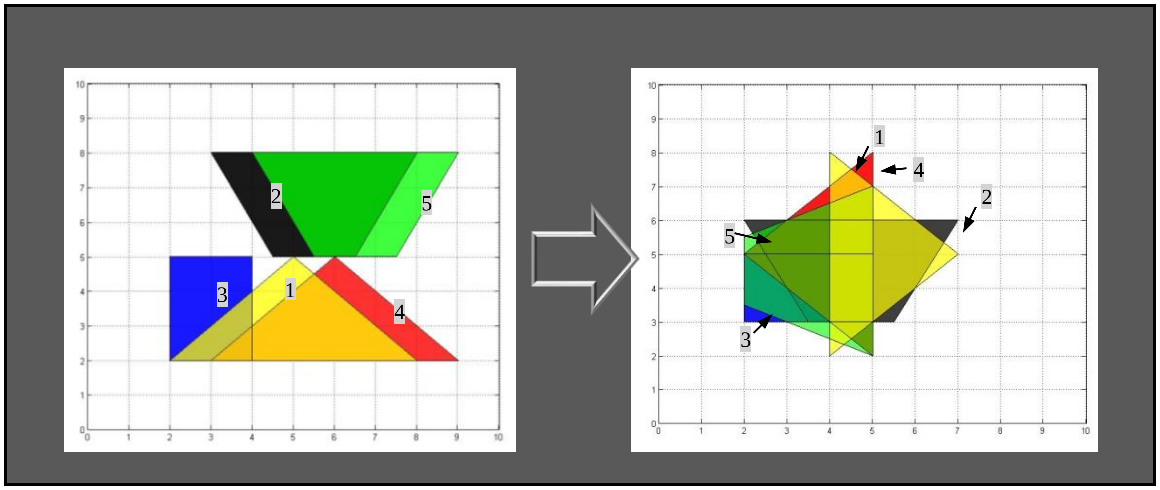


Figure 27. Overlapping parts before and after planning

5.4 SLRTE Model Result

Scheduling-enabled LRTE model has been developed to enable the system to determine the best order of parts in the assembly system to minimize the number of pegs replacements. The following results are generated by running the model considering different time limits.

Table 14. Summary of SLRTE's results

	Number of part	# of Fixed peg 1,000 seconds	# of Fixed peg 10,000 seconds	# of Fixed peg 150,000 seconds
Case 1	5	13	13	13
Case 2	10	25	25	29
Case3	50	N.A.	N.A.	171

5.5 RTPO Model Result:

The Robot Travel Path Optimization model is a fundamentally different type of model that minimizes not only the number of peg replacements but also the robot travel path in a dynamic environment. RTPO can use any of the four presented models (Basic Model, NLRTE, LRTE, and SLRTE) to determine the peg locations. Hence, a simple comparison between the new and the previous models' results won't be possible. This section describes how the RTPO model uses the Basic Model and the LRTE model to find the optimized robot travel paths for three different-size case studies.

For the first case study with 5 different parts, the optimized routes that the robot should travel to replace the pegs are given in Figure 28. By using the Basic Model, 14 pegs need to be replaced. Using the LRTE model significantly drop this number to 4 pegs. Hence, by using the LRTE model instead of the Basic Model, the total travel path is reduced by 42%. In addition, reducing the number of pegs that need to be replaced, the total vertical movement decreases by 72%.

In Figure 28, the routes are mapped between each pair of successive jobs. Therefore, for planning 5 parts on the system, 4 graphs are given. In this figure, the red stars and green circles represent current and next peg locations, respectively.

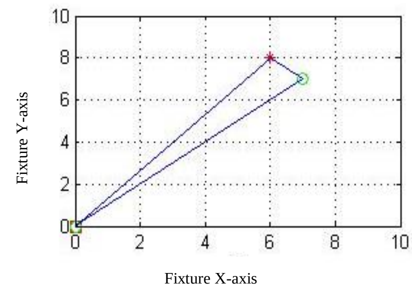
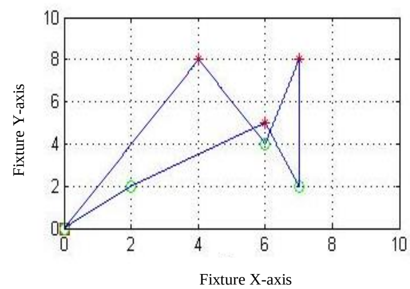
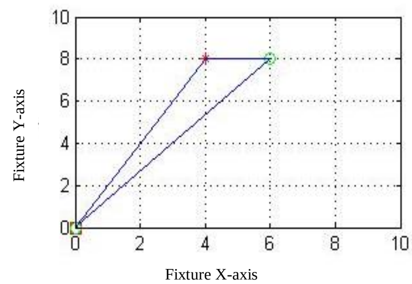
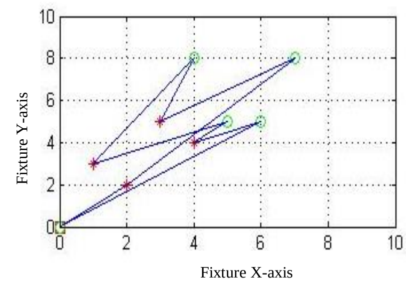
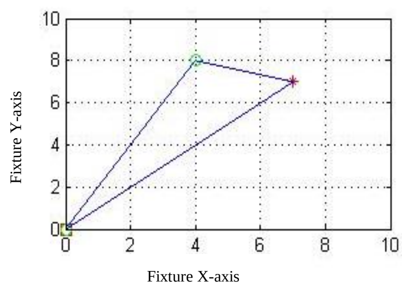
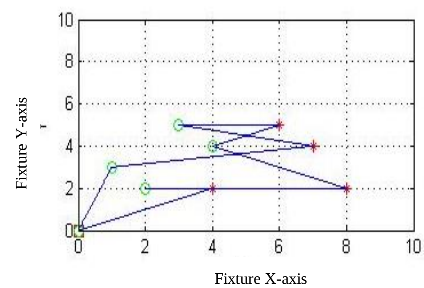
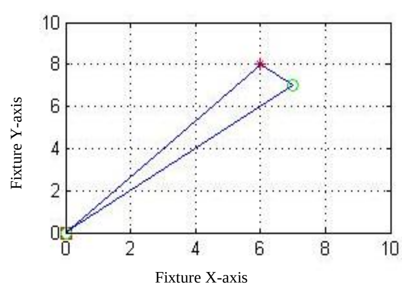
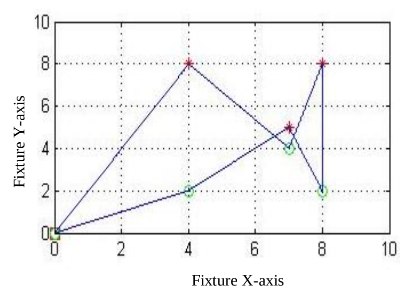
<table>
<tr><th></th><th>a) Optimized robot travel path using LRTE and RTPO models</th><th>b) Optmized robot travel path using Basic model and RTPO models</th></tr>
<tr><td>Robot travel path to fix part 2 after part 1</td><td>
</td><td>
</td></tr>
<tr><td>Robot travel path to fix part 3 after part 2</td><td>
</td><td>
</td></tr>
<tr><td>Robot travel path to fix part 4 after part 3</td><td>
</td><td>
</td></tr>
<tr><td>Robot travel path to fix part 5 after part 4</td><td>
</td><td>
</td></tr>
</table>

Figure 28. Optimized robot travel path for Case 1 using a) LRTE and RTPO models b) Basic and RTPO Models

In the next example, 10 different part are going to be fixed on the cradle one after another by the minimum time and efforts. In the following picture, the optimized robot movement is depicted between each pairs of part using LRTE and Basic Models. When using the LRTE rather than the Basic Model, the total distance is reduced by 39% and the number or pegs by 59%.

a) Optimized robot travel path using the LRTE and RTPO models	b) Optmized robot travel path using the Basic model and RTPO models

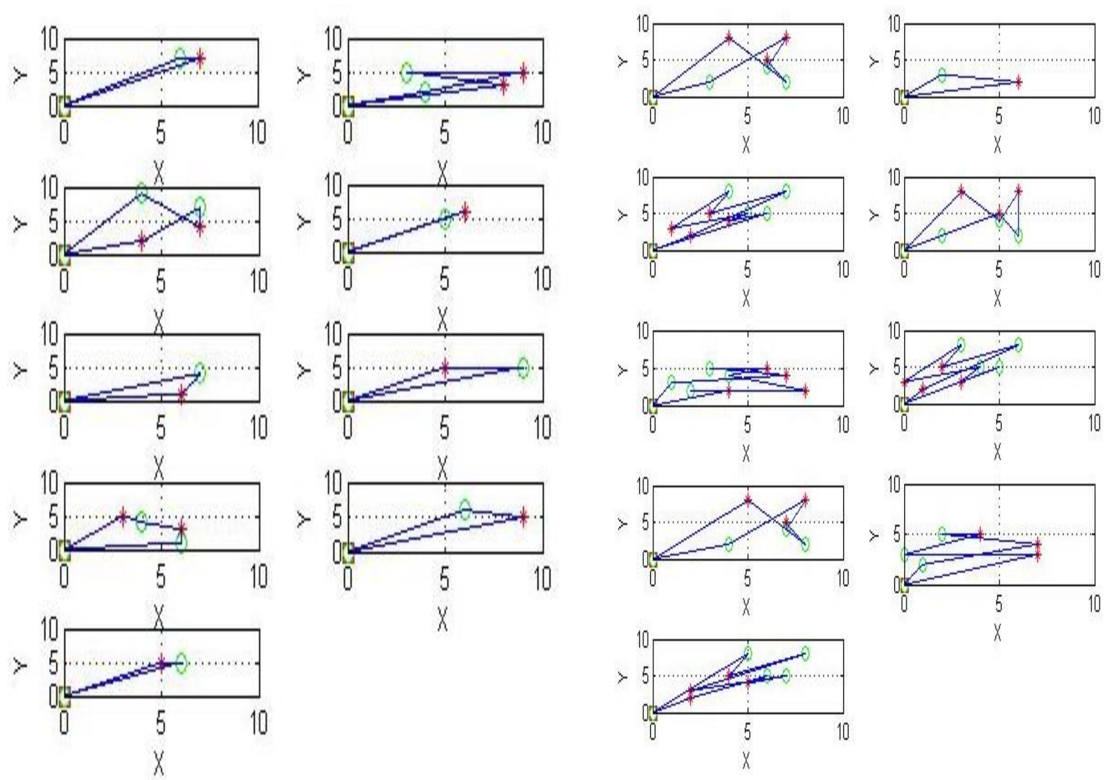


Figure 29. Optimized robot travel path for Case 2 using a) LRTE and RTPO models b) Basic and RTPO models

The following table summarizes the results for three numerical examples.

Table 15. Summary of RTPO model results

	Number of parts	Reduction in Horizontal Movement	Reduction in Vertical Movement
Example 1	5	42%	72%
Example 2	10	39%	59%
Example 3	50	56%	45%

5.6 New Fixture Design Results

As explained in chapter 3, the new fixture design can be applied to the different proposed models by making few modifications. In this section the similar case studies are solved on the new fixture using the Basic and the LRTE models.

Table 16. Summary of results for new fixture design using LRTE and Basic Model

	Number of parts	# of fixed pegs using Basic Model	# of fixed pegs using LRTE
Example 1	5	2	16
Example 2	10	12	24
Example 3	50	29	121

5.7 Discussion:

As described in the previous sections, the proposed models have been progressively developed to improve the quality of the results. In this section a comparison is made between the first four models, the Basic, NLRTE, LRTE, and SLRTE models, to show their efficiency in converging to the optimal solution. The results and solution times for the models are given in Table 17 and also in the following figures.

Table 17. Comprehensive comparison between models 1 to 4

	Basic			NLRTE			LRTE			SLRTE		
	1000 sec	10000 Sec	150000 sec	1000 sec	10000 sec	150000 sec	1000 sec	10000 sec	150000 sec	1000 sec	10000 sec	150000 sec
Case 1	2	2	2	7	8	12	12	12	12	13	13	13
Case 2	7	8	12	8	12	23	23	23	23	25	25	29
Case 3	29	29	29	N.A	N.A	100	110	110	120	N.A	N.A	171

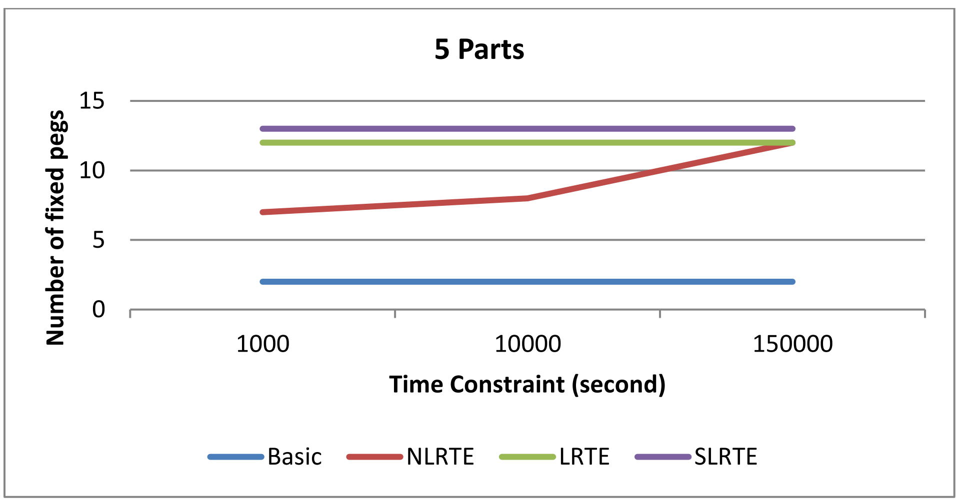


Figure 30. Comparison of different models in case 1

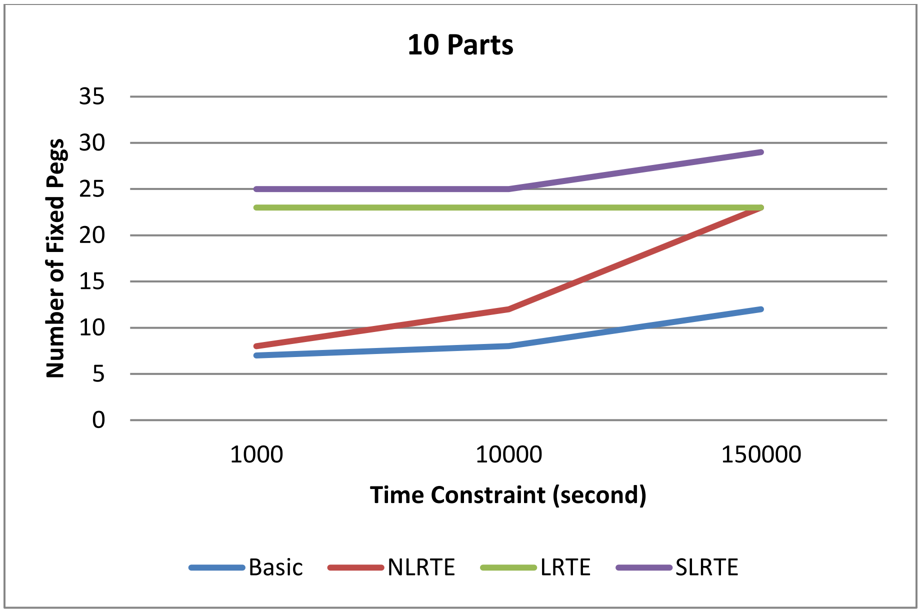


Figure 31. Comparison of different models in case 2

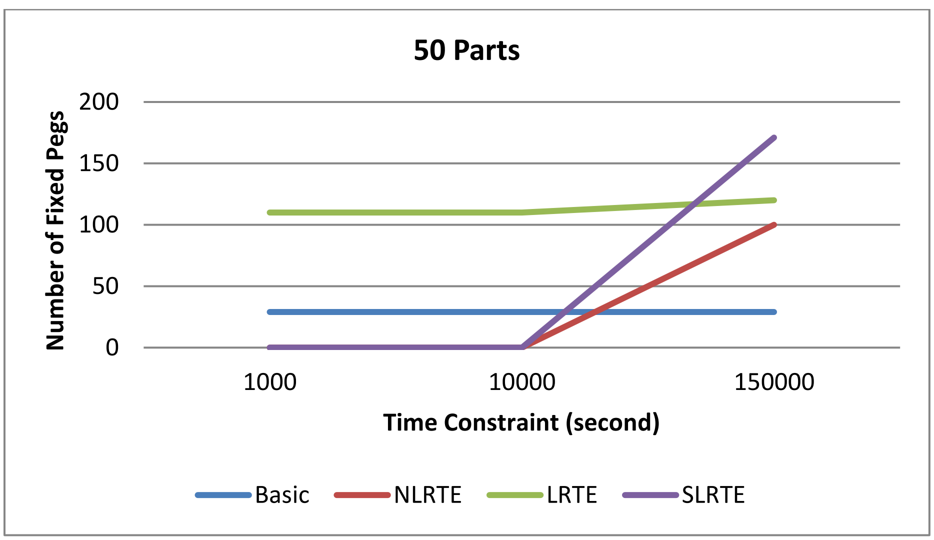


Figure 32. Comparison of different models in case 3

As mentioned before, the solution time is always a big concern for Mixed Integer Programming models. Therefore, in Figure 30 to 32 by relaxing the time limit for different models, the models' convergence to the optimal solution is examined. As shown, the result for the Basic Model is almost robust and it does not change by expanding the time limit. However, its results are often dominated by other models.

As could be expected, the NLRTE model needs a significant amount of time to converge to the appropriate solutions. The non-linear nature of that model cause the solution time to increase exponentially. Hence, in the last case the NLRTE was not able to solve the model at time limit of 1,000 and 10,000 seconds.

As shown in the above figures, the LRTE model was very effective in converging to the optimum solution. For instance, in the first two cases, the best results were given at time limit of 1000 seconds, and they were significantly improved compared to the first two models. In the second case, the best solution that was achieved by the NLRTE model in 150,000 seconds was reported by the LRTE in the first 1,000 seconds.

Finally, the SLRTE model improved the results by enabling the system to decide about job sequences. Therefore, the required time to converge to the optimal solution in the last case with 50 different parts increased and the model could not find a feasible solution in the first 10,000 seconds.

Chapter 6. Conclusion

In the last few decades, the manufacturing world has changed dramatically to meet customers' needs and expectations. One of the main changes that have happened in the manufacturing industries is moving from mass production to mass customization. Manufacturing one product that can meet all customers' needs in different markets is no longer possible. As a result, manufacturing firms are trying to produce a variety of products that can meet different needs in different markets. This means manufacturers are unable to take advantage of mass production, economy of scale.

To overcome that problem, changeability in manufacturing systems has been developed. Changeability can improve a production system to adapt to a variety of products in a timely manner with fewer costs and efforts. Changeability enablers can be implemented on the different organizational levels from workstation to the whole production network including changeover-ability, reconfigurability, flexibility, transformability and agility. Modularity is the basic component of reconfigurability. On the operational level, one of the main strategies that can improve manufacturing changeability is designing a modular fixture that can hold a variety of parts with different geometries.

Among different types of modular fixtures, in this study a hole-pattern modular fixture has been considered to increase the changeability of an automated assembly system. This assembly system is controlled by an algorithm and a robot that is located on top of a conveyor belt loop. The program determines the part locations as well as the pegs' positions in order to minimize the total time and efforts associated with fixture preparation. By inserting four pegs in appropriate positions, the system is able to produce the necessary reaction forces to fix a given set of parts with wide geometrical variability, doing so with minimal changeover time.

The main goal of this study was to improve the sustainability of an automated assembly system by reducing the fixture setup time. To achieve this goal, several mathematical models have been developed to be used in different circumstances. First, the Basic Model was proposed to find the best peg locations among predefined sets. Second, NLRTE Model was developed that enables the system to take all part rotations and translations into consideration. Due to the non-linearity nature of NLRTE model, by increasing in the parameters' sizes the solution time was increased exponentially. Then LRTE model has been developed to overcome this problem by linearizing the previous model. Following this, the SLRTE model has been proposed that enables the system to schedule different jobs in the system to improve the result. These models' objectives were the minimization of the number of pegs that need to be replaced between each pair of jobs. This objective can be referred as minimizing the vertical robot movement that need to be done. The RTPO not only considers robot's vertical but also horizontal movement in the system objective function. Eventually, a new fixture design is proposed that can improve the system performance by being able to adapt to more parts' geometries. By making a few modifications, the five previous models have been extended to be used for this new fixture design.

At this point, researchers propose three case studies with different parameters' sizes to investigate the models' efficiencies. The results show that the different models are able to significantly reduce fixture setup time.

It should be noted that since the problem has been formulated as a mixed integer programming model, the required time to find the optimum solution must be investigated. Hence, the researchers set different time limits for the models to draw a comprehensive comparison between them in terms of their quality of results and required time to converge to those solutions.

Limitations:

As discussed earlier, the research question was formulated as a mixed integer programming (MIP) problem. The results showed that the solution times were considerable. For instance, the SLRTE model could not find any feasible solution in a 10,000 seconds limit. Researchers tried to develop a Genetic Algorithm to improve the solution time, but due to the huge size of the problem (recall formula 3-1) the proposed Genetic Algorithm was not time-effective and could not converge to a feasible solution after a day of running. Note that formula 3-1 was proposed for the basic problem and that after enabling the model to rotate and translate the parts on the cradle the number of feasible solutions was much higher.

Future Work:

Future research can be targeted at developing more efficient algorithms in terms of solution time and quality. Besides that, developing a general algorithm that can simultaneously optimize vertical and horizontal movements might be considered as a future research goal.

By extending the developed mathematical models for the new fixture design, the authors are going to prove the adaptability of the proposed algorithms for other purposes and manufacturing problems. For instance, the proposed models can be effectively extended to plan the movements of a crane in a manufacturing site. In this system, crane represent the robot in the proposed assembly system and the pegs are the parts that need to be relocated. Another example is planning an automated assembly of printed circuit boards. For such a system, the goals of the proposed models are to find an efficient way to drill holes and place different electronic components on the boards.

References:

[1] H. ElMaraghy, G. Schuh, W. ElMaraghy, F. Piller, P. Schönsleben, M. Tseng, and a. Bernard, "Product variety management," *CIRP Ann. - Manuf. Technol.*, vol. 62, no. 2, pp. 629–652, Jan. 2013.

[2] J.-L. Hou and A. J. C. Trappey, "Computer-aided fixture design system for comprehensive modular fixtures," *Int. J. Prod. Res.*, vol. 39, no. 16, pp. 3703–3725, Jan. 2001.

[3] Z. M. Bi and W. J. Zhang, "Flexible fixture design and automation: Review, issues and future directions," *Int. J. Prod. Res.*, vol. 39, no. 13, pp. 2867–2894, Jan. 2001.

[4] H. Wang, Y. (Kevin) Rong, H. Li, and P. Shaun, "Computer aided fixture design: Recent research and trends," *Comput. Des.*, vol. 42, no. 12, pp. 1085–1094, Dec. 2010.

[5] D. Mourtzis, S. Fotia, and M. Doukas, "Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth," *IFIP Adv. Inf. Commun. Technol.*, vol. 460, pp. 592–601, 2015.

[6] S. J. H. V. Maler-Speredelozzi, Y. Koren, "Convertibility Measures for Manufacturing Systems," *CIRP Ann. - Manuf. Technol.*, vol. 52, no. 1, pp. 367–370, 2007.

[7] Y. Koren, "Chapter 3 General RMS Characteristics . Comparison with Dedicated and Flexible Systems," no. 1.

[8] a. Chaube, L. Benyoucef, and M. K. Tiwari, "An adapted NSGA-2 algorithm based dynamic process plan generation for a reconfigurable manufacturing system," *J. Intell. Manuf.*, vol. 23, no. 4, pp. 1141–1155, 2012.

[9] M. Jonsson and G. Ossbahr, "Aspects of reconfigurable and flexible fixtures," *Prod. Eng.*, vol. 4, no. 4, pp. 333–339, 2010.

[10] S. Arzanpour, J. Fung, J. K. Mills, and W. L. Cleghorn, "Flexible fixture design with applications to assembly of sheet metal automotive body parts," *Assem. Autom.*, vol. 26, no. 2, pp. 143–153, 2006.

[11] T. Algeddawy and H. Elmaraghy, "Product Variety Management in Design and Manufacturing : Challenges and Strategies," in *4th International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2011)*, 2011, pp. 518–523.

[12] H. Fuwen, "S e n s o r s & T r a n s d u c e r s An Intelligent Vacuum Fixture Controlled by Built-in PLC in CNC System," *Sensors & Transducers*, vol. 181, no. 10, pp. 45–51, 2014.

[13] S. Nelaturi, A. Rangarajan, C. Fritz, and T. Kurtoglu, "Automated fixture configuration for rapid manufacturing planning," *Comput. Des.*, vol. 46, pp. 160–169, Jan. 2014.

[14] A. S. Wallack and J. F. Canny, "Modular Fixture Design for Generalized Polyhedra," in *International Conference on Robotics and Automation*, 1996, no. April, pp. 830–837.

[15] H. Wang, Y. (Kevin) Rong, H. Li, and P. Shaun, "Computer aided fixture design: Recent research and trends," *Comput. Des.*, vol. 42, no. 12, pp. 1085–1094, Dec. 2010.

[16] X. Kang and Q. Peng, "Recent Research on Computer-Aided Fixture Planning," *Recent Patents Mech. Eng.*, vol. 2, no. 1, pp. 8–18, Jan. 2009.

[17] M. Vasundara and K. P. Padmanaban, "Recent developments on machining fixture layout design, analysis, and optimization using finite element method and evolutionary techniques," *Int. J. Adv. Manuf. Technol.*, vol. 70, no. 1–4, pp. 79–96, Aug. 2013.

[18] H. Du and G. C. I. Lin, "Development of an automated flexible fixture for planar objects," *Robot. Comput. Manuf.*, vol. 14, no. January, pp. 173–183, 1998.

[19] B. Shirinzadeh, "Flexible and automated workholding systems," *Ind. Robot An Int. J.*, vol. 22, no. 2, pp. 29–34, 1995.

[20] L. Xiong, R. Molfino, and M. Zoppi, "Fixture layout optimization for flexible aerospace parts based on self-reconfigurable swarm intelligent fixture system," *Int. J. Adv. Manuf. Technol.*, vol. 66, no. 9–12, pp. 1305–1313, 2013.

[21] Y. Wu, S. Gao, and Z. Chen, "Automated modular fixture planning based on linkage mechanism theory," *Robot. Comput. Integr. Manuf.*, vol. 24, no. 1, pp. 38–49, 2008.

[22] G. Peng, G. Wang, W. Liu, and H. Yu, "A desktop virtual reality-based interactive modular fixture configuration design system," *Comput. Des.*, vol. 42, no. 5, pp. 432–444, May 2010.

[23] Y. (Kevin) Rong and Y. Bai, "Automated Generation of Fixture Configuration Design," *J. Manuf. Sci. Eng.*, vol. 119, no. 2, pp. 208–219, May 1997.

[24] H. Tohidi and T. AlGeddawy, “Planning of Modular Fixtures in a Robotic Assembly System,” in *48th CIRP Conference on MANUFACTURING SYSTEMS - CIRP CMS 2015*, 2015.

[25] C. E. Miller, a. W. Tucker, and R. a. Zemlin, “Integer Programming Formulation of Traveling Salesman Problems,” *J. ACM*, vol. 7, no. 4, pp. 326–329, 1960.