HIGH-STIFFNESS, LOCK-AND-KEY HEAT-REVERSIBLE LOCATOR-SNAP SYSTEMS FOR THE DESIGN FOR DISASSEMBLY

by

Mohammed Mounir Shalaby

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering and Scientific Computing)
in The University of Michigan
2008

Doctoral Committee:

Associate Professor Kazuhiro Saitou, Chair
Professor Sridhar Kota
Professor Romesh Saigal
Assistant Professor Donald E. Malen

Dedication

To my parents Prof. Mounir Shalaby and Prof. Samia Hemeda;

To my wife Dr. Maii Abu-Taleb;

To my son Ahmad

Acknowledgement

I would like to express my sincere gratitude to my advisor, Prof. Kazuhiro Saitou, for his endless support and guidance during the throughout every step of this thesis. I am incredibly fortunate to have him as my advisor. I would like to express my appreciation to my sponsors “Toyota Motor Corporation, Japan”, “Sony Corporation, Japan” and the National Science Foundation (NSF).

Thanks to my friend and office mate, Karim Hamza, as we discussed a lot of research topics together that really inspired me. He also helped my with some optimization and programming issues. Thanks to my friend Hatem Orban for his help in some ABAQUS related issues. I am grateful to Dr. Byungwoo Lee, for his advices regarding the Screw theory and offering me to work on variation analysis during my internship. I am also grateful to Dr. Hosam Fathy and my past office mate Dr. Naesung Lyu for their help. I would like to thank all my Egyptian friends; they made my stay enjoyable and were really supportive.

Table of Contents

List of Figures

List of Tables

List of Appendices

Chapter 1: Introduction

1.1 Background

The increasing use of consumer goods and the reduction of their product lifecycle have lead to an exceptional number of abandoned products. More than 70% of the United States landfills reached their permitted capacity in 1997 [1]. It has been reported that around 3 billion units of electric and electronic waste (E-waste) will be scrapped during the rest of the decade in the US alone [2]. As a result, the moral sense of obligation and the regulatory requirements in many countries have driven manufacturers to consider effective part reuse and material recycling at the end of product life (EOL) at the design stage. For instance, the European Union (EU) directive on Waste Electrical and Electronic Equipment (WEEE) currently requires manufacturers to be responsible for collecting, treating and recycling the equipment they sold after 2005 [3].

Although meeting the regulatory requirements is obligatory, EOL treatment of equipment is generally governed by economic considerations [4]. Thus, it is crucial to design and manufacture products with the ability to recycle and reuse their components at EOL with minimum cost and maximum return. Disassembly of the retired products is a key factor in achieving component reuse and/or remanufacture. It is also crucial to the recycling of shredded components to produce the same grade raw material.

1.2 Motivation

1.2.1 Recycling of consumer goods

Conventionally, when components reach their EOL stage, they are recycled by directly shredding parts, separating the materials (magnetic, density, *etc*.), then recycling as shown in the typical recycling process in Figure 1.1 [5]. Unfortunately, such processes ignore the possible reuse and remanufacturing of components, for example for use as spare parts or in different products. In addition, the presence of incompatible materials after shredding is a significant barrier to achieving optimum material quality after recycling. For instance, aluminum cannot be recycled to the same grade if it contains contaminants [6]. Another example is mixed plastics: if they come from shredding, they will be of substantially low value, especially because plastic purification has (<90%) efficiency [5].

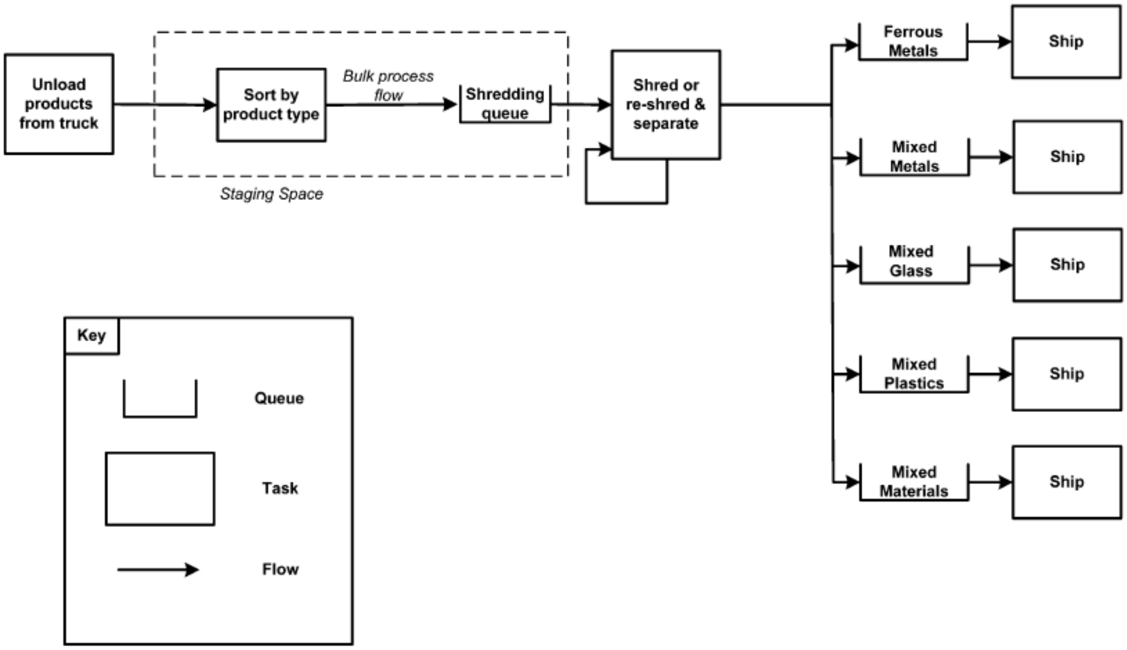


Figure 1.1: Typical material recovery process [5].

It is obvious that better material recycling and component reuse can be achieved if disassembly is included in the recycling process. Recycling of electronic components now includes disassembly and reuse as shown in Figure 1.2 [7]. While the total cost of disassembly includes several components (logistics, material handling, *etc*.), a key component is the cost and effort associated with the actual disassembly action [8]. Disassembly is a labor intensive process and is hard to automate because the disassembly plants receive a wide variety of products of different types; each product has a different disassembly procedure. Thus it is important to make the disassembly process economic and to some extent automated to minimize the labor cost.

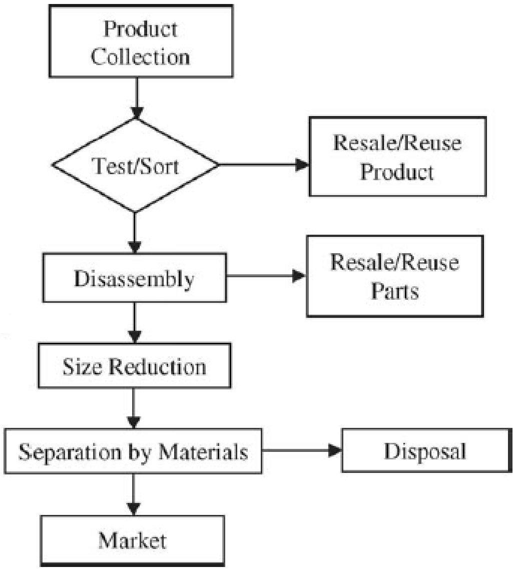


Figure 1.2: Simplified diagram for recycling of electronic products [7].

1.2.2 Recycling of vehicles

With the increase in the volume of the vehicles produced in modern society, it is important to find more energy-efficient vehicles. In addition, it is essential to find efficient ways of disposing scrapped vehicles at end of their lives. Automotive manufactures use more plastics and aluminum and reduce the usage of steel in their vehicles aiming for weight reduction and, consequently, higher energy efficiency. For instance, aluminum space frame, Figure 1.3 [9], offers significant environmental benefits over traditional steel sheet bodies owing to its light weight (40% lighter), and lower noise and vibration characteristics [10]. Aluminum space frame vehicles can have lower production costs than those from steel; aluminum also provides greater flexibility in the overall design and production process [11]. Although the production of aluminum from its ore is energy intensive, the recycling of aluminum is energy efficient, and requires only 5% of the energy required to produce the aluminum from its ore. Unfortunately, recycling contaminated aluminum, with incompatible materials, will produce cast aluminum, which has minimal use due to its poor material properties [6]. As a result, to improve the recyclability of aluminum space frame bodies, clean separation between incompatible materials is essential to remove any contaminants and to generate a closed-loop recycling process.

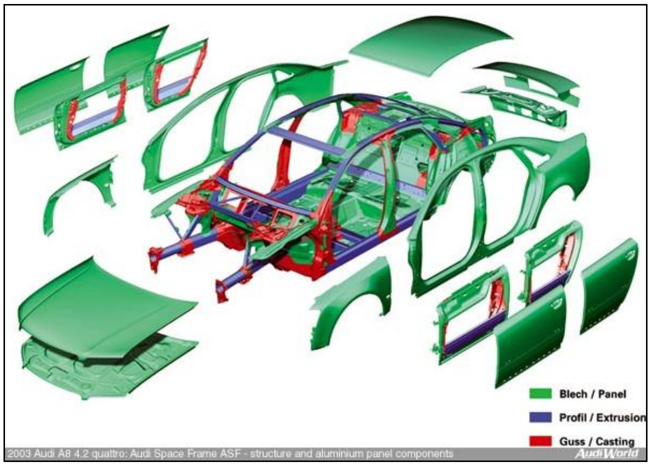

🟩 Blech / Panel
🟦 Profil / Extrusion
🟥 Guss / Casting
2003 Audi A8 4.2 quattro: Audi Space Frame ASF - structure and aluminium panel components
AudiWorld

Figure 1.3: Audi Space Frame [9].

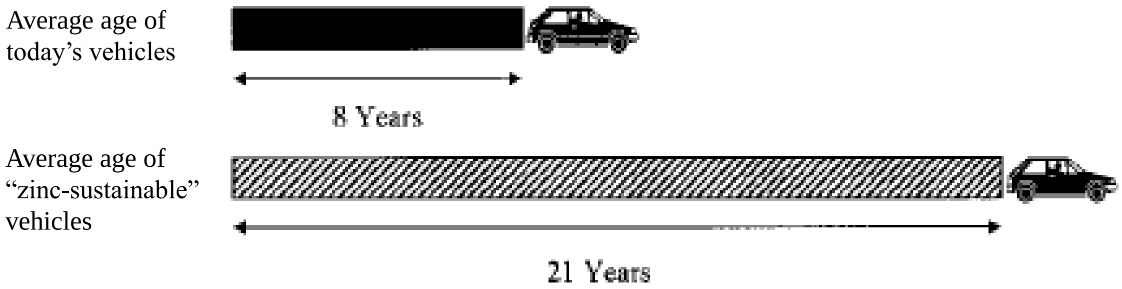

Average age of today's vehicles
8 Years
Average age of "zinc-sustainable" vehicles
21 Years

Figure 1.4: Comparison of the average age of today's vehicles in the United states and the average age of vehicles required to achieve sustainability from zinc perspective [12].

Recycling vehicles is also important to achieve sustainability from a materials perspective. For instance, the average age of today's vehicles is less than the average age necessary to achieve materials sustainability [12], Figure 1.4. Legislative rules have been enforced to require manufacturers to consider recycling vehicles at the end of their lives. For instance, the European Union directive 2000/53/EC [13] requires the reuse and recovery of 85% by mass of vehicles. This number is to increase to 95% by 2015 [14]. The directive also encourages manufacturers to improve the ease of dismantling of vehicles. In Japan, the End-of-Life Vehicle (ELV) recycling law, came into force in January 2005, and requires that the overall recycling rates of ELV be 88% in the years 2005~2009, 92% in the years 2010~2014 and 95% after 2015 [15]. As a result, it is crucial for manufacturers to consider the EOL recycling strategies at the design stage.

1.2.3 Design for Disassembly

While a number of strategies suggested by the concepts of Design for Assembly (DFA) and Design for Manufacturing (DFM) [16] can be applied to disassembly, a number of researchers pointed out that products designed for easy assembly do not necessarily facilitate easy disassembly [1], [17]-[23] for the following reasons:

- Adhesives may be used to easily join parts, but the resulting joints are hard to detach without destruction. In addition, the residue of one part over the other can easily prevent closed-loop recycling needed to ensure optimum grade of the recycled material.
- The conditions of products at disassembly may be different from the conditions during initial assembly. For example, bolted joints can be extremely hard to detach due to corrosion and wear.
- Snap-fits are easy to attach, but they are usually hidden and may be hard to find and disassemble (Figure 1.5a) at the disassembly stage. They are also prone to accidental disassembly (Figure 1.5b) and can have unfavorable aesthetic appeal (Figure 1.5c-d).

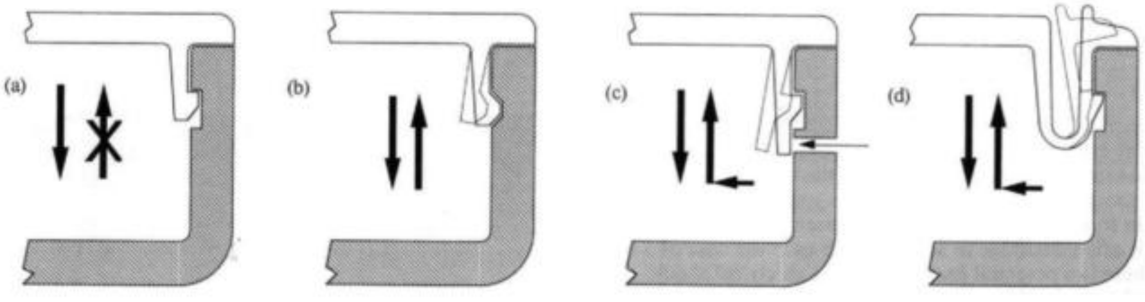


Figure 1.5: Different snap-fit types (a) non-disassemblable snap, and disassemblable snaps (b) prone to accidental disassembly, (c) and (d) affecting the aesthetic appeal [24].

Design for Disassembly (DFD), a newer design methodology currently receiving researchers' attention [17], [25]-[27], has been proposed in the literature to allow the design of a product while taking into account the need for disassembly for component repair, reuse and recycling. There are certain guidelines imposed by DFD regarding the materials and fasteners used in a product, as well as the product structure itself to ease disassembly [28], [29]. For example:

- Minimize the number of different types of materials used.
- Use the same or at least compatible materials for subassemblies.
- Optimal use of removable, yet robust, fasteners and joints. Kuo [30] suggests maximizing the use of single robust joints that allow many parts to become loose when these joints are removed.
- Choose easy-to-remove fasteners that are easily accessible.
- Eliminate incompatible adhesives.
- Minimize the number of parts.

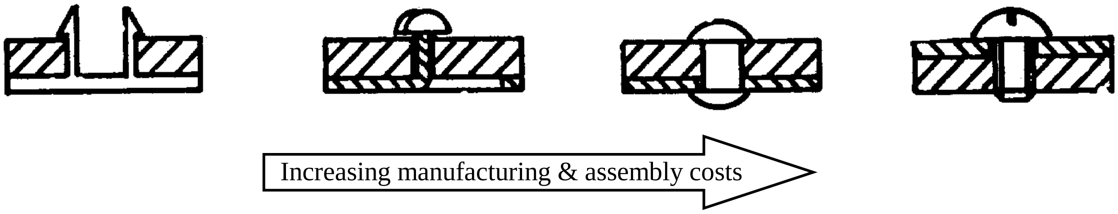


Figure 1.6: Different joining types and their relative manufacturing and assembly costs.

It can be concluded from above that the use of the types of fasteners preferred for DFA, like adhesives and screws, may not be suitable for DFD. On the other hand, although integral attachments, e.g. snap-fits, are a preferred joining method for the design for DFA/ DFM [31], [32], they actually appear to be a good candidate for DFD as well. Integral attachments minimize the different materials and parts used, they do not need adhesives for assembly, and they reduce the product manufacturing and assembly costs, (Figure 1.6), when enough quantities are produced. Table 1.1 shows a case study comparing the cost of screws to the cost of snap-fits [33]. On the other hand, unless designed with the ease of disassembly in mind and placed at known locations, integral attachments can be hard to disassemble without destruction. Furthermore, low rigidity and low tolerance between mating parts are issues that need to be handled correctly during the design stage of these features.

Table 1.1: Screw cost vs. snap-fit cost for simple two piece enclosure [33]

	Screws	**Snap-fit**
Additional tooling cost (due to complexity)	Negligible	$32 000
Labor cost for assembly	$0.08/unit	Negligible
Screws Cost	$0.05/unit	Negligible
Total Cost	$(0.13× # units)	$32 000
Breakeven quantity	≈ 250 000 units	

Reversible snaps have been used in a vast number of electronic equipments, for example, battery covers for CD players and remote controls, Figure 1.7. Due to their low rigidity, such snaps have only limited use to such small sized appliances because they are simple to unlock.

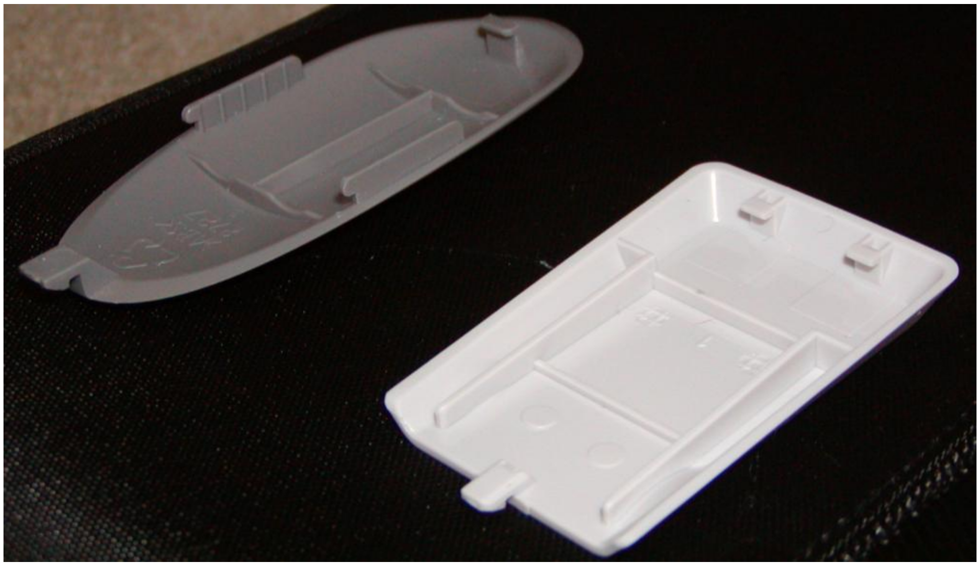


Figure 1.7: Remote control covers utilizing locators and reversible snaps

1.3 Thesis Goal

Given the thesis motivations, this research aims to achieve the following goal:

Develop a general computational method for designing the fail safe "lock-and-key" heat-reversible locator-snap systems that allow non-destructive and clean separation between components for general purpose applications.

The above goal has the following keywords that are discussed in detail below:

- **Fail safe:** The resulting locator-snap system has to ensure that it is not prone to accidental disassembly under normal working conditions. Products are used in different working conditions with varying temperatures, different applied static and dynamic loads, *etc*. The locator-snap system should have the same performance without being accidentally unlocked for any normal reason.

- **Lock-and-Key:** The lock-and-key concept not only ensures that the resulting design is fail safe, it also requires that enclosures do not easily disengage by preventing personnel uncertified to do any maintenance/repair operation. Lock-and-key fasteners easily disengage if the right procedure is followed. The lock-and-key concept is similar to a security code, or a pin number, that has to be entered correctly or you will not be granted access.
- **Heat-Reversible:** Since snaps need to be as stiff as possible, double tapered snaps (Figure 1.5b) cannot be used as they are prone to accidental assembly, while single snaps that require mechanical tool/force for disassembly, (Figure 1.5c and d) affect the aesthetic appeal. Thus the proposed locator-snap system uses single tapered snaps and heat provides the required displacement for disassembly; this way the stiffness and aesthetic appeal are not affected.
- **Clean Separation:** This is to ensure that there is no residue from one material on the other during disassembly. Residues of incompatible materials affect the recycling quality and can prevent the closed-loop recycling needed to produce the same grade recycled material.
- **Locator-Snap System:** The locator-snap system is a system of joints utilizing locators and snaps as the joining method between two different components. It is the system followed in this research since it meets DFD guidelines due to the following properties:

- Locators and snaps are usually made of the same material as the product itself and are usually integrally attached to the same part. This minimizes the number of materials and the number of parts used, and improves recycling quality.
- If the following keywords (fail safe, lock-and-key, and reversible) are achieved, the resulting locator-snap system will allow easy and non-destructive detaching between mating parts at a desirable time, and makes the disassembly process more economic.

1.4 Approach

The concept of the heat-reversible locator-snap system is a system of locators and snaps attached on the internal surfaces around the mating line of a thin-walled part. While assembled, the snaps and the locators engage with the corresponding catches and protrusions molded on the mating part, thereby constraining their relative motions. During assembly, the elasticity of the thin-walled parts is exploited to enable the snapping action, (Figure 1.8). During disassembly, in-plane thermal expansion – constrained by locators – and the temperature gradient along the wall thickness are exploited to realize the out-of-plane bulging of the enclosure wall that releases the snaps, (Figure 1.9).

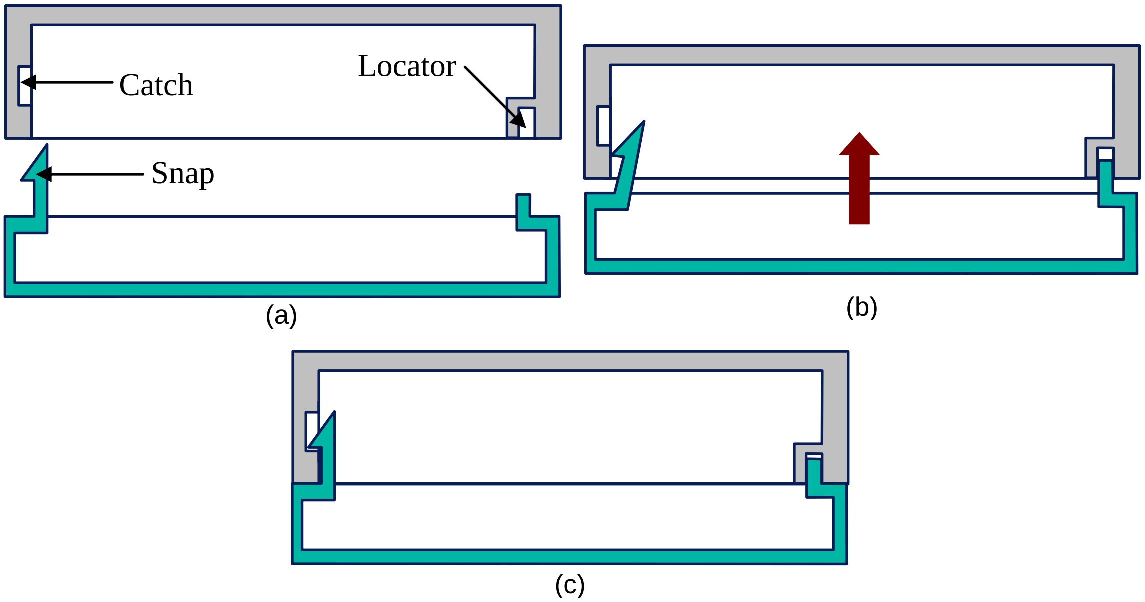


Figure 1.8: Assembly of heat-reversible locator-snap system: (a) before assembly, (b) push, and (c) lock.

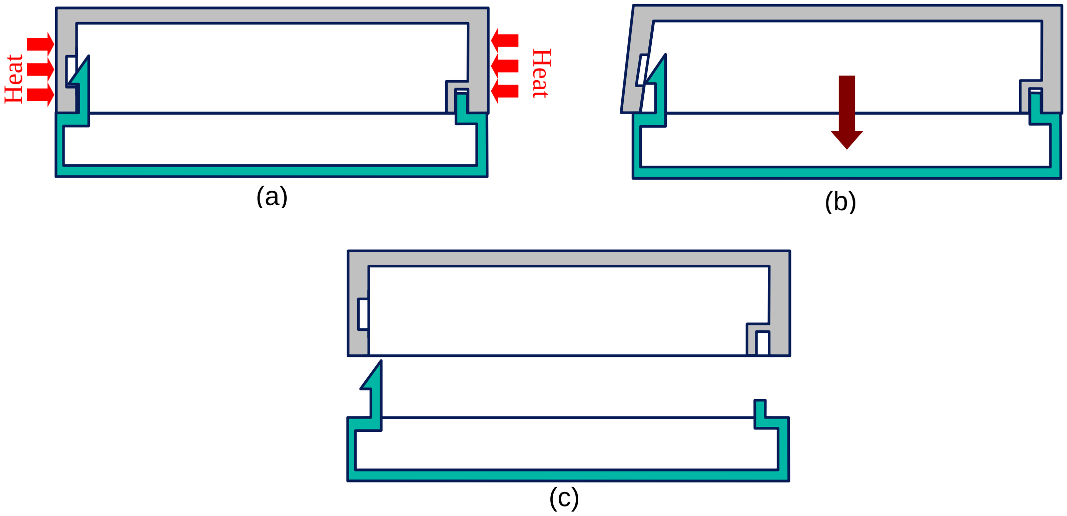


Figure 1.9: Disassembly of heat-reversible locator-snap system: (a) heat the assembly, (b) pull apart, and (c) unlock.

In order to achieve a high-stiffness, heat-reversible locator-snap system that fulfils the desired goal, certain parameters need to be adjusted to realize the release of snaps with minimum heating area and maximum stiffness while satisfying any motion and structural requirements. First, we need to identify the orientations, numbers, and locations of locators and snaps. In addition, the number, locations, and sizes of the heating areas need to be identified as well. It is obvious that the problem involves multiple unknowns and objectives that cannot be obtained using single closed-loop formulation. As a result, this research proposes a computational method for high-stiffness heat-reversible locator-snap systems. To achieve the thesis goals, the following approach is applied:

- Identify the geometry of the two mating parts that need to be joined and define the mating line between both parts.
- Develop a model utilizing the Screw Theory to identify the feasibility any given set of locators and snaps that are examined during optimization. A set of locators and snaps is considered feasible if the joined parts are under-constrained before snap engagement, while they are fully-constrained after snap engagement.
- Develop a generic optimization problem based on the proposed method to find the optimum orientations, numbers, and locations of locators and snaps, and the numbers, locations, and sizes of the heating areas. These variables should realize the release of snaps with minimum heating area and maximum stiffness while satisfying any motion and structural requirements.
- Develop a parallel version of a multi-objective optimization algorithm [34], [35] (namely parallel Genetic Algorithm) that can be utilized to solve the optimization problem efficiently using load balancing.

1.5 Organization of the dissertation

This dissertation describes the following achievements with the aim of developing a general computational method for designing the fail-safe lock-and-key heat-reversible locator-snap systems:

- Chapter 1 introduces the heat-reversible locator-snap system design problem and the motivation behind it, as well as an overview of the adopted approach.
- Chapter 2 presents a review of the relevant literature concerned with the analysis and design of snap-fits, plus the concept of Design for Disassembly using reversible joints. It also reviews the use of Screw Theory in motion and constraint analysis. Finally, it presents a review of the relevant literature for parallel genetic algorithms.
- Chapter 3 describes the design concept and method of lock-and-key, heat-reversible locator- snap systems.
- Chapter 4 discusses in detail a case study of an automotive fender/frame assembly.
- Chapter 5 investigates in-depth two case studies of heat-reversible joints applied to electric consumer products.
- Chapter 6 comprehensively examines the proposed actual implementation of the parallel GA algorithm with load balancing that is needed to solve large-scale and time-consuming optimization problems. A case study that uses the proposed algorithm in the optimization of heat-reversible locator-snap joints on flat panel TV enclosure is discussed.
- Chapter 7 addresses the contributions and recommended future work.

Chapter 2: Related Work

In this chapter, the literatures related to the research presented in this dissertation are reviewed. The work on this research belongs to the area of Design for Disassembly (DFD) and the design and use of snap-fits. First, Design for Disassembly is discussed. Since Design for Disassembly is a broad design methodology that includes many sub-categories, the review in this dissertation covers some aspects of the general Design for Disassembly concept. This review is biased towards comparing DFD and Design for Assembly (DFA), plastics disassembly, disassembly sequence planning, and product embedded disassembly. A more detailed review on the Design for Disassembly using reversible joints is then discussed, which is closely related to the thesis. Second, a review on the analysis and design of the joint used to assemble parts in this research, namely snap fits, is discussed. This chapter further gives the reader a review of the tools used to achieve the dissertation goal. The first tool used is the Screw Theory, which is utilized to analyze the relative motion and constraint status of the joined parts. Section 2.3 gives a review of the usage of the Screw Theory in motion and constraint analysis. A load balanced Parallel Genetic Algorithm is developed during this research to be used as a tool to solve the generic optimization problem generated; thus a review on the current work in that area is also discussed in Section 2.4.

2.1 Design for Disassembly

Disassembly is a critical requirement in expanding the life span of products as it is the first step in the product's service and repair cycle. It is also an important step towards product recycling and re-manufacturing at the end of products life cycle, Sections 1.2.1 and 1.2.2. As a result proper disassembly is required to reduce the service costs and maximize the recycling efficiency [25], [36]. Design for Disassembly (DFD), a powerful design methodology recently receiving researchers' attention, has been proposed to allow the design of a product while improving product disassemblability for component repair, reuse and recycling at the end of the product life. Different manufacturers have been considering DFD at the design stage of their products [37], [38], [39].

Due to the similarity between the DFA and DFD, researches initially suggested the application of DFA methodologies to DFD. Initial work on disassembly time estimation have been developed based on the assembly time estimation methods [36], [32]. However, a number of researchers pointed out that products designed for easy assembly do not necessarily facilitate easy disassembly [1], [18]-[23]. Boothroyd and Alting [17] have an extensive review on the research efforts in the fields of DFA and DFD and their relationships. Additionally, they pointed out the importance of the design of products for the ease of disassembly as a method to overcome the increase in the abandoned products that is currently of high environmental concerns. Lee and Gadh [12] indicated the differences between assembly and disassembly motions when destructive disassembly techniques are utilized. They, in addition, developed an algorithm to simulate the disassembly process directly from the CAD models. Tavakoli *et al*. [40] did a comparison between DFA and DFM guidelines and concluded that the main generic difference between both methods is in the way they deal with joint types. They also recommended that both DFA and DFD tools have to be integrated with the CAD tools to be used in the design process.

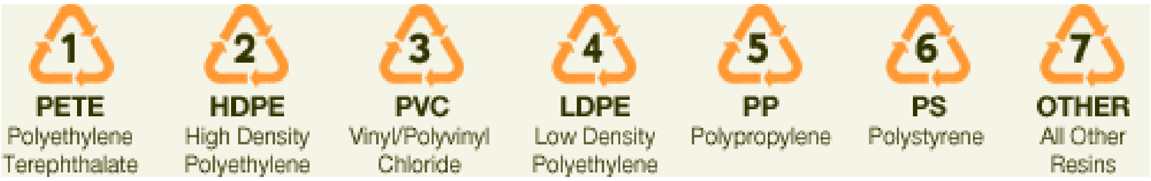


Figure 2.1: Examples of ISO labels for polymers.

Rios *et al*. [5] used discrete event simulation to compare the impact of the current mixed plastics recovery verses thorough plastics identification and separation. They suggested that disassembly and whole part identification can produce substantial yields in the amount of recovered thermoplastics. They also pointed out that disassembly should not be a bottleneck in the whole recycling process. Masanet *et al*. [41] studied the effectiveness of the design for recycling guidelines on plastic components and the results of their findings are summarized in the list below:

- The use of ISO labels (Figure 2.1), used to identify the type of polymer used in the plastic part during the sorting process, is a highly effective strategy; yet, it is important to periodically ensure that they remain accurate.

- The use of one polymer or one color for each polymer for large items is not an efficient strategy for large recycling facilities that recycle different batches from different manufacturers together as the resulting batch will remain heterogeneous.
- Limiting the use of paint to less than 1% helps in improving plastics recyclability.
- Molding and gluing on metal part is only problematic for manual disassembly systems, but not in automated disassembly such as shredding and grinding of batches.
- Current snap-fits are not recommended in manual disassembly as the laborer has to put the tool down each time a snap is removed. In automated disassembly, destructive separation is used which does not differentiate between screws or snap-fits.

It can be deduced from the above list that the use of reversible snap-fits that can easily disengage by a single triggering action can help improve the recyclability of plastics as it will become faster and easier in manual disassembly. It will also reduce amount of metallic scrap in plastics; thus improve the recycled quality of plastics. Reap *et al.* [42] suggested robotic semi-destructive disassembly, and recommended the use of easily detachable or breakable snap-fits over the use of screws due to their ease of disengagement as an appropriate DFD method.

Most of the research on DFD concentrated on the issues of sequence planning, disassembly evaluation, and analysis and product recovery to improve the economics of disassembly [43], [44]. Johnson and Wang [45] discussed the influence of the disassembly sequence on the economics of disassembly. In Figure 2.2, the upper curve shows the disassembly sequence in the descending order of the recovered components' value, while the lower curve shows the disassembly sequence when removing the least valuable components first. The disassembly cost is presented by the middle line. It is obvious that it is important to disassemble components with highest recovery value first until a breakeven point is reached. Viswanathan and Allada [43] proposed a Configuration Value (CV) model to evaluate and analyze the effect of configuration on disassembly.

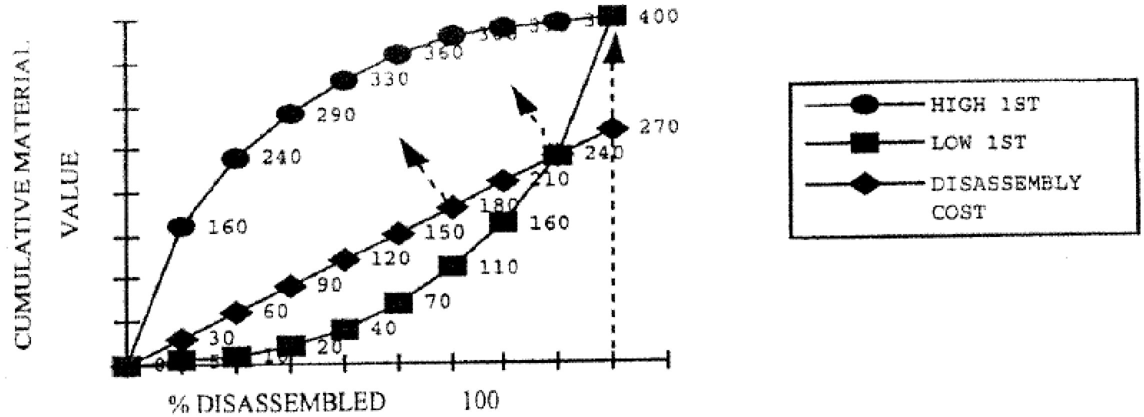


Figure 2.2: Effect of disassembly sequence on the disassembly revenues [45].

The disassembly evaluation chart was proposed by several researchers [47] and [48] to serve as a tool to evaluate the ease of disassembly. Das *et al.* [49] presented a multi-factor model to evaluate the Disassembly Effort Index (DEI) score as a measure of the ease of disassembly. Shetty *et al.* [28] presented quantitative measures and procedures to evaluate the ease-of-disassembly of products during recycling. Hulla *et al.* [50] developed a multi-criteria decision making methodology to determine the optimal EOL disassembly scenario from both, environmental and economical perspectives. Desai and Mital [51] and [52] developed a scoring method by assigning time-based numeric indices to each design factor that affects the ease of disassembly. The flowchart of the method is given in Figure 2.3.

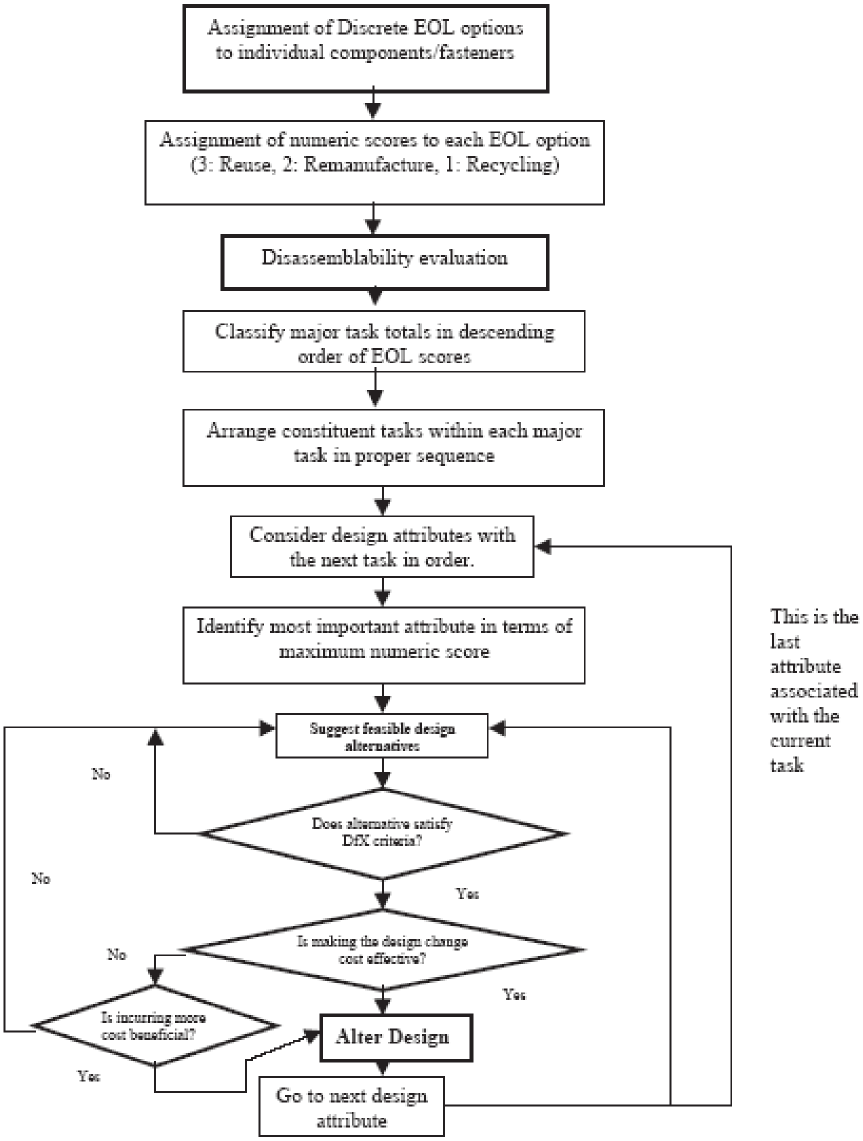

Assignment of Discrete EOL options to individual components/fasteners
Assignment of numeric scores to each EOL option (3: Reuse, 2: Remanufacture, 1: Recycling)
Disassemblability evaluation
Classify major task totals in descending order of EOL scores
Arrange constituent tasks within each major task in proper sequence
Consider design attributes with the next task in order.
Identify most important attribute in terms of maximum numeric score
Suggest feasible design alternatives
No
Does alternative satisfy DfX criteria?
No
Yes
Is making the design change cost effective?
No
Yes
Is incurring more cost beneficial?
Yes
Alter Design
Go to next design attribute
This is the last attribute associated with the current task

Figure 2.3: Flowchart for DFD using Desai and Mital scoring method [51].

The practicality of the disassembly process during disassembly has also been covered. O’Shea *et al.* [53] used dynamic programming to find the optimal tool selection path during disassembly. Sodhi *et al*. [8] proposed a U-effort model to help designers select the type of fastener that has the easiest and most economic unfastening action. The logical flow on which U-effort index evaluation is based on is given in Figure 2.4.

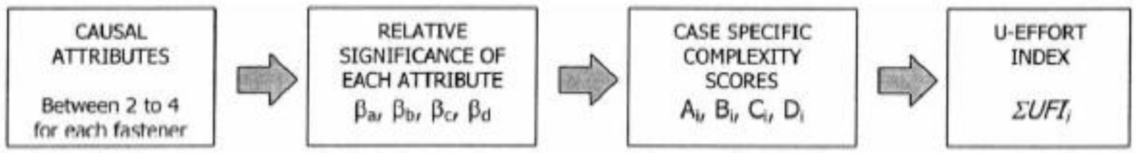

CAUSAL ATTRIBUTES
Between 2 to 4 for each fastener
RELATIVE SIGNIFICANCE OF EACH ATTRIBUTE
$\beta_a, \beta_b, \beta_c, \beta_d$
CASE SPECIFIC COMPLEXITY SCORES
A_i, B_i, C_i, D_i
U-EFFORT INDEX
ΣUFI_i

Figure 2.4: The UFI derivation process [8].

While these works study the improvement of the disassembly process through planning the tools used and the disassembly sequence, and maximizing the ease of disassembly while maximizing the recycling revenues, they do not address the very primary issue in disassembly, which is joint selection and design as part of the whole design process during the entire life cycle of the product, while permitting efficient disassembly.

2.1.1 Design for product-embedded disassembly

Masui *et al.* [54] used nichrome wires embedded along the desired boundary of separation, for the active disassembly, Figure 2.5. In their example, the method was applied for disassembly of Cathode Ray Tubes (CRT), where thermal stresses crack the glass upon the application of electrical current to the embedded nichrome wires. Although, destructive disassembly was justifiable in this case, the method cannot be generalized as there are other cases where destructive disassembly is not desired.

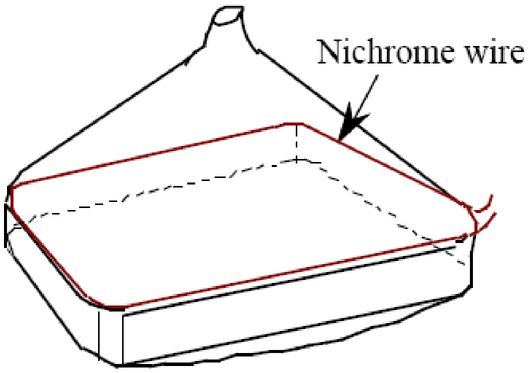


Figure 2.5: Example of CRT disassembly using Nichrome wires [54].

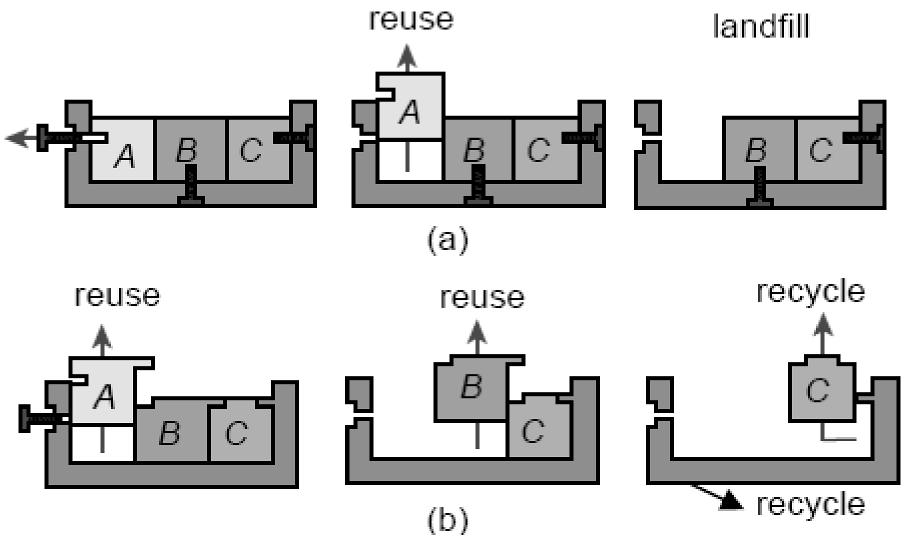


Figure 2.6: (a) Conventional assembly (b) assembly with product embedded disassembly [59].

Takeuchi and Saitou proposed the concept of design for product-embedded disassembly [55]-[60] from a different perspective. This is a new approach to Design for Disassembly that aims at designing products with built-in disassembly means to be activated at the end of product life. The relative component motions are constrained by the locators (tabs, slots, lips, rests, etc.) integrated into the components, in such a way that the removal of one or few fasteners would cause the self-disintegration of the assembly in a unique sequence, much like the domino effect. Figure 2.6 illustrates the concept of product-embedded disassembly. In Figure 2.6a, three components *A*, *B* and *C* are fixed to the casing with three fasteners. In order to fully disassemble all the components, the three fasteners have to be removed, which is labor intensive process. On the other hand, in Figure 2.6b, the motions of components *B* and *C* are constrained by the locators integral to the components and the casing. Since no additional fasteners need to be removed, components *B* and *C* can also be disassembled with the removal of only one fastener, which is less labor intensive. The research on heat-reversible locator-snap systems, presented in this dissertation, can be integrated with the product-embedded disassembly approach. This can be achieved by designing heat-reversible locator-snap systems that act as the fasteners triggering the domino disassembly action.

2.1.2 Design for Disassembly using reversible joints

Easy-to-disengage joints can help reduce disassembly efforts, thus making the recycling process economically feasible. Chiodo *et al*. [61]-[65] incorporated the concept of Active Disassembly using Smart Materials (ADSM), originally developed at Brunel University in 1996, by developing the non-destructive self-disengaging screws and brackets made of a special Shape Memory Polymer (SMP). Shape Memory Alloys (SMA) and SMP behave like standard engineering materials while below the transformation temperature. When heated above this critical temperature, SMA and SMP undergo shape change to their original shape. This behavior is called Shape Memory Effect (SME), Figure 2.7. Screws manufactured using SMP are produced so that they lose their threads when heated above the transformation temperature, thereby allowing disassembly. Compression springs act as actuators for the eventual active disassembly. Upon heating compression springs actively dismantle the assembly as SMP screws regain their original shape.

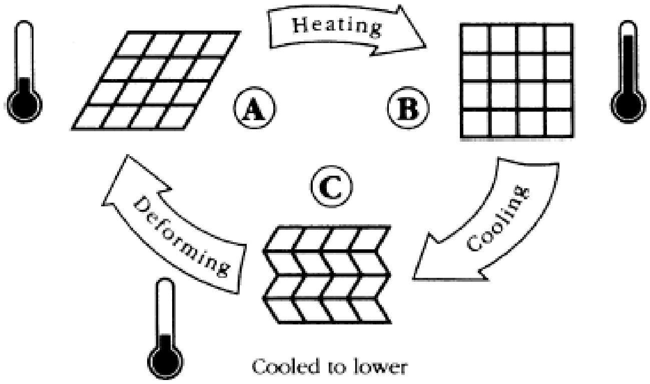


Figure 2.7: The Shape Memory Effect principle in SMA and SMP [65].

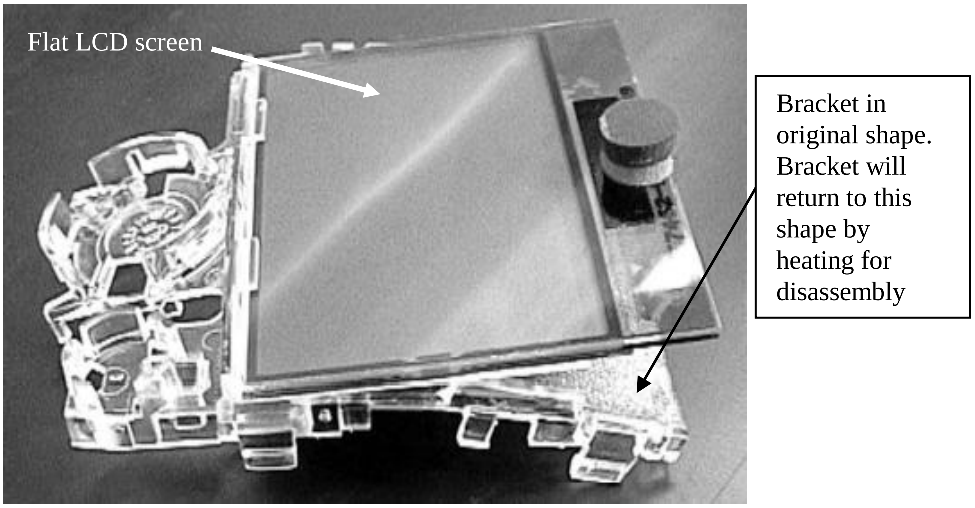
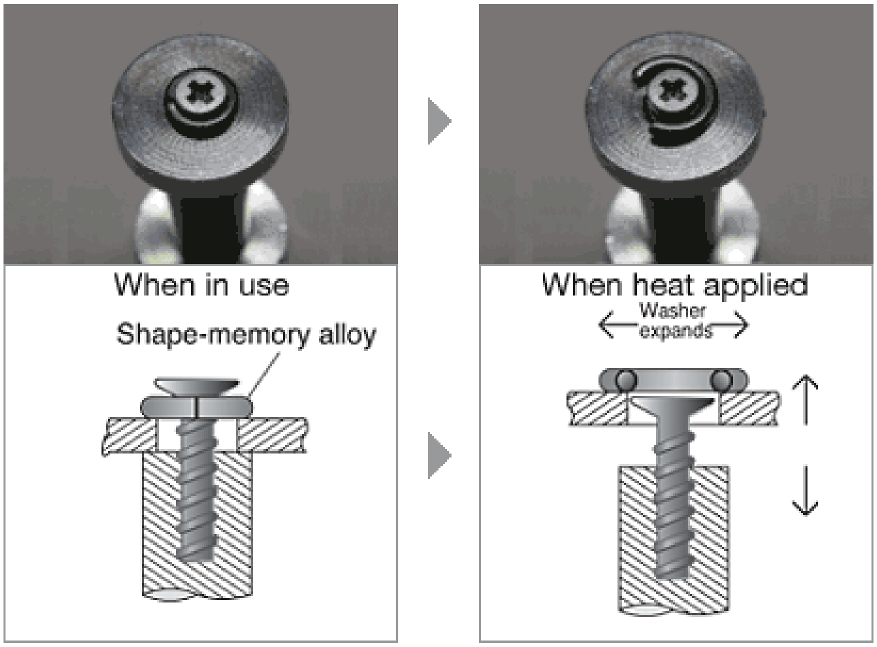

Flat LCD screen
Bracket in original shape. Bracket will return to this shape by heating for disassembly

Figure 2.8: Manipulated polycarbonate LCD bracket used for SMP bracket-molding cavity. The position of the LCD screen indicates the intended SMP bracket configuration after SME transformation [64].

The method was applied on different electronic equipment, CD players, cellular phones, LCD display, etc. Figure 2.8 shows an example of SMP bracket. The bracket has the original bent configuration shown in Figure 2.8. The bracket then undergoes shape memory transformation to meet the manufacturer's original specifications (being flat not curved). During disassembly, the bracket undergoes shape memory transformation, due to heating to return to its original shape, thus releasing the LCD screen.

For review on active disassembly using SMP refer to Arnaiz *el al.* [66]. Jones *et al*. [67] later used residual electrical energy in batteries to generate the energy needed for disassembly. While the results seemed effective in the particular cases presented, regardless of the failures in some experimental trials, the method lacks generality since it requires the use of specialized and costly materials such as SMP. In addition, each screw needs to be heated to a desired temperature which makes the process energy intensive. Finally, although springs are used to allow active disassembly, they keep SMP screws loaded all the time. As a result, creep failure analysis has to be considered.



When heat is aplied,the shape-memory alloy washer expands and causes the screw to come loose

Figure 2.9: Shape Memory Alloy washer concept [69].

Sharp Corporation, NEC Tokin Corporation, Union Seimitsu Co., Ltd., and Tokai University developed washers employing TiNi shape-memory alloy, Figure 2.9. The washers remain securely fastened while in use but expand when heated, releasing the head of the screw and enabling the two combined pieces to come apart easily [68], [69]. The method was applied on LCD TVs and the results were promising.

Li *et al.* [70], used topology optimization [71] and compliant mechanisms [72] to design reversible integral attachments (snap-fits) that can be detached by the application of localized heat. A metallic Thermal Force Applicator (TFA), integrated with an engaging snap, is heated and the resulting thermal deformation induces the release of the snapped joint, through the transmission of the deformation of TFA to the snap. Later, the researches utilize the localized heat without TFA, for better deformation characteristics [73]. However, because the unlocking motions of these snap designs rely solely on the local thermal deformations of the snap. The deflection to height ratio is 0.005 ~ 0.01, this results in opening actions that are too small for practical applications. Finally, they use variable thermal boundary conditions [74], a heating zone and a cooling zone. Figure 2.10 shows the geometry resulting from topology optimization with the heating and cooling zones indicated. Figure 2.10 (b) has magnification factor equal to 2.5. Although, the deflection to height ratio has a value of 0.025, this value is still below practical implementations. In addition, the snap geometries generated through topology optimization cannot be easily injection molded. Finally, they focus only on the design of a heat-reversible snap rather than handling the entire assembly, and the locators and snaps as a system to achieve the system level objectives.

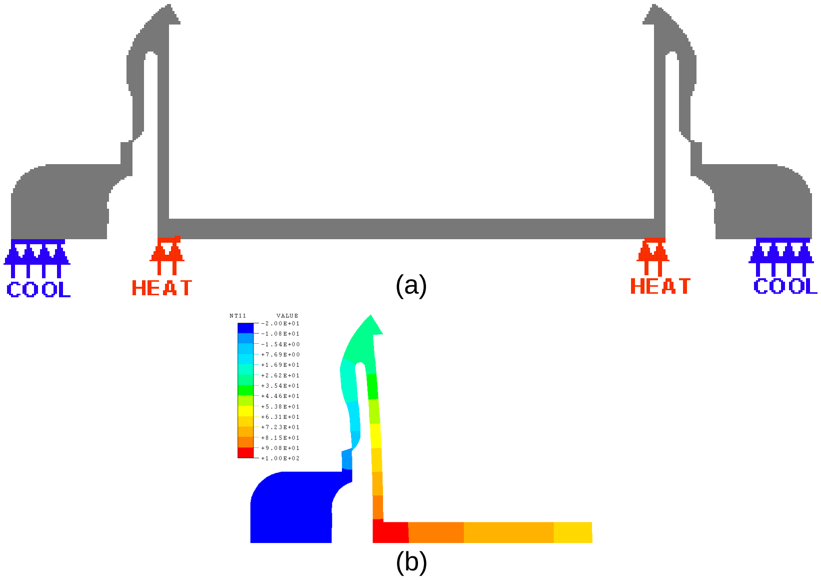


Figure 2.10: Use of variable thermal boundary conditions: (a) topology optimization result, (b) snap deflection [74].

For more information about different active disassembly methods refer to Willems *et al*. [75]. The new heat-reversible locator-snap system presented in this research has the following advantages over the current reversible joints presented in this literature review:

- Unlike regular and shape memory screws, snap-fits are simply manufactured as an integral attachment on the mating parts using the same materials as the product itself. This minimizes the total number of parts; thus minimizing the production cost of the whole product and ease the recycling process make it more economical as the number of different materials at the recycling stage is minimized.
- Compared to snap-fits generated from topology optimization, the proposed heat-reversible locator-snap system basically has simpler and standardized geometry, which directly impacts the overall product cost as they require simpler mold designs.

- Heat-reversible locator-snap system presented in this research considers the snaps, locators and the geometry of the enclosure in the design process to achieve assembly and disassembly actions with high stiffness. It utilizes the deformation of one or both of the mating parts as a whole. Since the design method has the flexibility to locate the snaps at regions of maximum thermal deformation, the overall deformation achieved is much larger than that of cantilever snaps. At the same time, the stiffness of the overall product is comparable to that of a product that uses screws for assembly. The work on designing snap-fits using topology optimization focuses only on the snap shape, geometry and performance, not the entire locator snap system as in this dissertation.

2.2 Analysis and design of snap-fits

Snap-fit is the joint type implemented in this research to join different components. This section reviews the research done on the analysis and design of snap-fits.

Snap-fit is a preferred joining method for DFD for the following reasons [28], [75]-[77]:

- Have Easy assembly action
- Can be disassemblable
- Reduces overall product cost
- Does not need extra parts for separation
- Makes the recycling process more economic

- Provides clean separation between different components.

Early work on integral attachment design focused on the analysis of particular types of locking features such as cantilever hooks [78], bayonet-fingers [79], compressible hooks [80], *etc*. More recently, Genc *et al.* [81]-[83] discussed a feature-based method to integral attachment design, which classified snap-fit features into three categories:

- *Locating features*: eliminate the degrees of freedom between parts, transfer the service loads, and identify relative positioning between parts, as shown in Figure 2.11.
- *Locking features*: provide final locking between parts through elastic deflection. Examples of locking features are given in Figure 2.12.
- *Enhancing features*: such as guides to ease assembly and visuals to provide information on integral attachment location to ease disassembly, as shown in Figure 2.13.

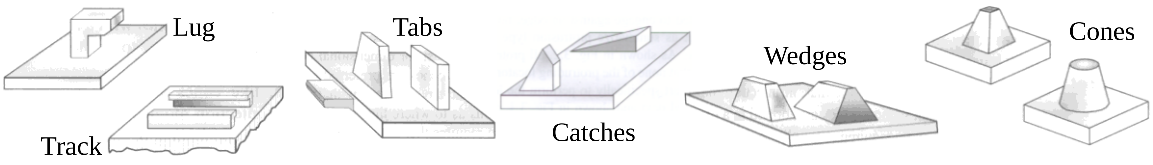


Figure 2.11: Examples of locating features [84].

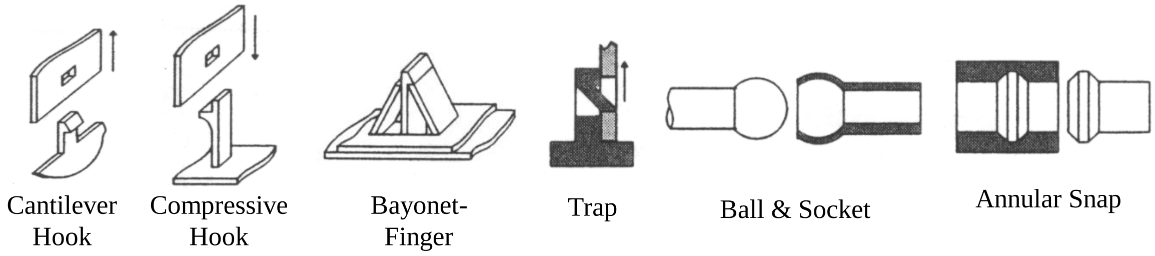


Figure 2.12: Examples of snap-fit types (locking features).

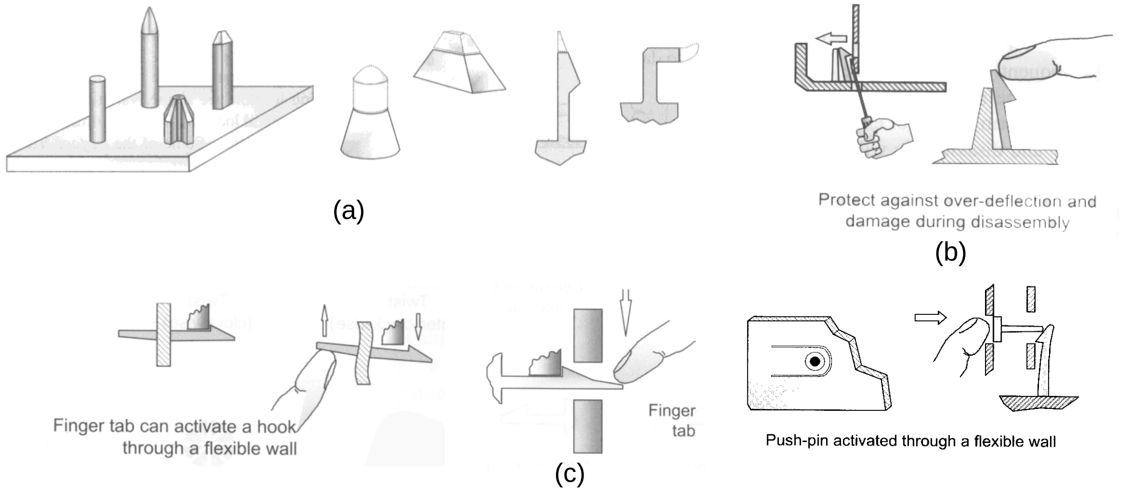

(a)
Protect against over-deflection and damage during disassembly
(b)
Finger tab can activate a hook through a flexible wall
Finger tab
Push-pin activated through a flexible wall
(c)

Figure 2.13: Examples of Enhancing features: (a) Assembly enhancers (guides), (b) Guards to protect against damage, and (c) Disassembly Enhances [84].

Luscher *et al.* [85] discussed a similar classification based on assembly motions, namely push, tip, slide and twist. A combination of these motion types has to satisfy four functions, namely approach, location, compliance and locking.

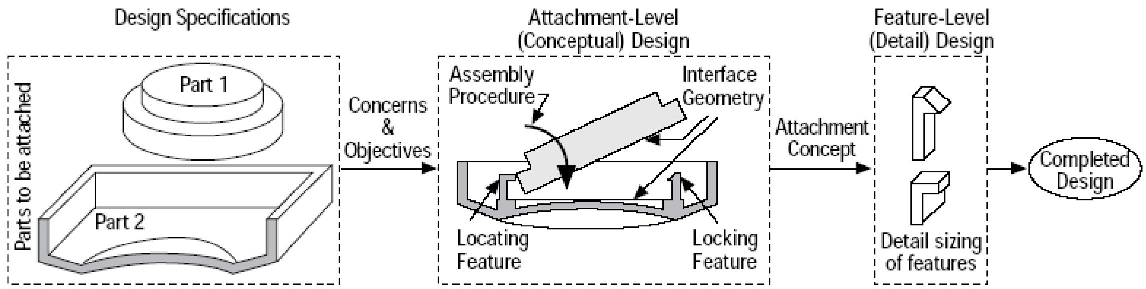


Figure 2.14: Different stages in the integral attachment design process [90].

Messler, Genc and Gabriele published a seven-part series of articles discussing the use of snap-fits as a key to assembly automation. Part 1 [86] serves as an introduction to the concept of integral attachments and defines its key terms. They divided the integral attachment design process into two levels, 1) the well established feature level design, which studies the elastic behavior of the attachments and is based on the mechanics of materials concepts. 2) Attachment level design [87], which considers integral snap-fit assembly as a system of compatible constraint and enhancement features that form a reliable and robust mechanical attachment between components. Figure 2.14 has as example of these two levels. They proposed a complete design methodology for the design of snap-fits that is composed by using four major tools (methods):

- A six step attachment level design methodology [88] to identify the compatible constraining and enhancing features, Figure 2.15.
- A six step locking feature selection methodology, Figure 2.16, which identifies the feasible locking features [89].
- A procedure to fully constrain parts using locking and locating features [90].
- A tool to quantitatively compare all the feasible designs obtained using the above tools [91].

Finally the whole design methodology is tested on a hand set speaker assembly and the results were presented in [92].

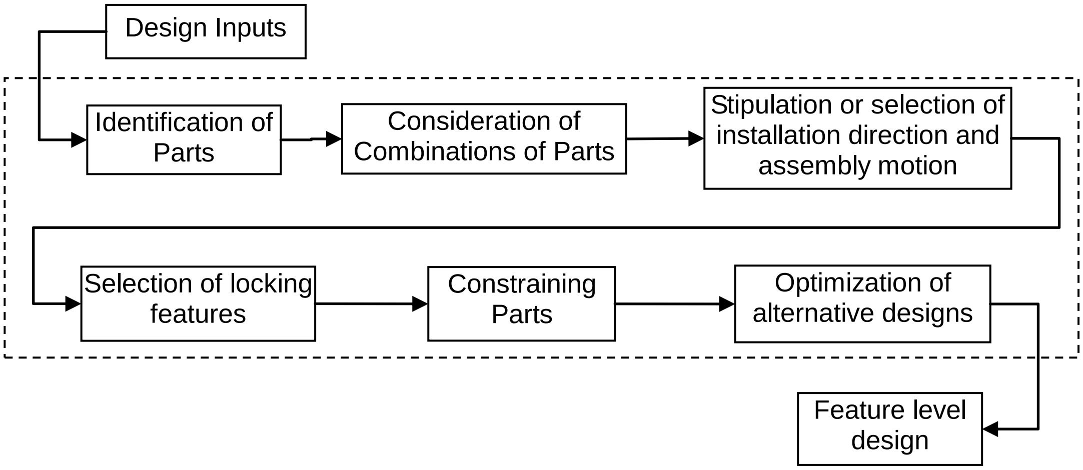

Design Inputs
Identification of Parts
Consideration of Combinations of Parts
Stipulation or selection of installation direction and assembly motion
Selection of locking features
Constraining Parts
Optimization of alternative designs
Feature level design

Figure 2.15: A six-step attachment-level design methodology proposed in [88].

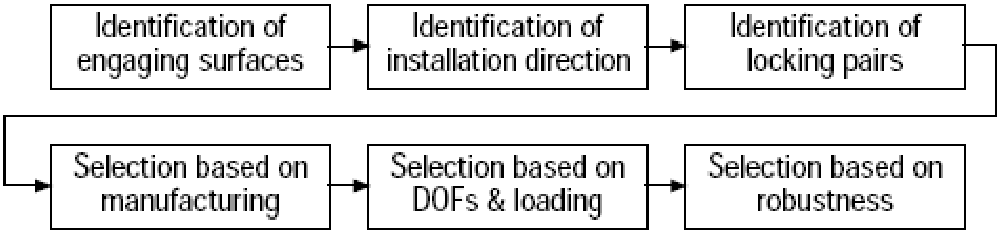

Identification of engaging surfaces
Identification of installation direction
Identification of locking pairs
Selection based on manufacturing
Selection based on DOFs & loading
Selection based on robustness

Figure 2.16: A Six-step methodology for locking feature selection proposed in [89].

Bonenberger collected ongoing research on that field in a reference and design handbook [84] that can serve as a good start in understanding and organizing the snap-fit technology. This work, however, did not address the reversible snap-fit designs that are actuated by thermal deformation.

2.3 Screw Theory in motion and constraint analysis

This section gives the reader a review of the usage of the Screw Theory in motion and constraint analysis. The screw theory is utilized as tool in this research to analyze the relative motion and constraint status of the joined parts.

The Screw Theory, a pioneering work by Ball [93], is a universal analytical method used for motion and constraint analysis of rigid bodies. Since then, Screw theory has been applied to areas of mechanism, robotics and machine design. Among others, Waldron [94] utilized the Screw Theory to build a general method to determine all relative degrees of freedom (DOF) between two rigid bodies making contacts to each other. Ohwovoriole and Roth [95] extended the Screw theory by introducing "repelling and contrary" screws, which are useful in the mathematical modeling of the parts assembly process. Sugimoto and Duffy [96] analyzed screw systems using linear algebraic methods.

Konkar and Cutkosky [97] developed a recursive algorithm to compute motions allowed by the mating features within mechanisms. Based on that algorithm, Adams and Whitney [98], [99] developed a method to determine the status (over-, under- or fully constrained) of rigid body assemblies with mating features. Their method also determines the motion type and range of under-constrained rigid body assemblies. Lee and Saitou [100] applied Adams and Whitney's method for automatically synthesizing 3D assemblies with prescribed in-process dimensional adjustability. In this research Screw Theory is used to analyze relative motion constraints on the two mating parts imposed by locators and snaps of given orientations.

2.4 Parallel Genetic Algorithms

A load balanced Parallel Genetic Algorithm (GA) is developed during this research to be used as a tool to solve the generated generic optimization problem; thus this section reviews the current work in that area.

GA was first introduced by John Holland in 1975 [101]. GA is a heuristic optimization method that mimics the evolution of living creatures by implementing the idea of the survival of the fittest. Different Multi Objective Genetic algorithm (MOGA) versions were introduced in the last decades to solve problems with multiple competing objectives. Solving such problems with evolutionary algorithms, like MOGA, has an advantage over other optimization methods. Due to the population-based nature of MOGA, it can capture several points on the Pareto Optimal Set in just one run. Generally speaking MOGA is a favored optimization method due to its ability to deal with large numbers of continuous and discrete variables simultaneously; it is also capable of dealing with highly multi modal discontinuous objective functions to find near global optimal solutions.

GA is very resource intensive and requires large number of objective function evaluations to converge, similar to all heuristic based methods. If the time to evaluate the objective function is long, the speed of the current workstations may not be enough to finish the optimization process at a desired time. As a result, the need for parallel versions of GA is becoming necessary.

Branke *et al* [102] proposed a method to find the Pareto optimum by dividing the search space over the processors using cone separation and apply separate MOGA to get the Pareto for each sub-space then merge them together. Hiroyasu *et al* [103] suggested a divided range MOGA, where the initial population is divided into subpopulations, based on the fitness value of a focused objective function that changes each time. Subpopulations run on different computers for a while. Finally the whole Pareto is then generated and is divided again over the processors and so on. Quagliarella and Vicini [104] used a subpopulation parallel MOGA model, sometimes called island model, and applied it to the wing design problem. Different variations of global manager-worked models, where the manager holds the whole population while workers just evaluate the objective functions, have been reported in literature. Stanley and Mudge [105] proposed an asynchronous model that simply allows a new generation to start being evaluated before previous generations are complete. Okamoto and Zhao [106] developed a synchronous model, where the mutation and crossover operations are performed by the worker processors. It can be seen that all the above models targeted load balancing by dividing the population or the design space equally over the available number processors. None of these works addressed the load imbalance occurring from the variation in the objective function evaluation time, which is not common in typical optimization problems. The variation in the objective function evaluation times may occur when the function used to evaluate the objectives is iterative or incremental. For instance, numerical solutions for differential equations, as in the case of Finite Element Analysis, fluid dynamics, *etc.*, usually use iterative and/or incremental schemes to reach a solution. The number of iterations or increments can vary significantly as the differential equations' parameters vary; thus the objective function evaluation time varies as well. Although the asynchronous MOGA model can be of benefit in that sense, starting a new generation without actually finishing previous generations can affect the probability of choosing the elitist chromosomes.

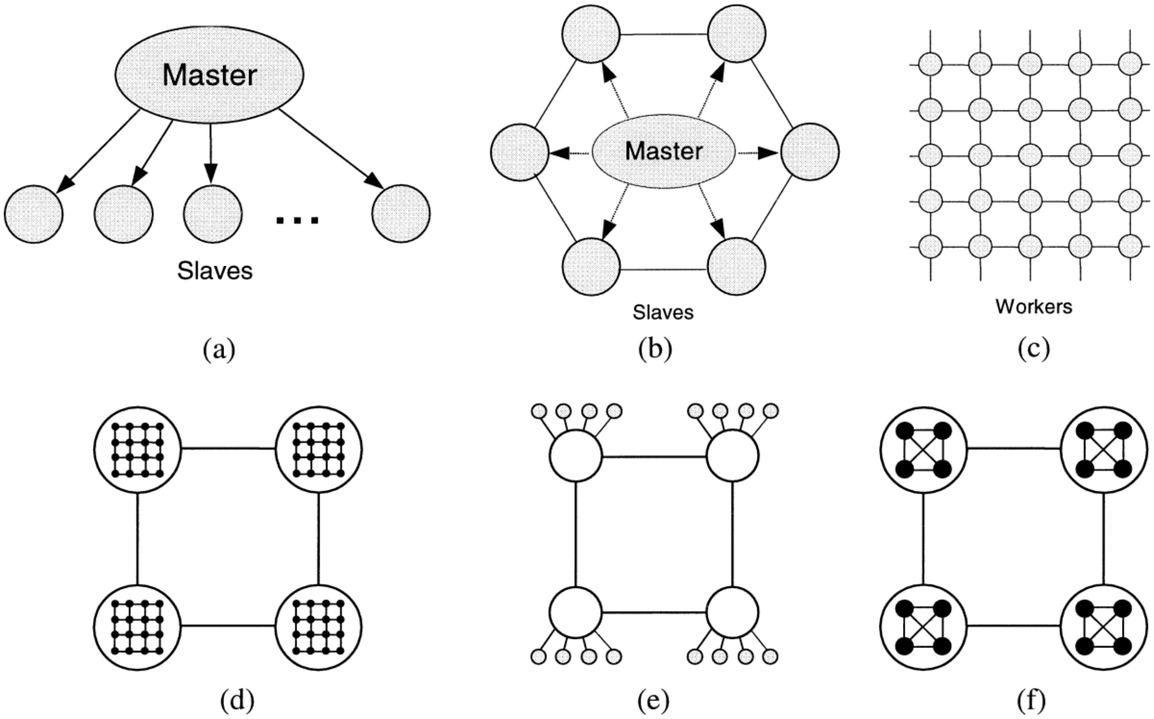


Figure 2.17: Different models of PGA: (a) global parallelization, (b) coarse grain, and (c) fine grain. Many hybrids have been defined by combining PGAs at two levels: (d) coarse and fine grain, (e) coarse grain and global parallelization, and (f) coarse grain plus coarse grain.[107]

Given that the parallelization of MOGA can be modeled in a similar way as parallel single objective GA (SOGA), a review of different parallel SOGA models was done. Different models of parallel GA are shown in Figure 2.17; detailed description of each model is given by Alba and Troya [107]. Adeli and Kumar [108] proposed dynamic load balancing for the global manager worker model by simply allowing each worker processor to evaluate one objective function at a time. Other researchers only considered static load balancing using global models where the population is equally divided on the worker processors. Marco and Lanteri [109] developed a two level parallelization, parallel GA using global manager worker and parallelization of the objective function evaluator, for shape design problems. Again, these methods lack the fact that the manager itself can evaluate the objective function when it is idle. In addition, evaluating a single objective function at a time adds a lot of overhead due to the excessive need of communication between processors.

The island model for SOGA, also called the distributed manager-worker model, had much attention recently. Hong *et al* [110] proposed an island model where subpopulations are developed separately from the start. When equilibrium is reached, migration between populations starts. Aguirre *et al* [111] developed a distributed GA model that implements parallel genetic operators to improve the efficiency. Alba and Troya recommend further study on the effect of migration on distributed GA performance [112]; they also suggest that asynchronous distributed GA is faster than its synchronous counterpart [107]. Cantù-Paz and Goldberg [113] gave a set of guidelines for the choice of parallel GA parameters.

Although a number of researchers implemented different versions of parallel GA as optimization algorithms, it can be seen that load balancing, especially for global parallelization, did not receive much attention. The reason behind that is probably because efficient load balancing techniques may not be needed when the time to evaluate the objective functions it too short or is not varying with respect to the design variables. In this research, the objective function evaluation time can vary significantly. To obtain objective function evaluations values, finite element analysis is required multiple times and the time needed to run a finite element simulation is highly dependent on the values of the design variables. Actually, the time to evaluate the objective functions can vary from milliseconds in case of an infeasible design to few minutes in some feasible cases. As a result, this research focuses on presenting an active load balancing scheme to Multi objective Genetic algorithm to improve its efficiency.

Chapter 3: Heat-Reversible Locator-Snap Joints

3.1 Design concept

There exists a wide variety of fasteners such as, screws, adhesives, snap joints and rivets. Such fasteners are used primarily for fixing two or more mating parts together, they can also be used for transferring loads between parts. Heat-reversible locator-snap systems utilize snaps and locators to perform the same function as normal joints; yet, upon heating at certain locations the snaps unlock to allow a clean disassembly action. Figure 3.1 illustrates the design concept of heat-reversible snap joints showing the locators, snaps, holes, catches, and the mating line.

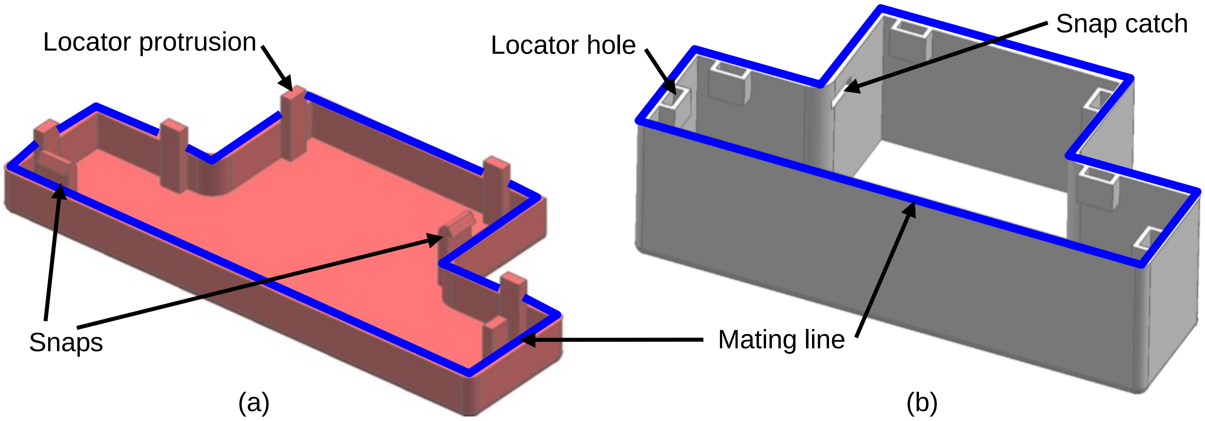
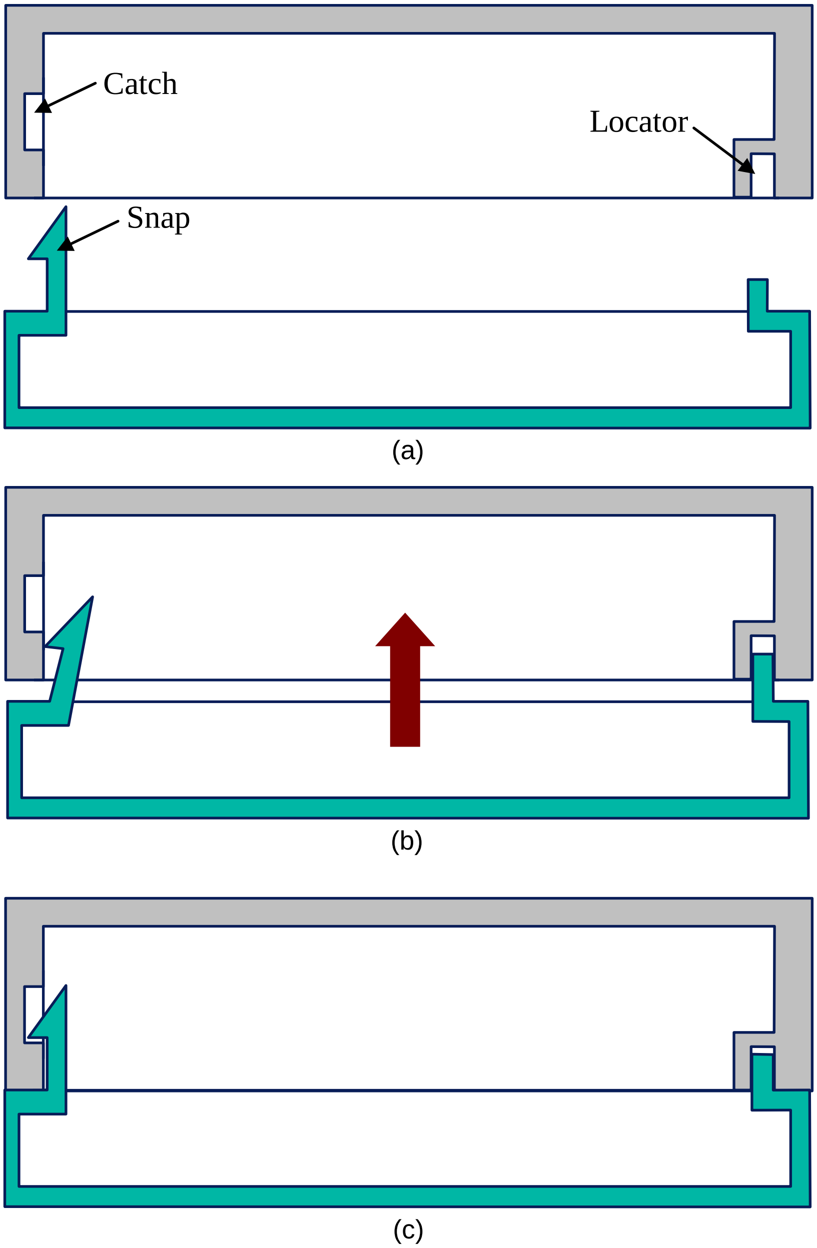


Figure 3.1: Heat-reversible locator-snap system: (a) enclosure top with 2 snaps and 6 locators, and (b) enclosure bottom with snap catches and locator holes.



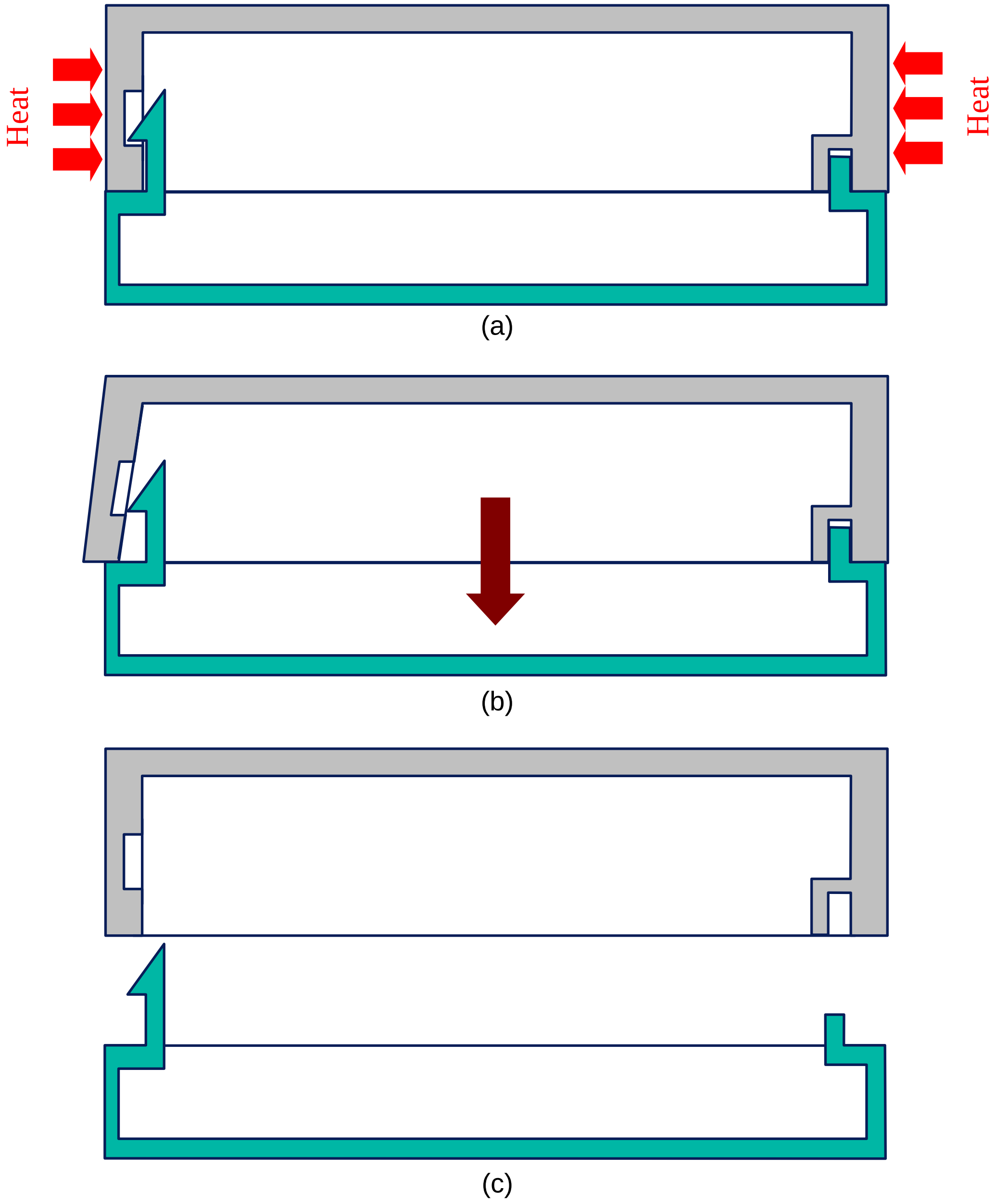
(a)

(b)

(c)

Figure 3.2: Assembly of a heat-reversible locator-snap system: (a) before assembly, (b) push, and (c) lock.

Heat-reversible locator-snap system consists of locators and snaps attached to the internal surfaces around the mating line of a thin-walled part. While assembled, the locators and the snaps respectively engage with the protrusions and the catches on the mating part, thereby constraining their relative motions. Heat-reversible locator-snap system is essentially a conventional locator-snap system found in literature [84]; hence its engagement process is analogous to the conventional locator-snap system engagement process as shown in Figure 3.2. During assembly, the elasticity of the thin-walled parts is exploited to enable the snapping action.



(a)

(b)

(c)

Figure 3.3: Disassembly of a heat-reversible locator-snap system: (a) heat the assembly, (b) pull apart, and (c) unlock.

Since the assembly and disassembly actions require elastic deformation of the enclosure structure, one of the parts needs to be of relatively thin thickness, while the other part can be of any geometry. The elasticity of the thinner enclosure, not the snaps or the locators, is exploited to enable the snapping action. This allows the locators and the snaps to be stiff enough to meet the structural requirements of the joints, compared to elastic cantilever snaps. Similarly, Figure 3.3 illustrates the disengagement of the two mating parts. Upon heat application to the enclosure (Figure 3.3 a), in-plane thermal expansion constrained by locators, and to some extent the temperature gradient caused by heat dissipation, cause out-of-plane bulging of the enclosure. As a result, the snap unlocks from the catch (Figure 3.3 b). Finally, the enclosures can be separated from each other in a reverse manner to the assembly.

3.1.1 Lock-and-key design concept

Locator-snap joint systems are usually less stiff compared to screw fasteners and are prone to accidental disassembly. This limits the usage of this type of joints to specific applications. In this research we are developing high-stiffness locator- snap system that can be used in almost any application. It is desired to ensure that assembly does not disengage accidentally during normal operation. It is also desired to make the disassembly process easy only when the right procedure is used.

The lock-and-key concept ensures that enclosures do not disengage accidentally but disengage easily when the right procedure is followed. The lock-and-key concept is realized by double-latch snaps that require displacement within a certain range to disengage (Figure 3.4), and multiple snaps that require heating multiple locations at different temperatures to disengage. Figure 3.4a represents the double-latch snap concept in the locked position. If the right amount of displacement is applied, both latches will be free and the parts can disengage easily, Figure 3.4b. If insufficient or excessive displacement is applied, at least one latch will remain engaged preventing undesired disassembly, Figure 3.4c and d respectively. The displacement required to unlock the snaps in heat-reversible locator-snap system is induced from heating the enclosures at certain locations at certain temperatures. Thus if the wrong heating areas are selected or the wrong temperatures are used, there will be a high chance that the amount of displacement be either insufficient or excessive; thus, the snaps remain locked.

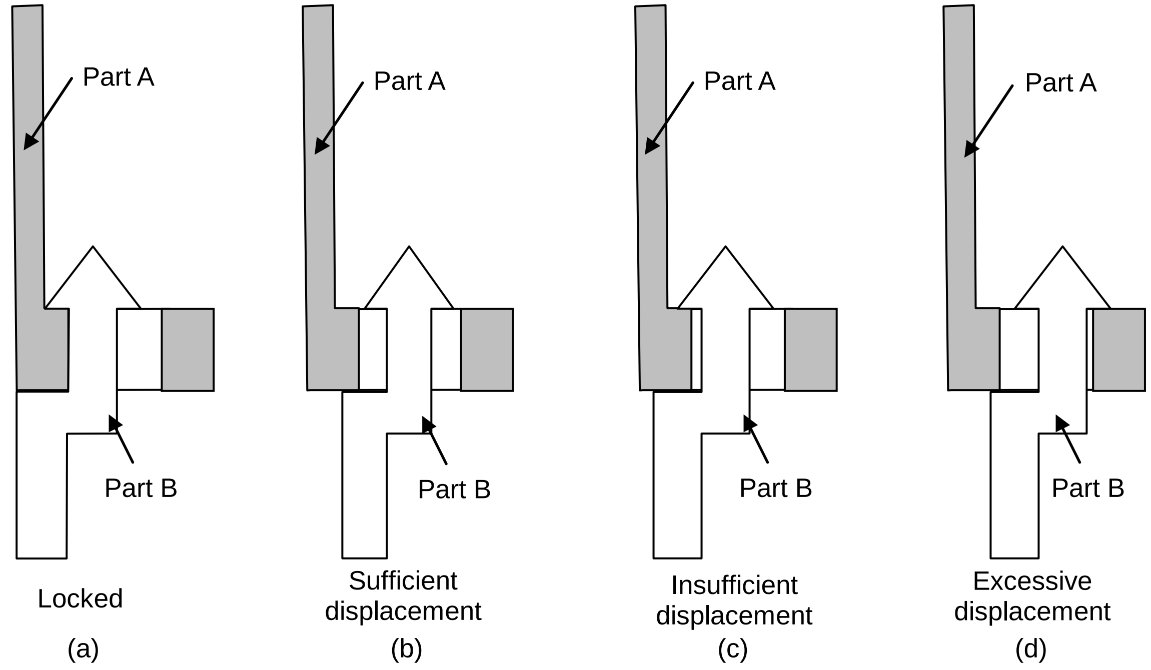


Figure 3.4: Double-latch snap (a) working position, (b) unlocked, (c) insufficient unlocking displacement, and (d) excessive unlocking displacement.

3.2 Design method

The method synthesizes optimal designs of the locator-snap system by solving the following optimization problem:

- **Given:**
    1. The geometry of the two mating thin-walled enclosure parts.
    2. The coordinates of the vertices of the polygon representing the mating line where locators and snaps will be placed.
    3. The feasible regions for heating.
    4. The library of locators and snaps that can be used.
- **Find the Design Variables:**
    1. Orientations, numbers, and locations of locators and snaps.
    2. Locations and sizes of the heating areas.
- **Minimizing the Design Objectives:**
    1. The number of locators and snaps.
    2. The compliance (reciprocal of the stiffness) of the whole enclosure.
    3. The heating area(s).
    4. The tolerance stack-up resulting from over constraining the mating enclosures with multiple locators fixing the same degree of freedom.
    5. Any other problem specific design objectives.
- **Subject to the Design Constraints:**
    1. The parts are under constrained and do not interfere with the neighboring parts before snap engagement.

2. The parts are not under constrained and meet the structural requirements after snap engagement.
3. Local heating at certain temperatures at certain locations induces displacement sufficient for unlocking snaps.
4. Uniform heating does not induce displacement required to unlock snaps.
5. Any other problem specific constraints.

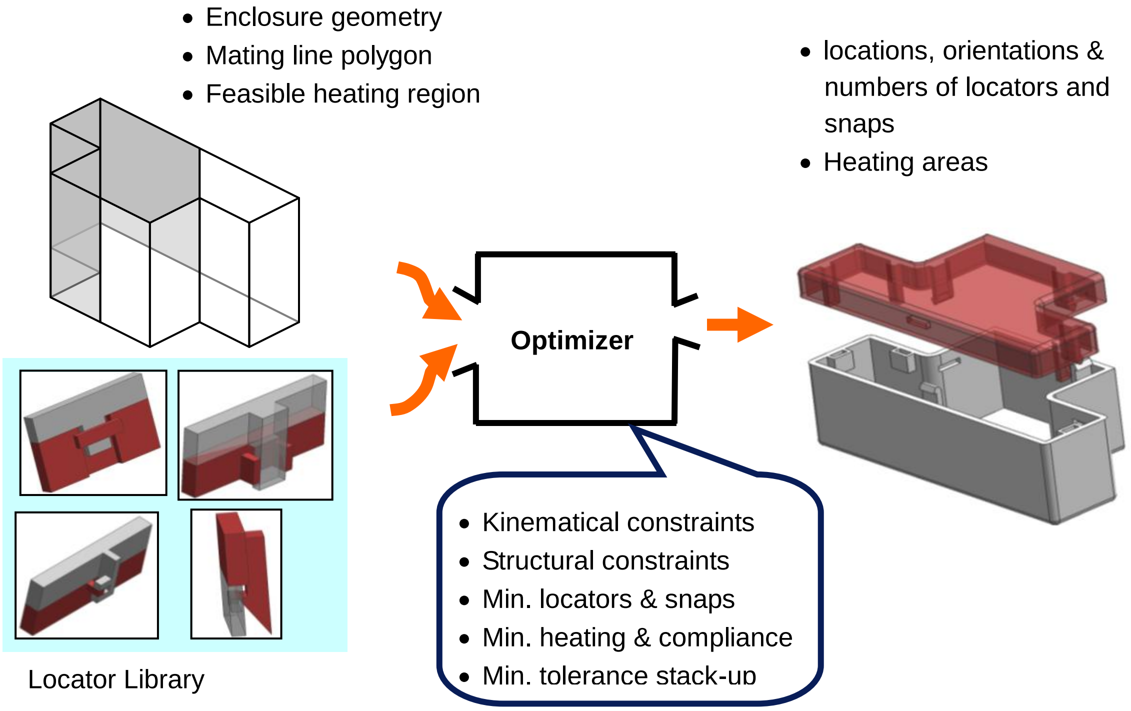

• Enclosure geometry
• Mating line polygon
• Feasible heating region
• locations, orientations & numbers of locators and snaps
• Heating areas
Optimizer
• Kinematical constraints
• Structural constraints
• Min. locators & snaps
• Min. heating & compliance
• Min. tolerance stack-up
Locator Library

Figure 3.5: Optimization problem for heat-reversible snap joints

Figure 3.5 summarizes the design problem of a heat-reversible locator-snap system of joints with an application on an eight sided polygon. The optimizer can be considered as a black box that takes the polygon geometry, locator library, the constraints and the required objectives; it then returns the locations, numbers and orientations of the locators and snaps, and the locations of the heating areas.

The following subsections describe in more detail the definition of the design objective, the problem inputs, the process of generating the set of feasible orientations of locators and snaps, and the optimization problem formulation.

3.2.1 Design objectives

The minimization of the number of locators and snaps, the compliance of the structure and the heating area(s) are obvious objectives to have simple and stiff design that is comparable to using rigid screws and energy efficient design that minimizes the amount of energy needed to heat the enclosures.

Tolerance stack-up is caused when the assembly of the enclosures is over constrained by multiple joints. Due to each enclosure's manufacturing variations including the manufacturing variations in the locators, the mating features are neither in perfect shape nor in perfect location or orientation. As a result, mismatch (gap or interference) exists between mating features of the two parts being joined.

Figure 3.6 illustrates typical stages in the assembly process of general flexible parts [114]-[118]. A part is placed on another part or a fixture, pull-up force is applied to close the mismatch or bring each mating feature to its nominal (clamping) location, where they are fixed. Then, the complete assembly is released from clamps and fixtures. The location of any point to be measured will be displaced during these stages.

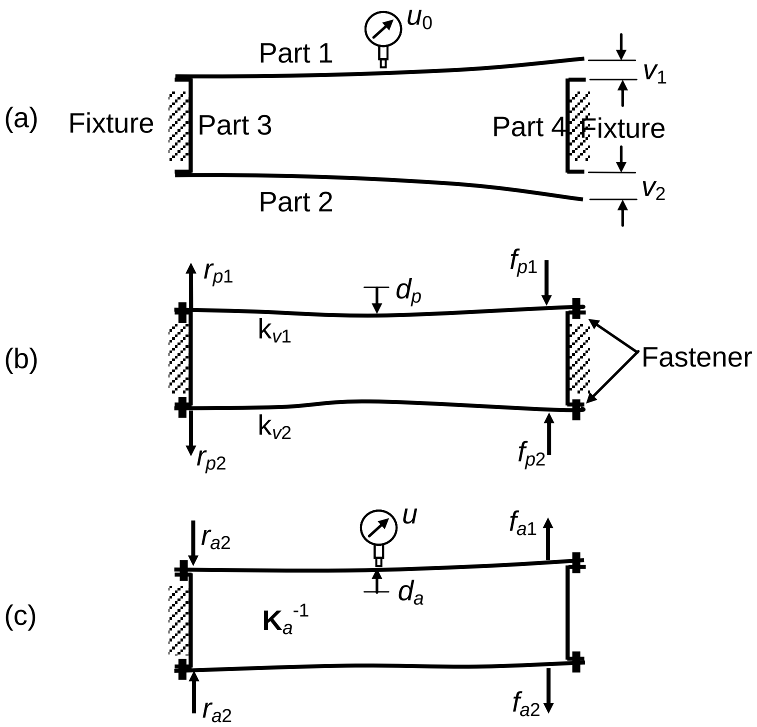


Figure 3.6: Steps to estimate assembly variation with flexible parts: (a) initial position, (b) clamping deformation, and (c) spring-back deformation.

The variation in the final location (u) of any point in the final assembly can be obtained using the following equation:

$$\mathbf{u} = \mathbf{u_0} + \mathbf{d_p} + \mathbf{d_a} \tag{3.1}$$

Where u_0 is the initial location of an arbitrary point of interest, $\boldsymbol{d_p}$ is the clamping deformation, and $\boldsymbol{d_a}$ is the spring back deformation. The evaluation of $\boldsymbol{u}$ is extremely time-consuming as it requires FEA and Monte Carlo simulations [119]. Thus, in order to avoid the evaluation of $\boldsymbol{u}$ during the optimization process, another objective is used. Since the deformation effect of each joint is usually local to the vicinity of that joint, the minimization of the tolerance stack-up is handled here by maximizing the distance between the joints (locators or snaps) that constrain the same degrees of freedom (DOF).

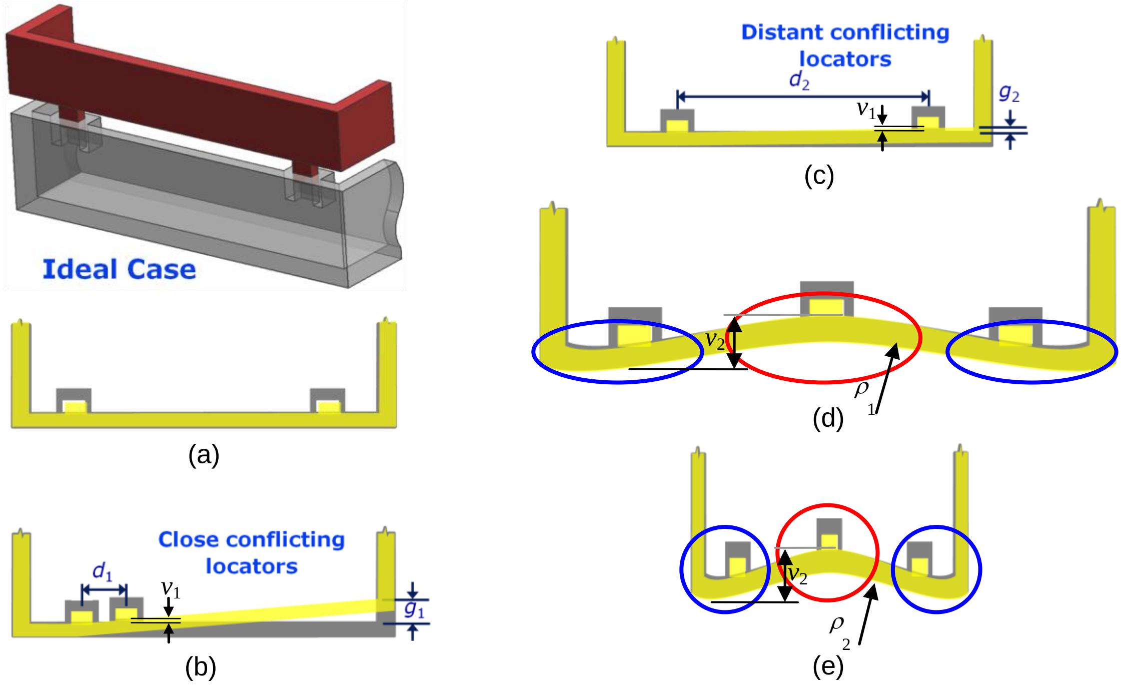


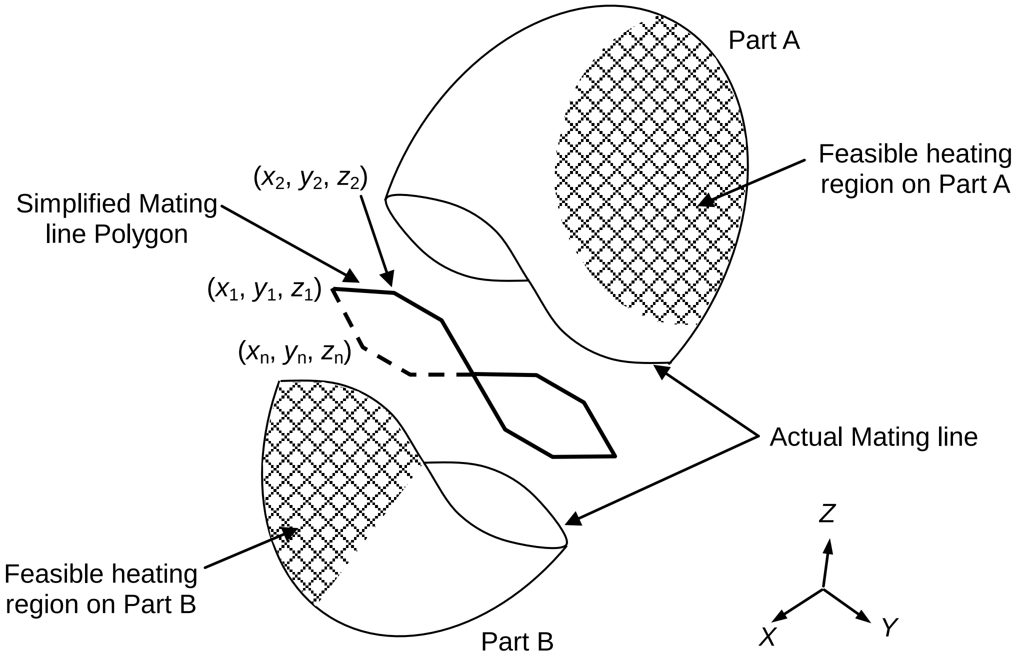
Figure 3.7: Tolerance stack-up example: (a) ideal case, and response to dimensional variations: (b) v_1 for a locator at a distance d_1, (c) v_1 for a locator at a distance d_2, (d) v_2 for distant locators, and (d) v_2 for close locators.

Figure 3.7 shows some examples of the effect of the fixation location on the tolerance stack-up. Figure 3.7a shows the ideal case. In Figure 3.7b and Figure 3.7c there is a dimensional variation v_1 in the right locator. As the distance, d, between the locators decreases, the gap, g, increases ($d_1 > d_2 \rightarrow g_2 > g_1$). Similarly, as shown in Figure 3.7d and Figure 3.7e, for the same dimensional variation, v_2, in the middle locator, as the distance between the locators decreases, the curvature, ρ, increases. As a result, it can be deduced that as the distance between locators that constrain the same DOF increases, the effect of dimensional variations is minimized.

3.2.2 Inputs

Figure 3.8, shows a generic enclosure composed of two parts A and B. The enclosures mate at a curved line that is simplified by the bold straight line shown in the figure. The simplified mating line can, then, be easily presented by the coordinates of its vertices in a consecutive order as shown in the following equation.

$$Coords = \{(x_1, y_1, z_1), (x_2, y_2, z_2), ..., (x_n, y_n, z_n)\} \tag{3.2}$$



Part A
Feasible heating region on Part A
(x_2, y_2, z_2)
Simplified Mating line Polygon
(x_1, y_1, z_1)
(x_n, y_n, z_n)
Actual Mating line
Z
X
Y
Feasible heating region on Part B
Part B

Figure 3.8: Part geometry, coordinates of vertices of the mating polygon, and the feasible regions for heating.

The feasible region where heating can be applied, P_h, is any area on the outer surface of the mating parts where heating can be applied. In general, the feasible heating region can be the whole outer surface of the two mating parts that can be accessed. In cases where the enclosures contain heat sensitive components the surfaces close to these components have to be excluded from the feasible heating region. In Figure 3.8, P_h consists of the two arbitrary shaded regions on the outer surfaces of Parts A and B. As an example, we can assume that the remaining surfaces of Parts A and B are not accessible or that they include heat sensitive components, thus it is infeasible to apply heat on such surfaces.

The locator library has the schematic geometry of the locators (and the associated protrusions) and the snaps (and the associated catches) available for a given problem. Since the snaps and locators are usually made of the same material as the enclosure, which is usually compliant, then they cannot be considered as rigid joints. In addition, since the locators are very small compared to the whole enclosure, it will be really complicated to include the locators with their actual geometry in the whole enclosure model. Thus, equivalent springs that simulate the actual behavior of the locators and snaps are used in the enclosure model instead of using the actual locators and snaps.

The equivalent springs, should have the same structural properties as the actual locators they represent. The stiffness $\boldsymbol{k} = (k_x, k_y, k_z)$ of the equivalent springs for a locator or a snap in local coordinates is obtained by measuring the tip deflections of the locator in response to a unit load in the directions the locator/snap constrain using finite element analysis. The equivalent locator/snap stiffness is dependent on the geometry of the locator and the panel it is attached to (locator and panel thicknesses (t_1 and t_2), locator and panel widths (w and W) and locator and panel heights (h and H)). Hence, the equivalent stiffness values are problem dependant and have to be measured for each problem separately. This information is also included in the locator library to be used in the design process.

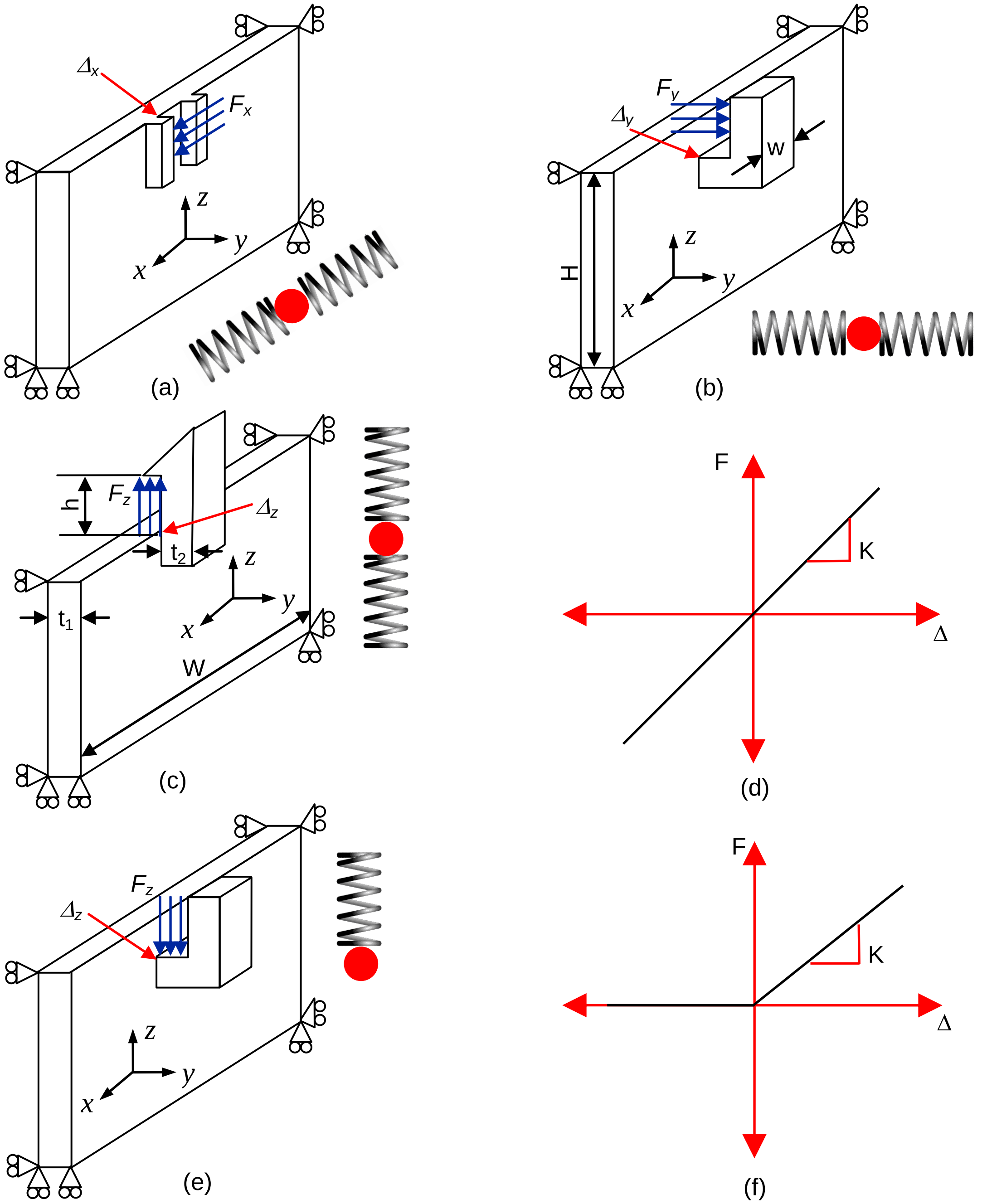


Figure 3.9: Measuring locator and snap stiffness: (a, b and c) bi-directional constraining locators/snaps in *x, y* and *z*-directions respectively, (d) force-displacement curve for bi-directional locators, (e) unidirectional constraining locator, and (f) force-displacement curve for unidirectional locators.

Figure 3.9 shows examples of locators and the equivalent springs associated with them. The locators in Figure 3.9a and Figure 3.9b constrain the motion in ±*x* and ±*y* directions respectively. The snap in Figure 3.9c constrains the motion in ±*z* directions. It should be noted that a *z* direction constraining locator will have the same behavior as the snap in Figure 3.9c. All the aforementioned locators are bidirectional and are assumed to have the same stiffness in both directions as they are assumed to be symmetric. As a result, the force displacement curve should be linear, as shown in Figure 3.9d. The stiffness values can be obtained by applying a unit force in the constraining direction and measuring the deformation value in the same direction using finite element analysis. The locator in Figure 3.9e has the same geometry as the locator in Figure 3.9b, yet it is unidirectional along the *z* axis; i.e. it constrains the motion in only one direction. The part in Figure 3.9e is free to move in the –*z* direction, but is constrained in the +*z* direction. As a result, the stiffness of such a locator is zero in the –*z* direction and is equal to (*F*/*Δ*) in the +*z* direction as shown in Figure 3.9f.

The wrench matrices $\left(\mathbf{W}_l^{local}\right)$ representing the motion constraints imposed by each locator and snap with respect to their local coordinate system are included in the Library as well. Then, the locator library can be represented as shown in the equation below.

$$LL=\left\{\left(\mathbf{W}_l^{local},\boldsymbol{k}_l\right),l\in L\right\}\cup\left\{\left(\mathbf{W}_l^{local},\boldsymbol{k}_l\right),l\in S\right\} \tag{3.3}$$

L and *S* are the sets of the available types of locators and snaps respectively. Figure 3.10 shows a sample of different locators and snaps that are included in the Library. The wrench matrices $\left(\mathbf{W}_l^{local}\right)$ representing the motion constraints imposed by the locators and the snaps in Figure 3.10 with respect to its local coordinate system are given in Equations (3.4)-(3.7).

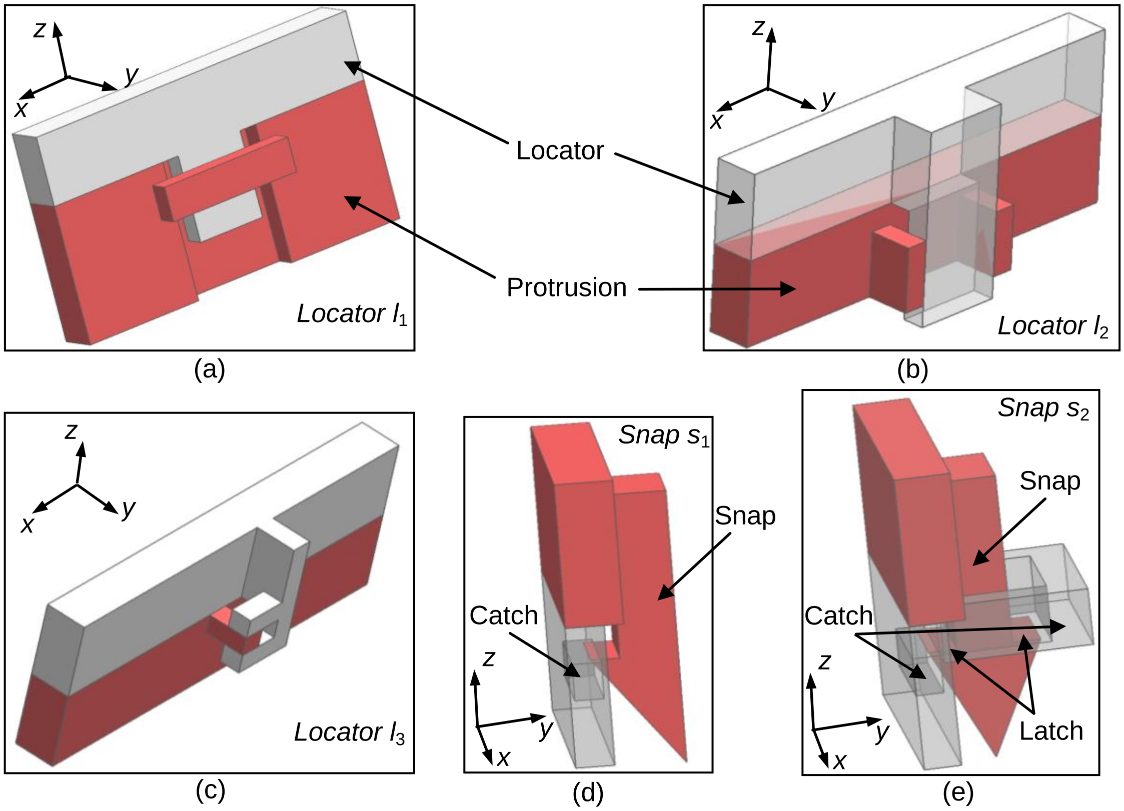


Figure 3.10: Sample locators and snaps in the locator library.

The locator in Figure 3.10a (l_1) constrains the motion in ±*y* and –*z* directions in the local coordinate system, whereas, the locator in Figure 3.10b (l_2) constrains the motion in ±*x* and –*z* directions in the local coordinate system, and the locator in Figure 3.10c (l_3) constrains the motion in ±*z* directions in the local coordinate system. Finally, the snap in Figure 3.10d (s_1) is for a single latching snap, whereas Figure 3.10e (s_2) is for a double-latching snap. Both snaps constrain the motion in +*z* direction in the local coordinate system. It should be noted that snaps can be defined to constrain the –*z* motion as well. The aforementioned conditions can be more precisely expressed using the Screw Theory, adopting the wrench matrix representation similar to [100] as follows[1]:

$$\mathbf{W}_{l_1}^{local} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{3.4}$$

$$\mathbf{W}_{l_2}^{local} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{3.5}$$

$$\mathbf{W}_{l_3}^{local} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \tag{3.6}$$

$$\mathbf{W}_{s_1}^{local} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \tag{3.7}$$

Where each row represents the directional (row) vectors of the force and moment in the local reference frame, which can be supported by a mating surface in a locator or a snap. For example, in the wrench matrix for (l_1), Equation (3.4), the 1st row has 1 at the 2nd column, indicating the locator surface can support the force in –*y* direction, while the –1 in the 2nd row and column indicates that the locator surface can support the force in +*y* direction. Finally the 1 in the 3rd row and column indicates that the locator surface can

[1] A review of the screw theory is given in Appendix A.

support the force in –*z* direction. Note that the moments (the 4th, 5th, and 6th columns) are ignored because our primal concern is the translational degrees of freedom. This is justified by using wide locators and snaps that will prevent enclosures from rotating.

In order to proceed with any further constraint analysis, locators and snaps are transformed to the global coordinate system based on their location relative to the edges of the mating line. The wrench matrix of a locator or a snap placed on an edge is transformed to the global coordinate system using two rotation matrices. The first matrix **R₁** defines the rotational orientation of the locator/snap with respect to the edge it is attached to. The second matrix **R₂** represents the directional cosines of the edge with respect to the global coordinate system as shown in Equation (3.8). The equivalent springs are also transformed to the global coordinate system using the same local-global transformation of the wrench matrices.

$$\left(\mathbf{W}_{n\times 3}^{global}\right)^{T} = \mathbf{R}_2\mathbf{R}_1\left(\mathbf{W}_{n\times 3}^{local}\right)^{T} \qquad (3.8)$$

$\mathbf{W}_{n\times 3}$ = the first 3 elements in each row for all the n rows of $\mathbf{W}$

In the example in Figure 3.11, the snap wrench matrix is defined in the coordinate system (*x*', *y*', *z*') and is inclined at an angle *theta* with respect to the edge's coordinate system (*x*, *y*, *z*). The rotation matrix **R₁** transforms the snap wrench matrix to the edge's coordinate system, while the rotation matrix **R₂** transforms the edge's coordinate system to the global coordinate system (*X*, *Y*, *Z*). **R₁** and **R₂** are given in Equations (3.9) and (3.10).

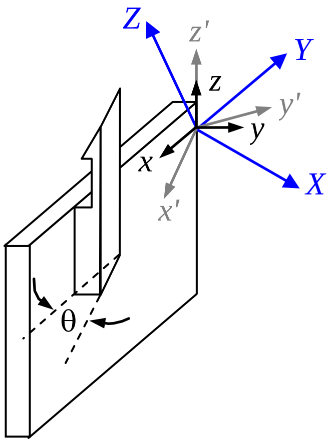


Figure 3.11: sample snap with respect to the local/global coordinate systems.

$$\mathbf{R}_1 = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.9}$$

$$\mathbf{R}_2 = \begin{pmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{pmatrix} \tag{3.10}$$

3.2.3 Generation of the feasible orientations of locators and snaps

In order to avoid examining a large number of infeasible designs during optimization, the set of all combinations of locators and snaps that satisfy the motion constraints of the optimization problem, presented in Section 3.2, is pre-calculated using Screw Theory. It is assumed that:

- Locators (and the associated protrusions) and snaps (and the associated catches) can be placed on either of the two mating parts (part A or part B in Figure 3.8).

- Each locator (and its associated protrusion) and/or snap (and its associated catch), l_i, can be placed only at a predefined discrete location, x_i, where x_i is the I.D. of the discrete location (*eg.,* nodes of finite elements) on the internal surface of edge *i* on which the locator or snap is placed.
- Each locator and/or snap can be placed at a predefined discrete orientation or angle (*eg*., among {0^o, 45^o, 90^o, 135^o, 180^o}), relative to the internal surface of the edge on which the locator or the snap is placed.
- Each edge of the mating polygon can have one or more locators (or the associated protrusions) and/or snaps (or the associated catches). Locators and/or snaps can be of any type and in any orientation.
- If an edge can have *k* locators or snaps of different types and/or different orientations, then this edge is considered as *k* apparent edges. For example, if a mating polygon, that actually has 4 edges, can have two different types of snaps and/or locators per edge, then the total number of apparent edges in the mating polygon will be considered as eight (2×4).

Based on the above assumptions, all possible combinations of locators, snaps, orientations, and edges can be enumerated. The total number of enumerations is given by m^n. Where $m = |L|+|S|$ is the total number of locator and snap types to choose from in the locator library *LL*, and *n* is the total number of apparent edges in a given problem. Since relative motion constraints on an edge are independent of the choice of the part on which the locators or snaps are placed, this choice can be ignored for the purpose of the analysis of motion constraints. As such, a combination of locators, snaps, orientations, and apparent edges can be represented as:

$$\boldsymbol{z} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n) \tag{3.11}$$

$$\boldsymbol{a}_i = (l_i, o_i); \quad i = 1, 2, \ldots, n \tag{3.12}$$

Where $l_i \in \{nil\} \cup L \cup S$ is the locator/snap type from the locator library (*nil* if no locator/snap), $o_i \in O$ is the orientation of a locator/snap relative to the i^{th} edge surface chosen from predefined choices O, (ignored if $l_i = nil$). Each combination, **z**, of locators, snaps, orientations, and apparent edges is tested against two motion constraints: 1) the parts are under constrained before snap engagement and 2) the parts are not under constrained (*i.e.*, can be over constrained) after snap engagement. After testing, only the combinations that satisfy both conditions are stored in a set F of feasible orientations to be examined during optimization.

$$F = \left\{ \boldsymbol{z} \mid \boldsymbol{z} \in \left(\left(\left\{ nil \right\} \cup L \cup S \right) \times O \right)^n \text{ and satisfies the motion constrains} \right\} \tag{3.13}$$

These two motion constraints are mathematically represented as follows. All the possible translational motions are included in a twist matrix $\mathbf{T}_{all} = (\boldsymbol{t}_1,\ldots,\boldsymbol{t}_r)^{\mathrm{T}}$, where $\boldsymbol{t}_i$ the twist vector in the i^{th} possible direction and r is the total number of possible translational directions. Based on the principle of virtual work, the forces and moments represented by the wrench matrix **W** = ($\boldsymbol{w}_1,\ldots,\boldsymbol{w}_{nw}$)$^{\mathrm{T}}$ constrains the motions represented by twist matrix **T** = ($\boldsymbol{t}_1,\ldots,\boldsymbol{t}_{mt}$)$^{\mathrm{T}}$ if and only if there exists a negative component in every column of the virtual coefficient matrix [98]:

$$\Delta(\mathbf{W},\mathbf{T}) = \begin{pmatrix} \delta(\boldsymbol{w}_1,\boldsymbol{t}_1) & \cdots & \delta(\boldsymbol{w}_1,\boldsymbol{t}_{mt}) \\ \vdots & \ddots & \vdots \\ \delta(\boldsymbol{w}_{nw},\boldsymbol{t}_1) & \cdots & \delta(\boldsymbol{w}_{nw},\boldsymbol{t}_{mt}) \end{pmatrix} \tag{3.14}$$

where $\sigma(\boldsymbol{w},\boldsymbol{t})$ is the virtual coefficient of wrench $\boldsymbol{w} = (\boldsymbol{f}^T, \boldsymbol{m}^T)$ and twist $\boldsymbol{t} = (\boldsymbol{\omega}^T, \boldsymbol{v}^T)$:

$$\delta\left(\boldsymbol{w},\boldsymbol{t}\right) = \boldsymbol{v}\bullet\boldsymbol{f} + \boldsymbol{\omega}\bullet\boldsymbol{m} \tag{3.15}$$

Equivalently, this can be written as:

$$\text{fully-constrained}(\Delta(\mathbf{W},\mathbf{T})) = \begin{cases} \mathit{true} & \text{if } \forall j, \exists i,\ \delta(\boldsymbol{w}_i,\boldsymbol{t}_j) < 0 \\ \mathit{false} & \mathit{otherwise} \end{cases} \tag{3.16}$$

Equation (3.16) gives a compact representation of the above two conditions for feasible locators and snap orientations that can be expressed using Equations (3.17) and (3.18):

$$\text{fully-constrained}(\Delta(\bigcup_{l\in L}\mathbf{W}_l, \mathbf{T}_{all})) = \mathit{false} \tag{3.17}$$

$$\text{fully-constrained}(\Delta(\bigcup_{l \in L \cup S} \mathbf{W}_l, \mathbf{T}_{all})) = \mathit{true} \qquad (3.18)$$

Where L and S are the sets of locators and snaps, respectively, and $\mathbf{W}_l$ is the wrench matrix of a locator (if $l \in L$) or a snap (if $l \in S$), and $\mathbf{T}_{all}$ is the twist matrix of all translational motions in all the possible directions.

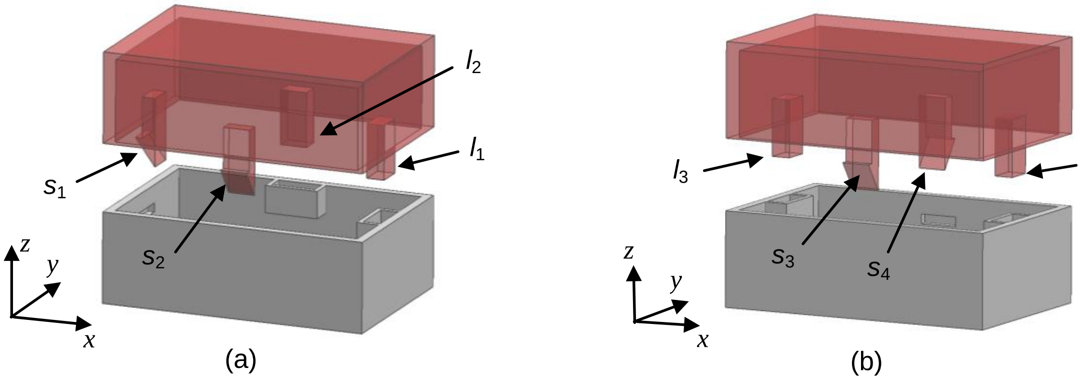


Figure 3.12: Examples of two different locator and snap orientations

Figure 3.12 illustrates examples of two motion constrains. In the figure, it is assumed that a locator can constrain the normal direction (positive and negative) of the surface on which it is placed and its direction of insertion (–z in Figure 3.12), a snap can only constrain its direction of disengagement (+z in Figure 3.12), and there is no neighboring part that might cause interferences. In the orientations shown in Figure 3.12a, both conditions are satisfied. Locators l_1 and l_2 constrain the motions in the ±x and –z, and ±y and –z directions respectively, but nothing constrains the +z direction. After snapping, snaps s_1 and s_2 provide the constraint in the +z direction, thereby fully constraining the two mating parts. On the other hand, in the orientations shown in Figure 3.12b, the second condition is not satisfied. Locators l_3 and l_4 constrain the motion only in the ±*x* and –*z* directions, whereas snaps s_3 and s_4 constrain the +*z* direction. As a result, the enclosures remain under constrained after snap engagement as it is free to move in the ±*y* direction.

The above conditions for the locators and snaps used in this example are expressed using the wrench matrix representation as given in Equations (3.19)-(3.21). It should be noted that these Wrench matrices are given in the global reference frame.

$$\mathbf{W}_{l_1} = \mathbf{W}_{l_3} = \mathbf{W}_{l_4} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{3.19}$$

$$\mathbf{W}_{l_2} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{3.20}$$

$$\mathbf{W}_{s_1} = \mathbf{W}_{s_2} = \mathbf{W}_{s_3} = \mathbf{W}_{s_4} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \tag{3.21}$$

Using Equations (3.19)-(3.20), for example, the virtual coefficients matrix for Figure 3.12a before snap engagement is given as shown in Equation (3.22):

$$\Delta(\bigcup_{l\in\{L_1,L_2\}} \mathbf{W}_l, \mathbf{T}_{all}) = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \end{pmatrix} \tag{3.22}$$

Since the 5th column has no negative entry, fully-constrained = *false*. If $\mathbf{W}_{s1}$ and/or $\mathbf{W}_{s2}$ are added, *i.e*. snaps are engaged, the virtual coefficients matrix will have at least one negative entry in each row, thus fully-constrained = *true*. On the other hand, the virtual coefficients matrix for the design in Figure 3.12b after snap engagement is given as:

$$\Delta(\bigcup_{l\in\{L_1,L_2\}\cup\{S_1,S_2\}} \mathbf{W}_l, \mathbf{T}_{all}) = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \tag{3.23}$$

Since the matrix does not have negative values in +*y* or –*y* axis, Equation (3.18) is not satisfied; the design is always under-constrained in the *y* direction.

Since Equations (3.17) and (3.18) do not prohibit over constraining of the enclosures, the same degree of freedom can be constrained by multiple locators and/or snaps. Over constraining of the enclosures may cause undesirable tolerance stack-up. In this dissertation, the effect of tolerance stack-up is minimized by maximizing the distance between locators/snaps that constrain the same degree of freedom, as mentioned in Section 3.2.1.

3.2.4 Simultaneous optimization of locators/snaps and heating areas

In addition to the satisfaction of motion constraints, an enclosure assembly must satisfy the following thermal and structural requirements:

1. Snaps do not accidentally unlock under normal working conditions.
2. Local heating at certain temperatures and locations induces displacement sufficient for unlocking snaps.
3. Uniform heating does not induce displacement sufficient for unlocking snaps.
4. Any other problem-specific thermal or structural requirements.

In the optimization problem stated earlier, Requirement 1 is stated as minimizing the compliance of the structure and is regarded as one of the objective functions to be minimized. Requirement 2 is for the desired reversal behavior of locator-snap system. Requirement 3 is for preventing accidental disassembly during the use in elevated temperatures. Examples of Requirement 4 include guarding against thermal damage and resonance vibration. There are other practical considerations that can be included in the requirements but are not discussed in this dissertation. For instance, when parts are over-constrained, manufacturing variations result in residual stresses and therefore creep.

The following four design variables are defined for the simultaneous optimization of locators/snaps and heating areas:

- $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ where $\boldsymbol{x}_i$ is the id vector of the d_i finite element nodes on the apparent edge i on which locators or snaps are placed; $x_{ij}= nil$ if the j^{th} locator/snap is not placed on apparent edge i.

- $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_p)$ where p is the number of heated polygons, $\boldsymbol{y}_i$ is the coordinate vectors of the v vertices of the i^{th} polygonal area to be heated; $\boldsymbol{y}_i = nil$ if the i^{th} heating area is undefined.
- $\boldsymbol{z} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n)$ where $\boldsymbol{z}$ is a variable that defines a feasible combination of locators, snaps and orientations from the feasible set, F, defined in Equation (3.13), $\boldsymbol{a}_i = (l_i, o_i)$ are a choice of locator/snap and its orientation of the i^{th} edge as defined in Equation (3.12); $l_{ij} = nil$ if the i^{th} apparent edge does not have a locator/snap, in which case the value of o_i is ignored.

Using $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, the optimization problem can be written as:

$$\text{minimize } \left\{ f_1(\boldsymbol{y}),\ f_2(\boldsymbol{x},\boldsymbol{z}),\ f_3(\boldsymbol{x},\boldsymbol{z}), \cdots,\ f_q(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \right\}$$

subject to:

$$\text{min\_disp\_loc}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l) > h_{lower}$$

$$\text{max\_disp\_loc}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\ \boldsymbol{T}_l) < h_{upper}$$

$$\text{max\_disp\_unif}(\boldsymbol{x},\boldsymbol{z},T_u) < h\_$$

$$\text{struct\_req}(\boldsymbol{x},\boldsymbol{z}) = true$$

$$x_{ij} \in \{nil, L_i, L_i+1, \cdots, U_i\};\ i = 1,\cdots,n;\ j = 1,\cdots,d_i$$

$$\boldsymbol{y}_i \in P_h^v;\ \ i = 1,2,\cdots,p$$

$$\boldsymbol{z} \in F$$

Where:

- $f_1(\boldsymbol{y})$ is a measure for the p polygonal heating area(s) defined by the vertex vectors $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, …, $\boldsymbol{y}_p$. Although the exact area of the heating polygons can be calculated, counting the number of finite element nodes inside the heating polygons and using it as a measure for the size of the heating area(s) is accurate enough.
- $f_2(\boldsymbol{x}, \boldsymbol{z})$ is a measure for the compliance of the whole assembly. It is defined as the maximum deformation at the mating line during normal use (such as moving the component from one place to another, or deformation induced from the heating of the internal components while the product is turned on) of the product while snaps are engaged.
- $f_3(\boldsymbol{x}, \boldsymbol{z})$ is a measure for the tolerance stack-up. It is defined as the reciprocal of the minimum distance between the joints (locators or snaps) that constrain the same (DOF) and lie on different nodes. Thus, maximizing the distance between the conflicting joints. This is given as shown in Equation (3.24):

$$f_3(\boldsymbol{x},\boldsymbol{z}) = \left\{\min\left(\left|(x,y,z)^T_{li} - (x,y,z)^T_{lj}\right|\right)\right\}^{-1};\ i=1,\ldots,n,\ j=1,\ldots,n, \\ i \neq j,\ l_i \text{ and } l_j \text{ are conflicting} \tag{3.24}$$

Locators l_i and l_j are conflicting if the following inequality is satisfied:

$$Rank\left(\mathbf{W}^{global}_{li}\right) + Rank\left(\mathbf{W}^{global}_{lj}\right) > Rank\left(\mathbf{W}^{global}_{li} \cup \mathbf{W}^{global}_{lj}\right) \tag{3.25}$$

- $f_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, $i = 4, \ldots, q$ are any problem-specific objectives. For instance, minimization of the total number of locators and snaps, or minimization of the stresses induced during normal use.
- min_disp_loc($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{T_l}$) and max_disp_loc($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{T_l}$) are the minimum and maximum steady state thermal displacements, respectively, in the snap unlocking direction of all nodes on which snaps/catches are placed, when local heating is applied at the locations specified by $\boldsymbol{y}$ to the temperatures given by the temperatures vector, $\boldsymbol{T_l} = (T_{l1}, T_{l2}, \ldots, T_{lp})$.
- max_disp_unif($\boldsymbol{x}$, $\boldsymbol{z}$, T_u) is the maximum steady state thermal displacement, in the snap unlocking direction of all nodes on which snaps/catches are placed, when uniform heating is applied with a temperature T_u.
- h_{upper} and h_{lower} are the upper and lower values of the displacement, respectively, to unlock the double-latch snaps.
- h_-, is the height of snaps minus a small tolerance.
- struct_req($\boldsymbol{x}$, $\boldsymbol{z}$) is any structural requirement (other than f_2) during snap engagement.
- L_i and U_i are lower and upper bounds of the node numbers of the finite elements on edge i, respectively.
- P_h is the feasible region of the heating area.
- F is the set of feasible combination of locators, snaps, orientations, and edges defined by Equation (3.13).

The evaluation of min_disp_loc($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{T}_l$), max_disp_loc($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{T}_l$) and max_disp_unif($\boldsymbol{x}$, $\boldsymbol{z}$, T_u) require thermal and structural FEA, whereas the evaluation of f_2($\boldsymbol{x}$, $\boldsymbol{z}$) and struct_req($\boldsymbol{x}$, $\boldsymbol{z}$) require structural FEA only.

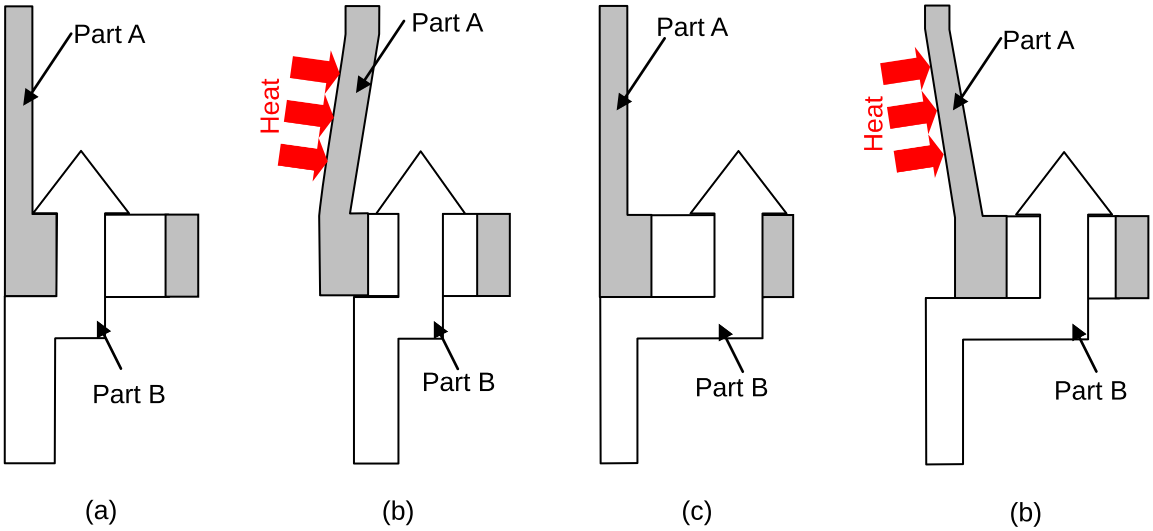


Figure 3.13: Examples double latching snap attachments with snaps attached to part B: Part A bulges outwards when heated (a and b) and part A bulging inwards when heated (c and d).

It should be noted that variables $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ do not explicitly specify the choice of the part on which a locator/protrusion, or a snap/catch, should be placed. Since the choice does not affect the motion constraints or the structural behavior during snap engagement, it can be arbitrary in the case of a locator/protrusion pair. In the case of a double-latching snap/catch pair its location can be arbitrary as well; yet, the snap geometry will change slightly as shown in Figure 3.13. Finally, in the case of a snap/catch pair, the choice is determined based on the thermal deformation upon heating. If upon heating the surface of the heated part bulges outwards; then a catch is placed on the heated part as shown in Figure 3.14a and b. If upon heating the surface of the heated part bulges inwards; then a snap is placed on the heated part as shown in Figure 3.14c and d.

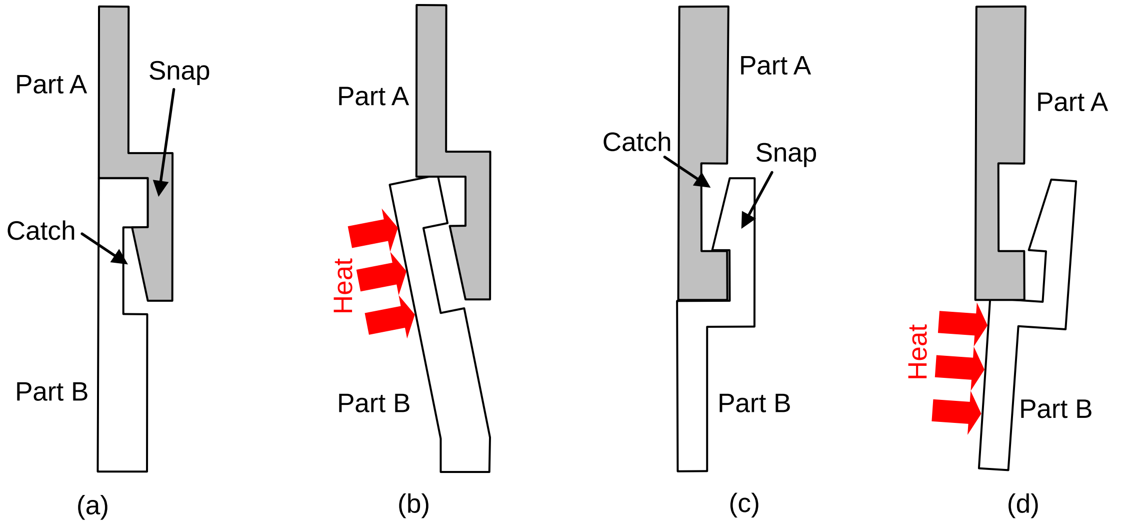


Figure 3.14: Examples of two different snap attachments with heating applied to Part B: snap attached to part A and bulging is outwards (a and b) and snap attached to part B and bulging is inwards (c and d)

3.3 Optimization Algorithm

The problem involves multiple objectives that need to be minimized. It also has mixed design variables (discrete and continuous variables) and can involve multiple local optima. With this large number of design variables and design objectives, the optimum solution is not really obvious to the designer and, thus, the design process requires optimization to reach the optimum solution.

The optimization problem is, therefore, solved using Multi Objective Genetic Algorithms (MOGA). MOGA is selected because of its ability to capture near-global optimal solutions and its ability to handle mixed (continuous and discrete) design variables. Since

MOGA cannot handle the constraints explicitly, the constraints are added as penalties to the objective functions. The optimizer returns the Pareto-optimal solutions that exhibit alternative designs with different trade-offs between design objectives.

3.3.1 Multi-Objective Genetic Algorithm (MOGA)

Genetic Algorithm (GA) is a heuristic optimization method that mimics the evolution of living creatures by simulating the process of natural selection in biological evolution. GA starts with randomly generating an initial population of design variables and chromosomes; and finds their fitness functions. For the subsequent generations, a selection process giving higher probability to chromosomes with higher fitness value, to aid the idea of survival of the fittest, is used to select chromosomes. Then, genetic operators, e.g. mutation and crossover, are applied to the selected chromosomes to finalize the new population and help keep diversity in the new population. Finally, the fitness functions for this population are calculated and the process is repeated until convergence or for a fixed number of generations.

MOGA is an extension of the conventional single objective GA. In conventional GA, all the objectives are aggregated to one value, for example by using their weighted sum. Yet in MOGA, each objective is evaluated independently and then the solution is ranked in the population based on the quality of all its objectives compared to the rest of the population. Among the multi-objective algorithms, Fast Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) is used in this research.

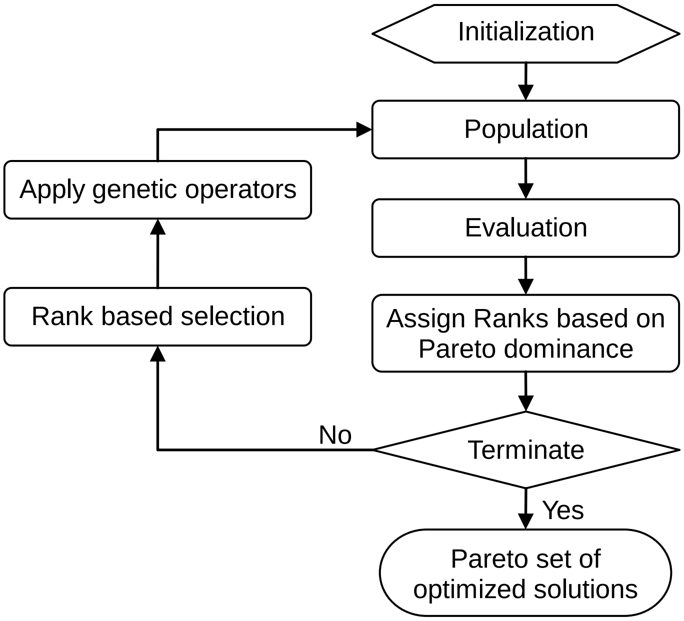


Figure 3.15: Flowchart of the NSGA-II used in this research

The flowchart in Figure 3.15 simply describes how NSGA-II works, and its basic steps are presented below[2]:

1. Create a population *P* of *n* chromosomes, encoded representation of the design variables, and evaluate their objective function values.
2. Rank each chromosome *c* in *P* based on the number of other chromosomes dominating *c* in Pareto sense (rank 0 is the Pareto optimal). Store the Pareto optimal chromosomes in the set *O* and create an empty subpopulation *Q*.
3. Select two chromosomes c_i and c_j in *P* with probability proportional to their ranks.
4. Apply genetic operators to c_i and c_j to generate c_i' and c_j'.
5. Evaluate the objective function values of c_i' and c_j' and store them in *Q*. If *Q* contains fewer elements than *n*, go to 3.

[2] Detailed description of NSGA-II is given in Appendix B.

6. Let $P \leftarrow Q \cup O$ and empty Q. Rank each chromosome in P and remove the chromosomes with the lowest ranks from P until the size of P becomes n.
7. Update the set O and increment the generation number. If the pre-specified maximum number of generations is reached, terminate the process and return O, otherwise go to 3.

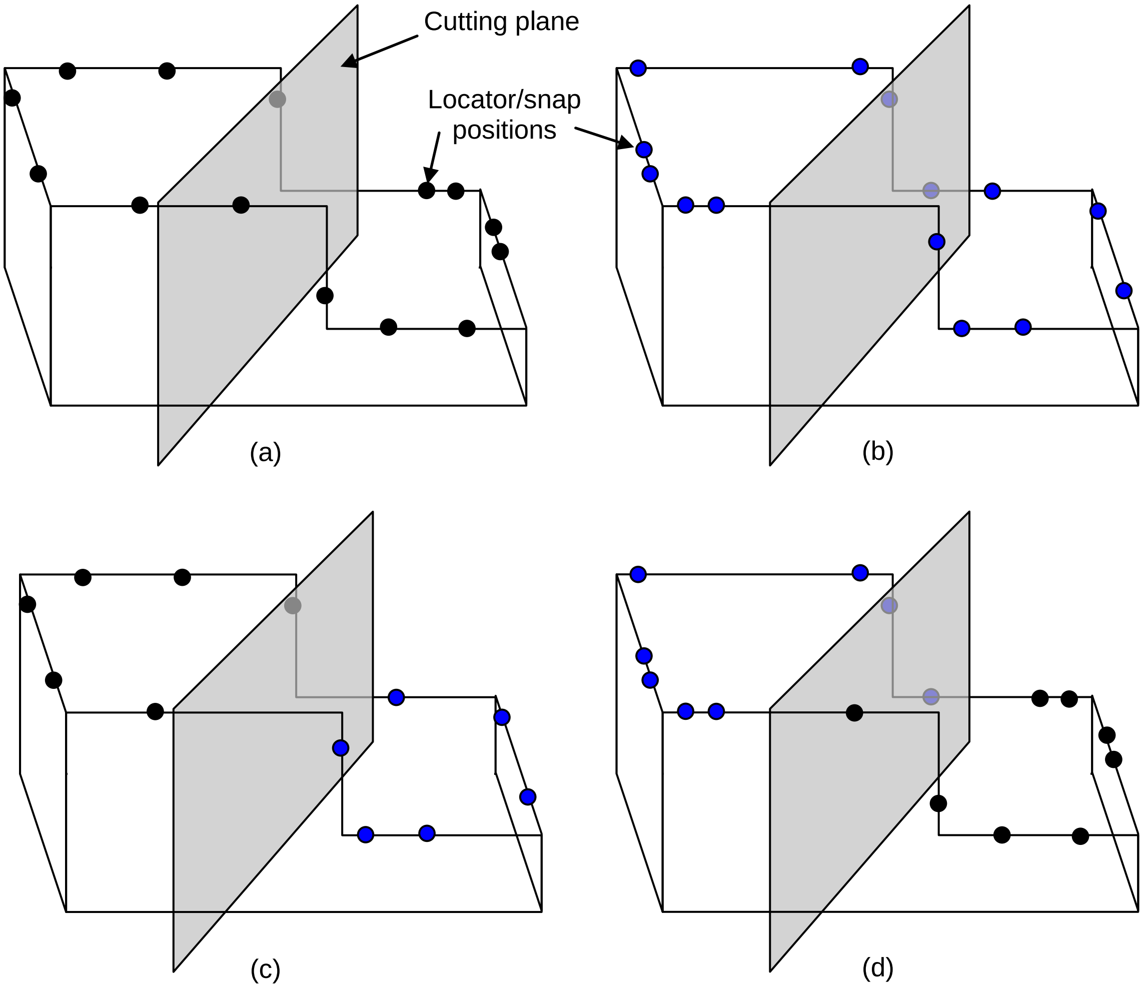


Figure 3.16: Geometric crossover: (a) parent p_1, (b) parent p_2, (c) child c_1, and (d) child c_2,

Because the locations of the locators and snaps are defined by the geometry of the enclosures, geometric based crossover [120] and the conventional uniform crossover [121] are used to deal with the discrete variables defining the locations of the locators and snaps (x_{ij}). Geometric crossover is applied using the following procedure:

1. Select a cutting plane randomly to cut the enclosure into two pieces.
2. Cut two parent designs p_1 and p_2 using the plane defined in step 1 (Figure 3.16a and Figure 3.16b).
3. Assemble two children designs c_1 and c_2 by swapping the pieces of p_1 and p_2 (Figure 3.16c and Figure 3.16d).

On the other hand, the real variables used to define the heating areas are defined using real coded genetic algorithms and use a combination of heuristic [122] and arithmetic crossovers [123].

Chapter 4: Heat-Reversible Locator-Snap System for Automotive Bodies

In this Chapter, the heat-reversible locator-snap system of joints is proposed as a joining method between internal frames and external panels in automotive bodies. The proposed locator-snap system allows non-destructive and clean separation between the plastic panel and the aluminum frame, in the next generation body structure (Aluminum Space Frame); thus, can highly improve the recyclability of aluminum.

4.1 Introduction

Aluminum space frame, Figure 4.1, is considered as the next generation body structure [124], [125] due to its lightweight (40% lighter than steel body [125]) and design freedom realized by the separation of “bone” and “skin” [126]. Panels, therefore, can be made of light weight materials, for instance plastics, as they carry minimal loads. They are also environmentally sound since the environmental impact (mainly CO_2 emission) during the use phase of the vehicle is very low due to the improved fuel efficiency (32% of fuel savings) owing to its lightweight. Table 4.1 compares the amounts of carbon dioxide emissions generated during the processing of the materials, manufacturing the body structures, use of the vehicle, and in recycling the materials for both steel and aluminum [127]. According to the previous results of life-cycle analyses [124], [128], it is essential to improve the quality of recycled aluminum space frame bodies in order to compare with the steel bodies, which currently, have lower energy consumption for production and higher recyclability as a raw material.

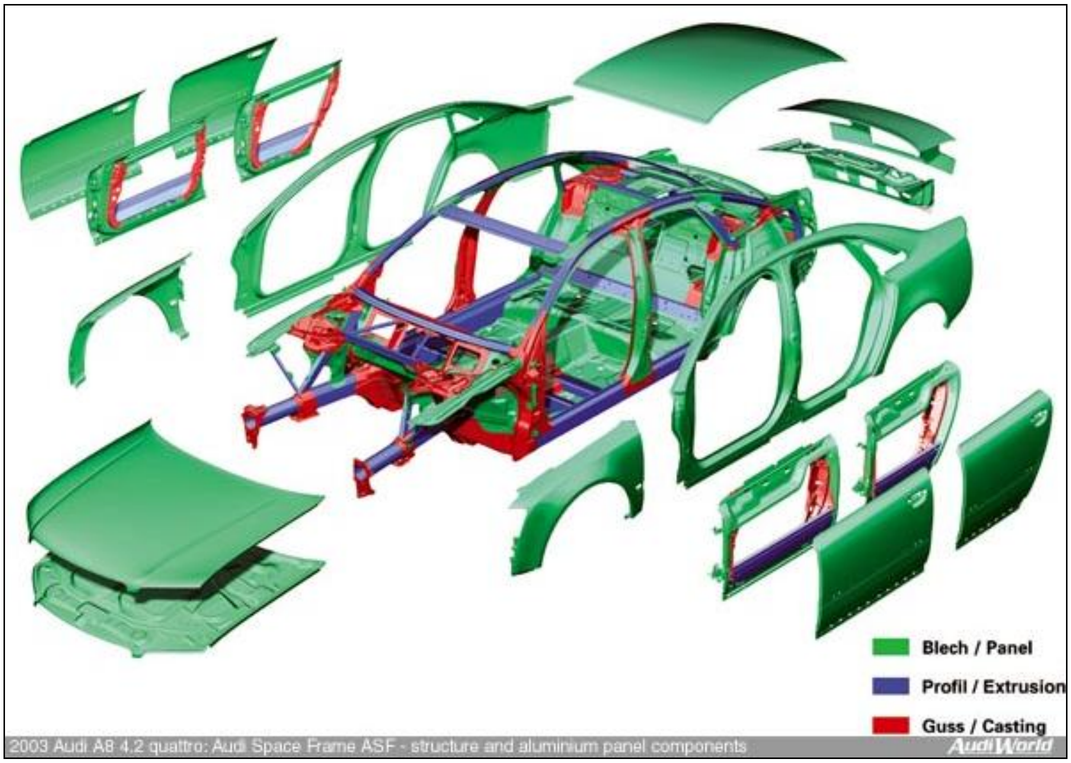

🟩 Blech / Panel
🟦 Profil / Extrusion
🟥 Guss / Casting
2003 Audi A8 4.2 quattro: Audi Space Frame ASF - structure and aluminium panel components
AudiWorld

Figure 4.1: Audi Space Frame [7].

Table 4.1: Total carbon dioxide emissions for steel and aluminum bodies in 1 Year [127]

Stage	**Steel (kg CO_2/Body)**	**Aluminum (kg CO_2/Body)**
Material processing	1913.5	2689
Manufacturing	19.5	18.6
Use	6772.5	6139.5
Recycling	282.5	75.7

One of the challenges in improving the recyclability of the aluminum space frame body structure is the clean separation between incompatible materials used in various body components, in particular, extruded aluminum structural frames and stamped (or sometimes injection-molded) external panels made of different materials. In the current aluminum space frame bodies, joining between the internal frames and the external panels is achieved using permanent joints such as self-piercing rivets and resistance spot-welding or using bolts. These permanent joints can only be detached destructively, inevitably leaving residues of mating materials that prevent the “closed loop” recycling of aluminum alloys. A study conducted by the Aluminum Company of America (ALCOA) [125] predicts that: if the current increasing trend of aluminum use in vehicles continues (which will further boost with the introduction of aluminum space frame bodies), there will be an abundance of unused cast aluminum, unless recycling to the same grade alloy (“closed loop” recycling) becomes economically feasible. It is essential, therefore, to find a joining method that allows easy, non-destructive detaching at a desired time.

4.2 Problem overview

In this case study, heat-reversible locator-snap system is proposed as the joining method between the front automotive fender (Figure 4.2a) and the automotive frame (Figure 4.2b). Such a system allows non-destructive and clean separation between plastic panels and aluminum frames; thus, can highly improve the recyclability of aluminum by allowing the recycling to the same grade.

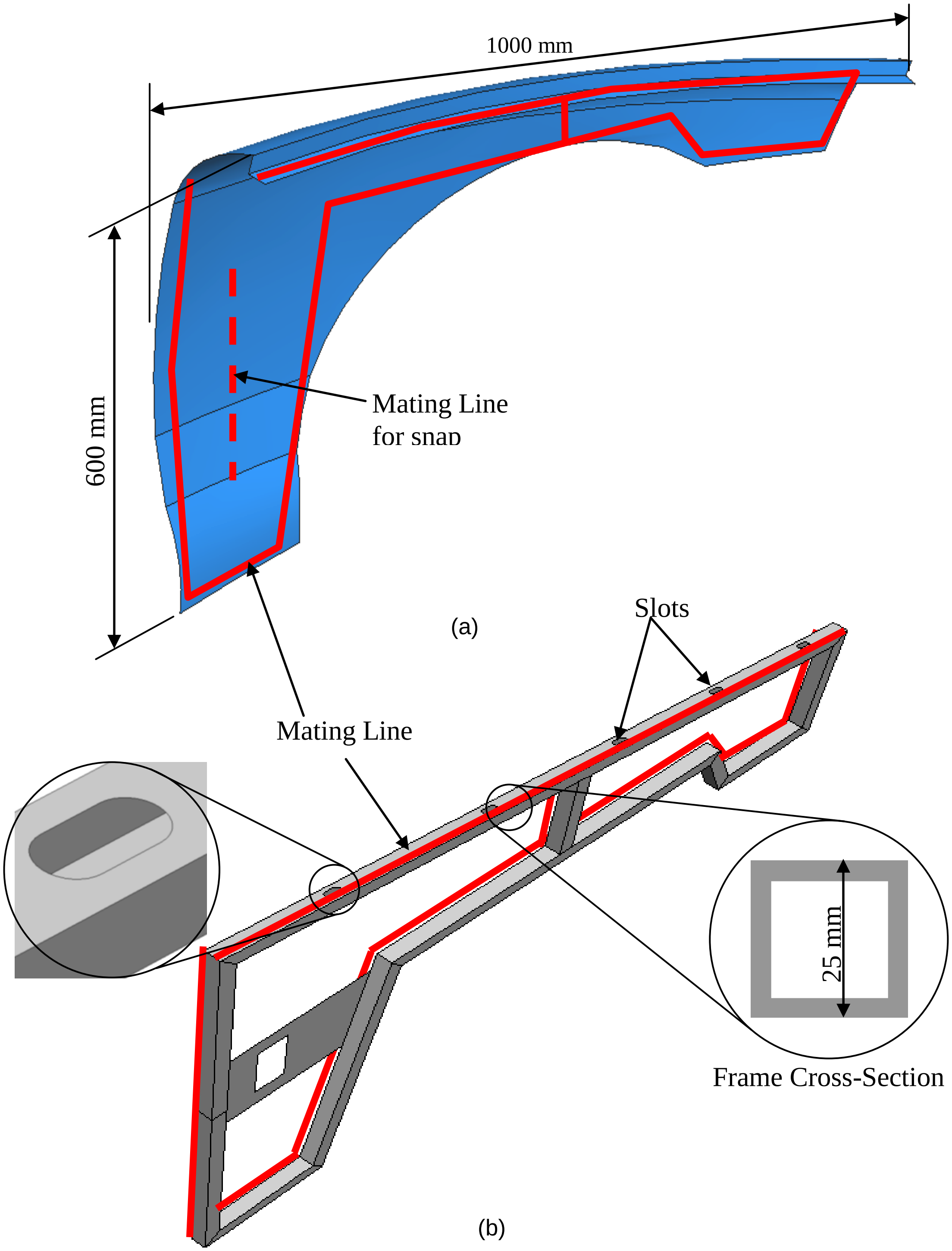


Figure 4.2: (a) simplified front fender panel and (b) internal frame.

The size of the panel is approximately 600 mm by 1000 mm, with a thickness of 3 mm. The panel is assumed to be injection-molded from Nylon 66 with 30% glass to minimize the weight of the panel; thus minimize the vehicle overall weight. The material properties are listed in Table 4.2. The frame is assumed to be made of a hollow aluminum beam of a square cross section with 25 mm at external sides.

Table 4.2: Material properties for nylon 66 - 30% glass filled.

Property Name (units)	Value
Density (g/cm3)	1.36
Elasticity modulus (MPa)	8500
Poisson Ratio	0.36
Melting point (^{o}C)	260
Thermal expansion coefficient (µm/m.^{o}C)	3.00
Specific heat capacity (j/kg.^{o}C)	1800
Conductivity (W/m.^{o}K)	0.40

Table 4.3: Vehicle sources of vibrations and their frequency ranges [129].

Vibration source	Frequency range (Hz)
Suspension and wheels	5-10
Engine	11-17
Body	25-40
Driveline	50-150

Automotive body panels are not desired to have natural frequencies within the ranges of the vibration inducing components, such as the engine and the suspension frequencies, to avoid resonance. A list of the common vibrations frequencies occurring during the normal operation of the vehicle is given in Table 4.3 [129]. As a result, locators should be positioned such that the natural frequencies of the panel, while attached to the frame, do not fall within any of the listed ranges.

4.2.1 Assumptions and inputs

The series of assumptions made in the given problem are summarized below:

- The orientations of the locators and the snaps are selected by inspection for each edge of the mating lines, Figure 4.3, and are verified to satisfy Equations (3.17) and (3.18). Consequently, the generation of the set of feasible locators and snaps is not needed and is disregarded. The design variable used to select a feasible combination of locators and snaps is also ignored.

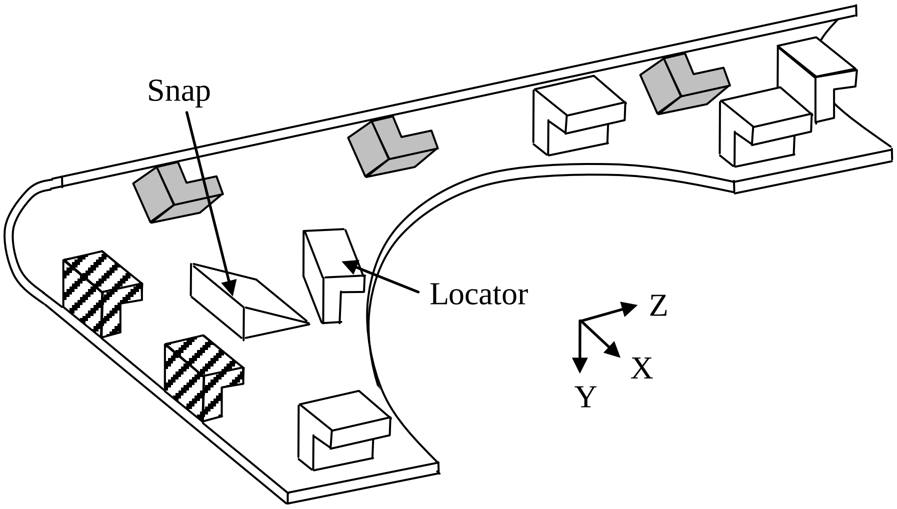


Figure 4.3: Schematic diagram of the orientations of locators and snaps (drawn not in scale) for the fender panel at the representative locations along the mating lines.

- The Assembly sequence is given, based on the selected set of orientations, as to 1) slide the fender vertically downwards, 2) slide the fender horizontally towards the vehicle front.
- The heating region is assumed to be one (p = 1) and of a rectangular shape (v = 4) for simplicity. Thus, one local heating temperature is considered (T_l = 200 ^{o}C).
- The panel is injection-molded; thus, the thicknesses of the locators are kept equal to the panel thickness to avoid undesired injection molding defects such as sink marks on the external side.

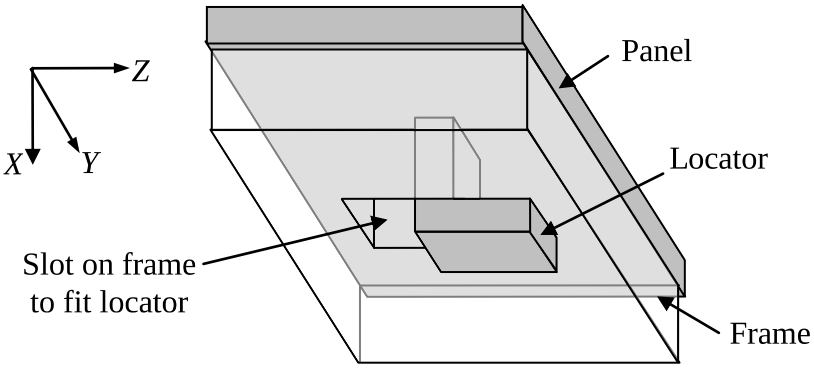


Figure 4.4: Close-up view of a locator on the top edge of the panel, grayed locators in Figure 4.3.

As shown in Figure 4.3, a single, L-shaped, locator type is used. The locators along the curled top edge of the panel (grayed), Figure 4.3, lock into the slots on the frame, Figure 4.2b, rather than wrap around the frame. These locators constrain the panel motion in +Z direction, and in both ±X directions. If the widths of the slots on the frame in the Y direction are equal to the widths of the locators, then these locators can also constrain the panel motion in both ±Y directions as well, Figure 4.4.

Figure 4.5 shows a close-up view of the cross sections of the locators and the frame for the shaded locators and the white locators in Figure 4.3. These locators wrap around the frame to constrain the panel motion in the ±*Y* directions and in direction normal to the locator surface normal to the panel (+z direction in local coordinates as in Figure 4.6).

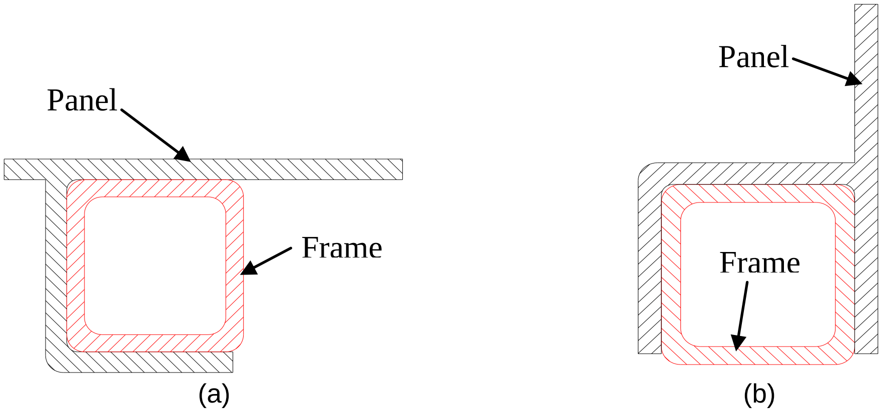
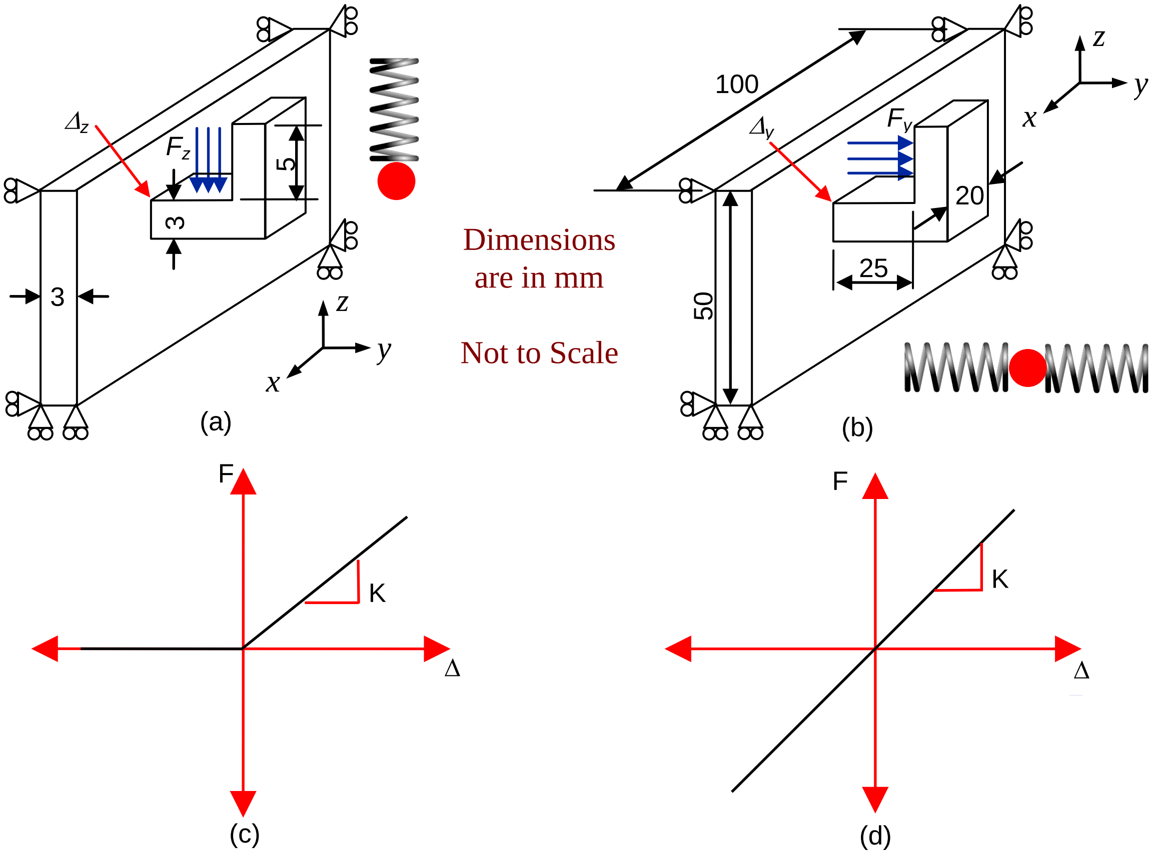


Figure 4.5: Close-up view of the cross sections of the locators in Figure 4.3 and the frame: (a) shaded locators, and (b) white locators.

Figure 4.6a and Figure 4.6b show the dimensions of the locators and show how the locator stiffness is measured. The locators shown constrain the motion in a single in-plane direction (+z), Figure 4.6a, and in the out-of-plane direction (±*y*), Figure 4.6b. Table 4.4 shows the values of the equivalent stiffness for the locators, measured as described in Section 3.2.2.

Table 4.4: Snap and locator stiffness values.

Locator Type	**Stiffness**
z-direction constraining locator (unidirectional)	4972.7 N/mm
y-direction constraining locator (bidirectional)	5192.9 N/mm
Snap	rigid



(a) (b)

Dimensions are in mm

Not to Scale

(c) (d)

Figure 4.6: (a, b) Locator geometry for front fender panel, (c) force-displacement curve for the in-plane unidirectional locator stiffness, and (d) force-displacement curve for the out-of-plane bidirectional locator stiffness.

Based on the above series of assumptions, the problem inputs are given below:

- The geometry of the panel and the frame and the mating lines between them are given in Figure 4.2.
- Due to the panel and frame geometries, the snap used in this case study is single latching. Thus, there is no upper bound on the snap displacement, $h_{upper} = \infty$, whereas, the lower bound on the snap displacement for unlocking is, $h_{lower} = 3$ mm.
- The evaluation of the objective functions requires FEA; thus, the panel will be discretized using a finite element mesh. The possible placement of the locators will

be on the nodal points that lie along the mating line. The total number of nodes along the mating line (possible locator locations) is 126 (n = 126).

- The feasible area for heating is given as the entire panel.

4.2.2 Design objectives

This case study utilizes only two of the objectives given in Section 3.2.4, namely:

1. Minimize the number of locators, thus minimize the complexity of the panel geometry and, consequently, minimize the manufacturing cost.
2. Minimize the heating area(s) required for disassembly, minimizing the amount of energy needed during disassembly, making the disassembly process more economic.

4.2.3 Problem constraints

Apart from the constraints given in Section 3.2.4, there are certain requirements that have to be met in this case study. These requirements are posed as the problem constraints and are given as follows:

1. The natural frequencies of the panel must not fall within the regions given in Table 4.3. This constraint is defined as struct_req(x).
2. Since the locator dimensions in this case study are large compared to the finite element mesh size, another constraint is added to avoid the overlap of neighboring

locators during at the actual manufacturing stage. This is done by applying a penalty if two locators are present in two successive nodal locations. This constraint also allows the modal analysis obtained from FEA to capture the vibration modes between locators.

4.2.4 Design variables

Out of the three design variables ***x***, ***y*** and ***z*** discussed in Section 3.2.4, only two design variables are used in this case study, namely ***x*** and ***y***. Since the orientation of the locators is already defined, the design variable used to select a feasible combination of locators and snaps, ***z***, is ignored.

The total number of variables defined by the design variable ***x*** is 126. Using the definition of ***x*** as given in Section 3.2.4 leads to the following:

$$\begin{aligned}
&\boldsymbol{x} = \left\{x_1, x_2, \cdots, x_{126}\right\} \\
&x_i \in \left\{nil, 1, 2, \cdots, 126\right\} \qquad (4.1) \\
&\Rightarrow \boldsymbol{x} \in \left\{nil, 1, 2, \cdots, 126\right\}^{126}
\end{aligned}$$

If a binary representation of x_i is used with the existence of a locator on a node *i* is defined by (1) and the absence is defined by (0), the range of ***x*** will be smaller, as given in Equation (4.2). As a result, the binary representation will be used in this case study.

$$\boldsymbol{x} = \{x_1, x_2, \cdots, x_{126}\}$$

$$x_i \in \{0,1\} \tag{4.2}$$

$$\Rightarrow \boldsymbol{x} \in \{0,1\}^{126}$$

4.3 Simultaneous optimization of locators/snaps and heating areas

The whole mathematical model is formulated as shown below:

$$\text{minimize } \{ f_1(\boldsymbol{y}),\ f_2(\boldsymbol{x})\}$$

subject to:

$$\text{min\_disp\_loc}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{T}_l) \ > \ h_{lower}$$

$$\text{struct\_req}(\boldsymbol{x}) \ = \ \mathit{true}$$

$$g_1 = \sum_{i=1}^{n-1} x_i x_{i+1} = 0, \quad (x_i \ \& \ x_{i+1}) \in \text{same edge}$$

$$\boldsymbol{x} \in \{0,1\}^{126}$$

$$\boldsymbol{y} \in P_h^4$$

Where:

- $f_2(\boldsymbol{x}) = \sum_{i=1}^{126} x_i$ is the total number of locators
- h_{lower} = 3 mm is the height of the snap plus a small tolerance.

- The local heating temperature, T_l, for the rectangular area = 200 ^{o}C (below melting point 260 ^{o}C of the material used). During heating, the rest of the panel is kept at 20 ^{o}C (Room temperature). The heat is assumed to be transferred to air through free (natural) convection only. The value of the convection heat transfer coefficient for air is chosen to be 8 W/m^{2}.^{o}K.

Since the Aluminum frame carries the majority of the loads in aluminum space frame bodies, the panels' natural frequencies should not lie within the ranges frequencies of the vibration inducing components listed in Table 4.3. Since the first two prohibited frequency ranges are too close to each other (only 1 Hz difference), and the very low frequencies (<5 Hz) can also be prohibited, they are all combined to a one wider prohibited range of < 17 Hz. To avoid that the 14th natural frequency and above do not fall within the prohibited ranges, an extra constraint is added to the 14th natural frequency prohibiting it from being lower than 200 Hz. Accordingly, struct_req($\boldsymbol{x}$) is given as:

$$17 \le \omega_i(\boldsymbol{x}) \le 25 \text{ or } 40 \le \omega_i(\boldsymbol{x}) \le 50 \text{ or } 200 \le \omega_i(\boldsymbol{x}) \text{ and } 200 \le \omega_{14}(\boldsymbol{x}) \tag{4.3}$$

Where $\omega_i(\boldsymbol{x})$, i = 1, 2, …., 14, is the i^{th} natural frequency of the panel attached to the frame (considered as rigid) with the equivalent springs at the locations specified by $\boldsymbol{x}$. Equation (4.4) is augmented in the penalty Equation.

$$\text{srtuct\_req}(\boldsymbol{x}) = true = 0 = \sum_{i=1}^{14} p(\omega_i(\boldsymbol{x})) + \left\{\max(0,\ 200 - \omega_{14}(\boldsymbol{x}))\right\}^2 \tag{4.4}$$

Where:

$$p(\omega_i(\boldsymbol{x})) = \begin{cases} \left(\frac{17}{2}\right)^2 - \left(\frac{17}{2} - \omega_i(\boldsymbol{x})\right)^2 & \text{if } \omega_i(\boldsymbol{x}) < 17 \\ \left(\frac{15}{2}\right)^2 - \left(\frac{65}{2} - \omega_i(\boldsymbol{x})\right)^2 & \text{if } 25 < \omega_i(\boldsymbol{x}) < 40 \\ \left(\frac{150}{2}\right)^2 - \left(\frac{250}{2} - \omega_i(\boldsymbol{x})\right)^2 & \text{if } 50 < \omega_i(\boldsymbol{x}) < 200 \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Table 4.5: GA parameters used in this case study.

Parameter	Value
Population size	130
Number of generations	140
Crossover probability	0.95
Mutation probability	0.05

The optimization problem is solved using a mixed, discrete and continuous, Genetic Algorithm [130],[120]. The percentage of the geometric based crossover [123] and uniform crossover used are 70% and 30%, respectively. The parameters used in GA for this case study are shown in Table 4.5. The Windows workstation used to run the optimization code has a 3.2 GHz Intel Pentium 4 processor with Hyper Threading. The time to finish the whole GA run is 12 days.

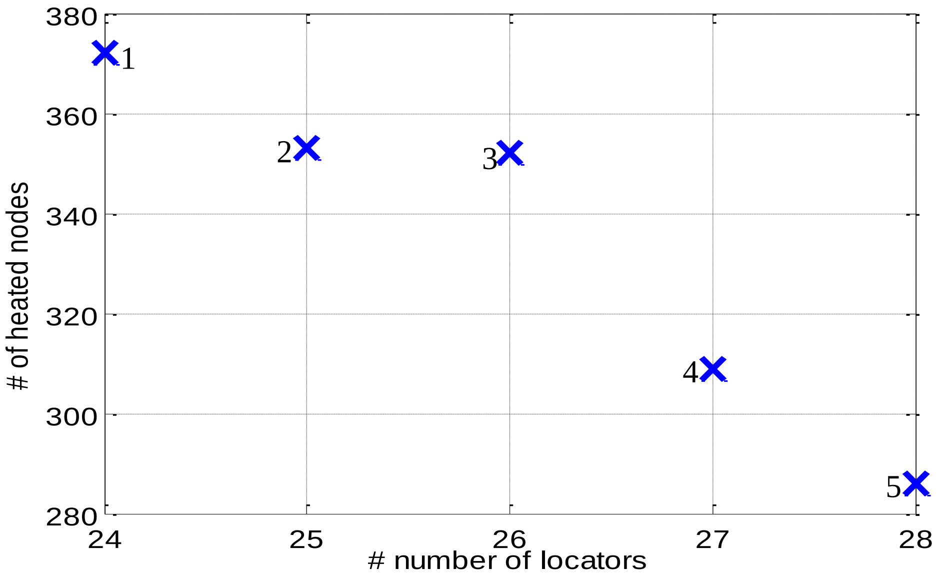


Figure 4.7: Pareto optimal solutions

Table 4.6: Natural frequencies of the fender panel with optimum locators (second column), and with bolted joints at the same location (third column)

Mode number	**Frequency – locators (Hz)**	**Frequency – bolted (Hz)**
1	200.70	252.17
2	208.88	272.18
3	212.16	278.18
4	233.01	283.31
5	245.30	303.37
6	257.78	340.22
7	271.54	344.77
8	274.36	352.74
9	286.16	404.36
10	331.00	417.34

Figure 4.7 shows the Pareto optimal solutions for the number of locators $f_2(\boldsymbol{x})$ and the number of heated nodes $f_1(\boldsymbol{y})$, showing the trade-off between these objectives. All solutions in Figure 4.7 satisfy all the constraints, with the first natural frequency higher

than 200 Hz. The second column in Table 4.6 shows the first 10 natural frequencies for Pareto solution 1 (solution with least number of locators, so least stiff) shown in Figure 4.7. For comparison, the third column of Table 4.6 shows the natural frequencies of the panel attached by bolted joints (i.e., bidirectional rigid connection in the in-plane and out-of-plane directions) at the same location. It can be seen that the frequency values with locators are comparable to the ones with bolted joint, indicating the high rigidity of the proposed heat-reversible locator-snap system of joints.

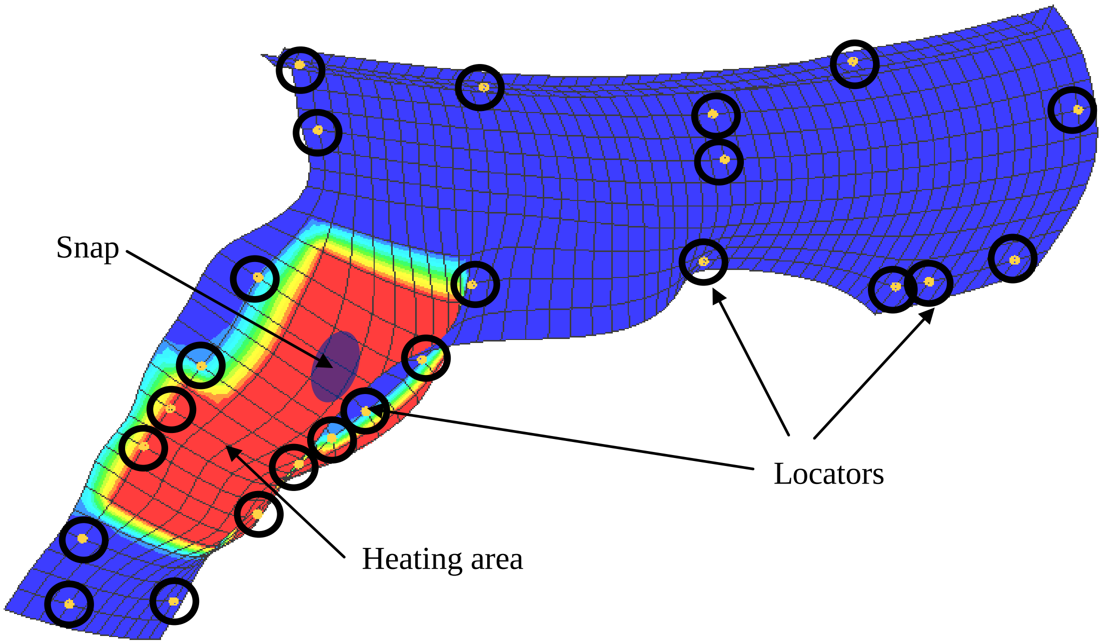


Figure 4.8: Pareto solution with minimum number of locators (24) and heated area (307 X 205 mm^2)

Figure 4.8 shows the deformed shape of Pareto solution 1 with the minimum number of locators. The locators are marked with circles, the heating area is shown in red, and the snap is shown with dark ellipse. The number of locators is twenty four (24) and the heating area is 307x205 mm^2. The maximum and minimum out-of-plane displacements

(Δ_y) along the mating line for snap placement are 5.608 mm and 3.018 mm, respectively. Therefore, single or multiple snaps with 3 mm height can be located along the mating line close to the location of maximum deformation and guarantee unlocking. The in-plane displacements (Δ_x and Δ_z), which might potentially interfere the smooth unlocking of the snap, has the maximum value of 0.45 mm and are negligible compared to the out-of-plane displacement.

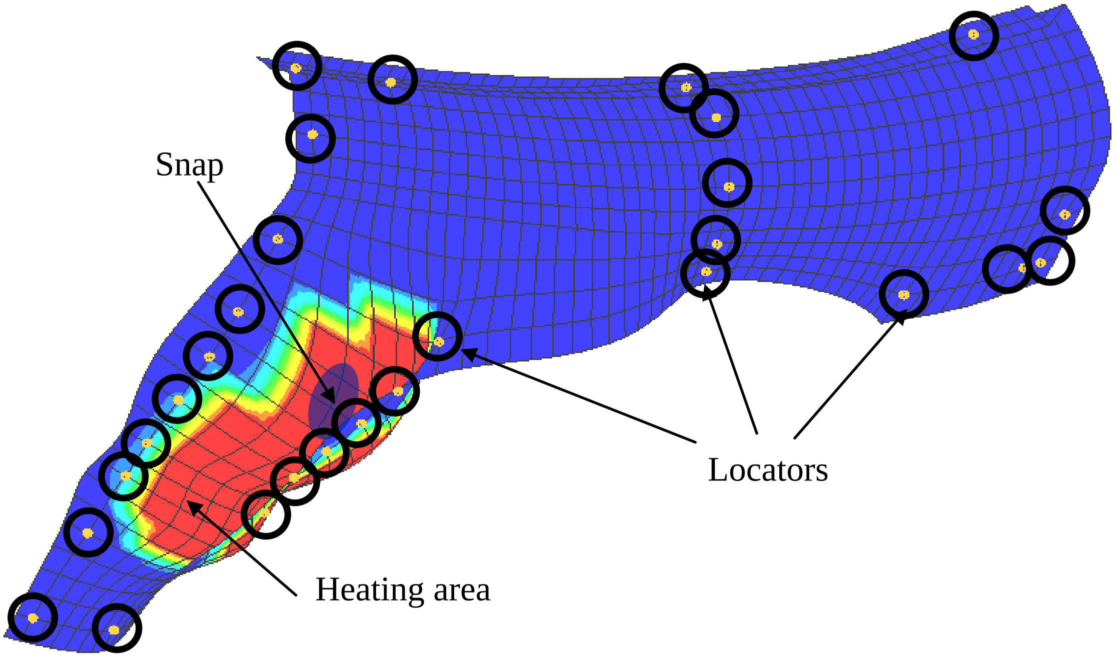


Figure 4.9: Pareto solution with minimum heating area (265X173 mm^2) and number of locators =28.

Similarly, Figure 4.9 shows the locations of locators, heating area, and snap, and the deformed shape of Pareto solution 5 with the minimum heating area. The number of locators is twenty eight (28) and the heating area is 265 x 173mm^2. The maximum and minimum out-of-plane displacements (Δ_y) along the mating line for snap placement are 5.812 mm and 3.006 mm, respectively. Again, single or multiple snaps with 3 mm height can be located at the center of the heated zone and guarantee opening. The in-plane displacements (Δ_x and Δ_z) has the maximum value of 0.48 mm and are negligible compared to the out-of-plane displacement. Figure 4.10 – Figure 4.12 show the results for the remaining Pareto optimal solutions (solutions 2 to 4). The designer can then select from solutions with less locators (thus lower manufacturing cost) and larger heating areas (more energy for disassembly), and solutions with smaller heating areas, yet more locators (higher manufacturing cost).

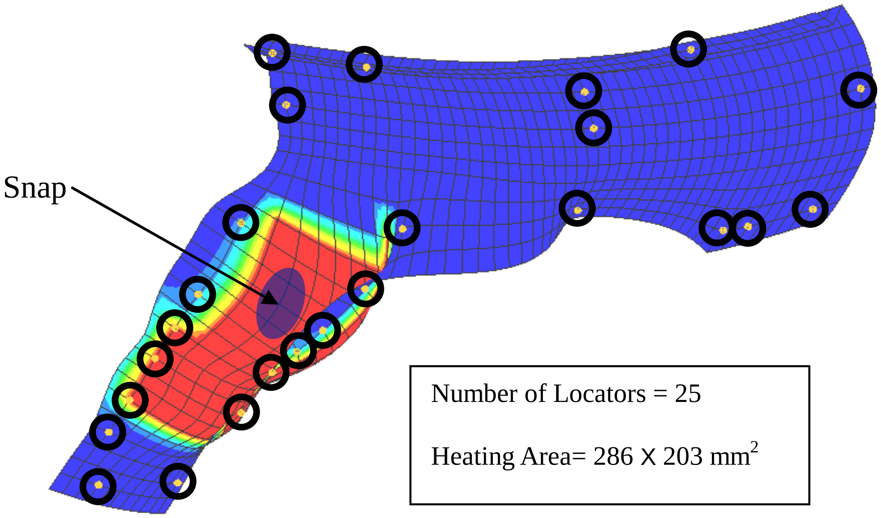


Figure 4.10: Pareto optimum solution 2

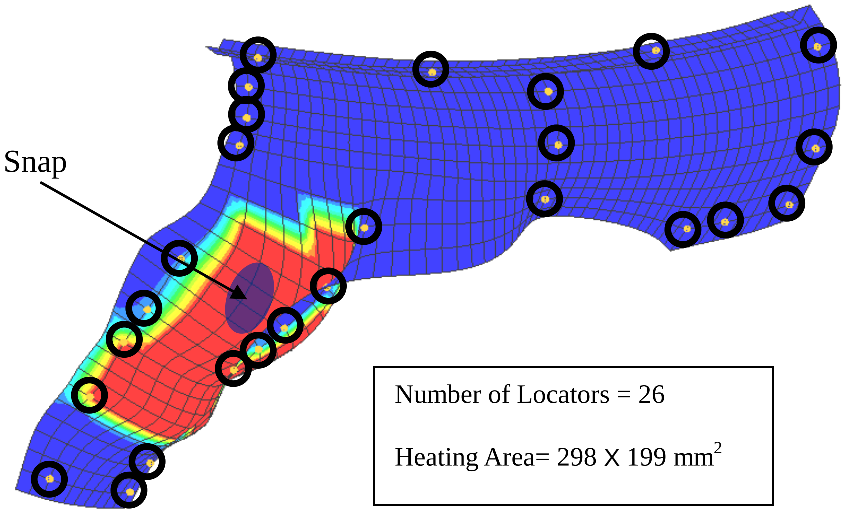


Figure 4.11: Pareto optimum solution 3

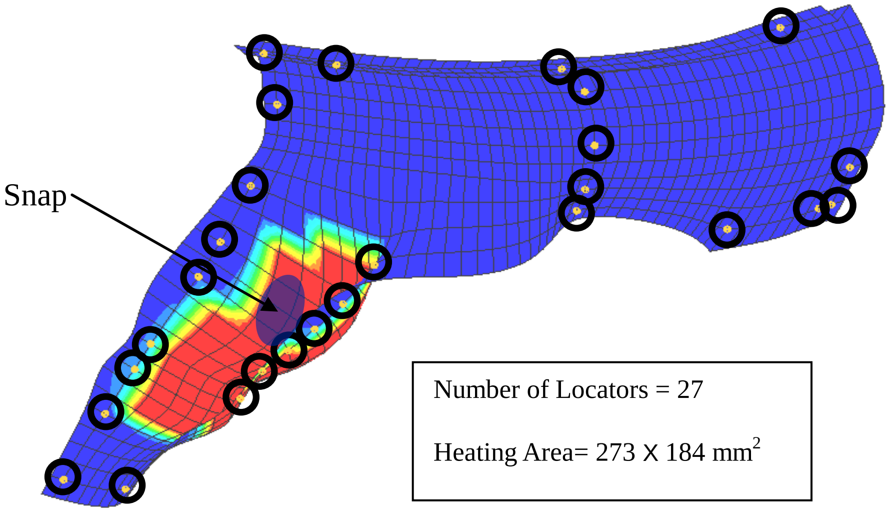


Figure 4.12: Pareto optimum solution 4

4.4 Conclusion

In this case study, heat-reversible locator-snap system is proposed as a joining method between internal frames and external panels in automotive bodies to allow easy, non-destructive, and clean detaching. They are expected to dramatically reduce the end-of-life environmental impacts of aluminum space frame bodies, which currently suffer from poor material recyclability. It should be noted that the resulting fender geometries, with locators and snaps, are extremely complex and require multiple sideways in the mold to manufacture. This adds to the die cost and can be a limiting factor in practically implementing these results.

Chapter 5: Heat-Reversible Locator-Snap System for Consumer Electrics

In this Chapter, the proposed heat-reversible locator-snap system of joints is proposed as a joining method between product enclosures in consumer electrics. The proposed system allows non-destructive and clean separation between the plastic enclosures. This allows easy access to the internal components during repair and maintenance, protection of internal components from destruction during disassembly, and minimizes the amount of different incompatible materials during material recycling.

5.1 Introduction

The rapid economic growth coupled with the urbanization and the growing demand on consumer goods have resulted in the increase in the production of electric and electronic equipments (EEE) and, consequently, their wastes [131]. The reuse and remanufacturing of components and the recycling of the remaining scrap to usable quality materials at the end-of-life are, therefore, important to minimize the resulting wastes. Examples of EEE include computers and its peripherals, phones, PDA, MP3 players, DVD players, televisions, *etc*. Figure 5.1 shows examples of common EEE products

Figure 5.1: Examples of electric and electronic equipment.

The huge increase in Waste electric and electronic equipment (WEEE), one of the fastest growing waste fractions, resulted in legislative and social pressures to drive manufacturers to take responsibilities for reducing the amount of materials that end up in the waste stream at product retirement. Products are now designed with increased emphasis on effective part reuse, remanufacturing and material recycling at the end of product life using the concept of Design for Disassembly. The first step in disassembly is the removal of product enclosures in a fast and efficient way. This needs to be done nondestructively to allow for 1) the access to the internal components during repair and maintenance operations of the products during their use cycle, 2) the protection of internal components from destruction during disassembly; therefore, keep them in a working condition for the reuse and/or remanufacture, and 3) the minimization of the amount of different incompatible materials during material recycling.

Accordingly, in this chapter we will apply the unified method presented in Section 3.2 on external product enclosures of electrical appliances. Two case studies are presented. In the first case study, heat-reversible locator-snap joints are proposed to join a T-shaped DVD player enclosure. In the second case study, to demonstrate the generality of our proposed method, we created a model for an enclosure with complex mating line geometry and used the proposed method for joining the enclosure. The use of double-latching snaps and multiple heating locations demonstrate the lock-and-key concept that makes the disassembly process easy when the right procedure is followed. This is important in electric goods to prevent accidental disassembly during normal use; it also makes the disassembly process hard for unauthorized repair personnel.

5.2 Case Study 1: DVD player enclosure

The method of designing heat-reversible locator-snap system, proposed in Chapter 3, is applied to a simplified DVD player enclosure made of two mating injection molded polystyrene parts. The enclosure geometry is 250×500×150mm with a T-shaped mating line and a wall thickness of 1.5 mm. The material properties are given in Table 5.1. Figure 5.2 shows the simplified model of a casing of the DVD player.

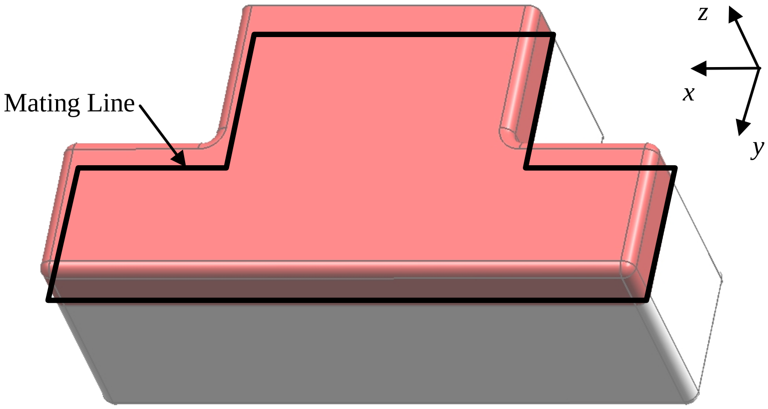


Figure 5.2: Simplified model of case assembly of a DVD player.

Table 5.1: Material properties of polystyrene.

Property Name (units)	Value
Density (g/cm^3)	0.908
Elasticity modulus (MPa)	3110
Poisson Ratio	0.373
Melting point (^{o}C)	240
Thermal expansion coefficient (μm/m.^{o}C)	12.0
Specific heat capacity (j/kg.^{o}C)	1990
Conductivity (W/m.^{o}K)	0.232

5.2.1 Assumptions and inputs

The series of assumptions and inputs given in the given problem are summarized below:

- Figure 5.3 shows the FE model of the lower part of the assembly. The mating polygon has 8 edges.

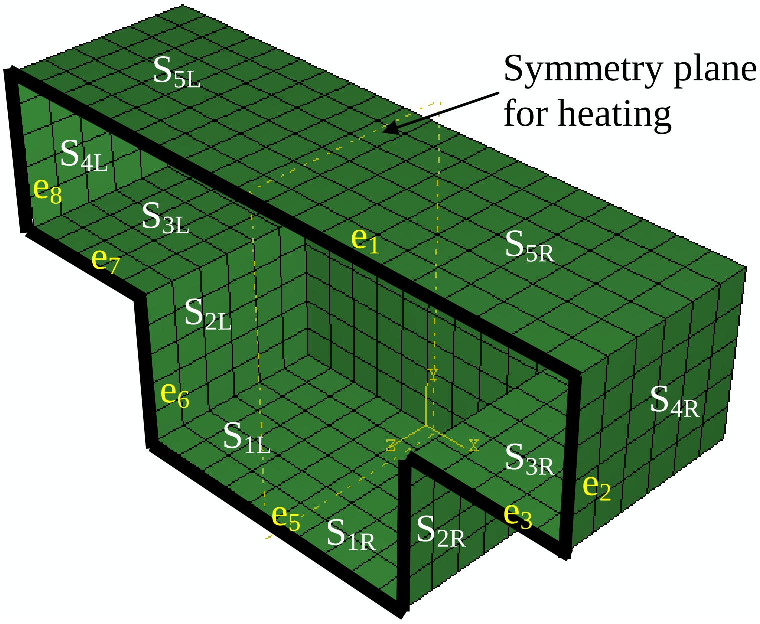


Figure 5.3: FE model of the lower part of assembly showing edges and feasible heating region.

- Each edge has only one snap or one locator type. This results in a total number of apparent edges (n = 8), shown as the thick black lines and labeled as $e_1,\ldots,e_8$ in Figure 5.3.
- The dimensions of the DVD player geometry are relatively small, thus the dimensions of the locators and snaps can be large enough to be considered rigid (in-plane and out-of-plane stiffness are infinite).
- Normal single-latching snaps are used; thus, there is no upper bound on the snap displacement, $h_{upper} = \infty$, whereas, the lower bound on the snap displacement for unlocking is, h_{lower} = 1 mm. h_- is also equal to 1 mm.
- The locator and snap library used in this case study consists of only one locator and one snap, and are shown in Figure 5.4a and Figure 5.4b.

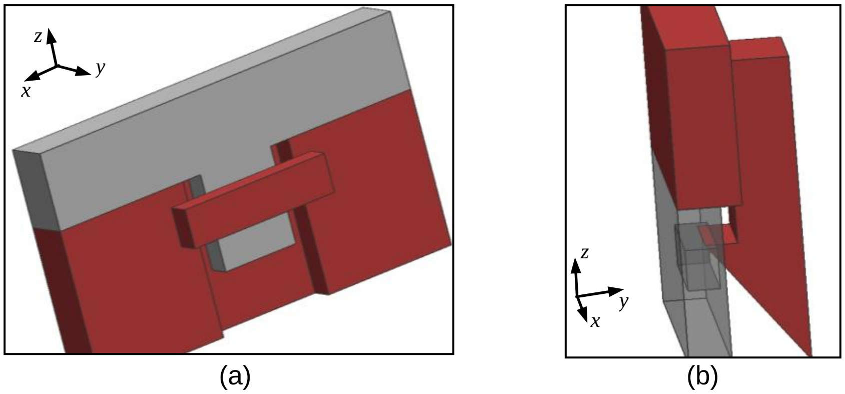


Figure 5.4: Locators and snaps used in the library in the current case study

- The feasible heating region, P_h, is considered as all the 8 surfaces of the lower part except its base surface.
- The temperatures for local heating of all the areas and the uniform heating are T_l = 200 ^{o}C and T_u = 50 ^{o}C, respectively, in a room of 20 ^{o}C.

- During heating, free convection to the air (convection heat transfer coefficient = 8 W/m^2.^{o}K) is considered as the only source of heat dissipation.

5.2.1.1 Definition of the heating area

The feasible heating region, P_h, is actually in a 3D domain. The use of this domain to identify the heating surface is rather complicated; consequently, the heating region is mapped to a simpler 2D domain as described below.

- The feasible heating region is subdivided into ten (10) sub-surfaces (labeled as $S_{1L}\ldots S_{5L}$ and $S_{1R}\ldots S_{5R}$), five (5) on each side of the plane of symmetry, as shown in Figure 5.3.
- The feasible heating regions of the two symmetric halves of the enclosure are, then, flattened to create the 2D rectangular regions as shown in Figure 5.5a.
- The coordinates of the heating area, ***y***, are defined in the 2D domain and then transformed to the 3D domain.
- Making use of the symmetry, heat can be applied on the right side, on the left side or on both sides. Instead of utilizing variable ***y*** to explicitly define 2 heating areas, one for each side, the case study considers two rectangular heating areas by using ***y*** with m = 1 and an auxiliary design variable t defined as:

$$t = \begin{cases} 0 & \text{if only left (L) side is heated} \\ 1 & \text{if only right (R) side is heated} \\ 2 & \text{if both sides are heated} \end{cases} \tag{5.1}$$

where the left (L) and right (R) sides are with respect to the plane of symmetry as defined in Figure 5.3. A sample heated area is shown in Figure 5.5a and its equivalent area(s) in the 3D model are shown in Figure 5.5b, c and d for t = 0, t = 1 and t = 2 respectively.

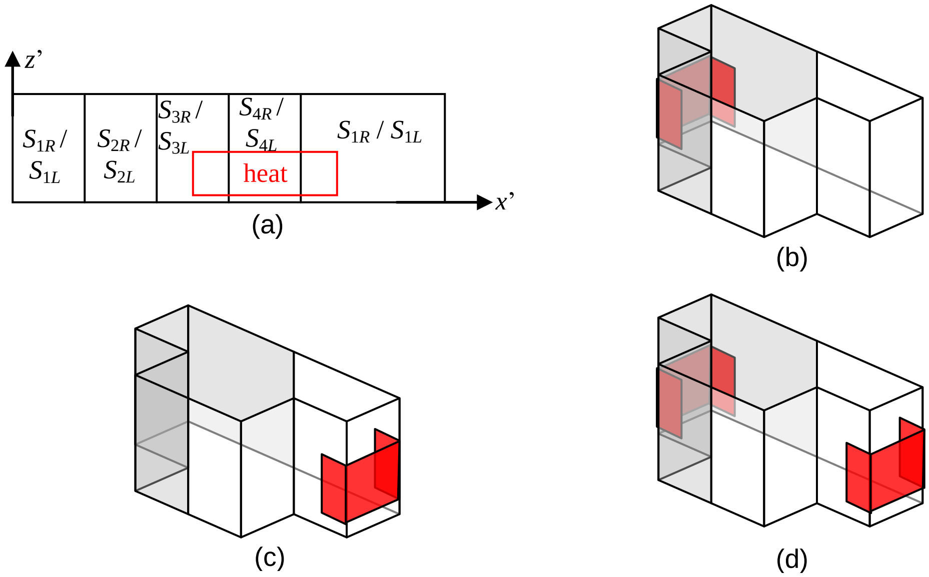


Figure 5.5: (a) Heating area in the flattened feasible heating sub-regions S_{1R}-S_{5R} and S_{1L}-S_{5L}, (b) corresponding heating area in 3D for t = 0, (c) corresponding heating area in 3D for t = 1, and (d) corresponding heating area in 3D for t = 2.

5.2.1.2 Generation of the feasible locator and snap orientations

The set of feasible orientations of locator and snaps is pre-calculated as discussed in Section 3.2.3. The wrench matrices of the locator library in the local coordinate system are given in Equations (5.2) and (5.3). Only the orientations shown in Figure 5.4 are considered, i.e. the locators and snaps are not allowed to change their orientations with respect to the mating line they are attached to.

$$\mathbf{W}_{l_1}^{local} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{5.2}$$

$$\mathbf{W}_{s_1}^{local} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \tag{5.3}$$

Knowing that the total number of edges (n = 8) and the total number of locator and snap types (m = 2), the total number of possible locator and snap combinations is (2^8 = 256). After testing all these combinations using Equations (3.17) and (3.18), only 224 combinations are feasible and are included in the feasible set F. In all cases, the assembly direction is to move the two parts toward each other in the z direction in Figure 5.2.

5.2.2 Simultaneous optimization of locators/snaps and heating areas

The whole mathematical model is formulated as shown below:

$$\text{minimize } \left\{ f_1(\boldsymbol{y}),\ f_2(\boldsymbol{x},\boldsymbol{z}) \right\}$$

subject to:

$$\text{min\_disp\_loc}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},T_l) > h_{lower}$$

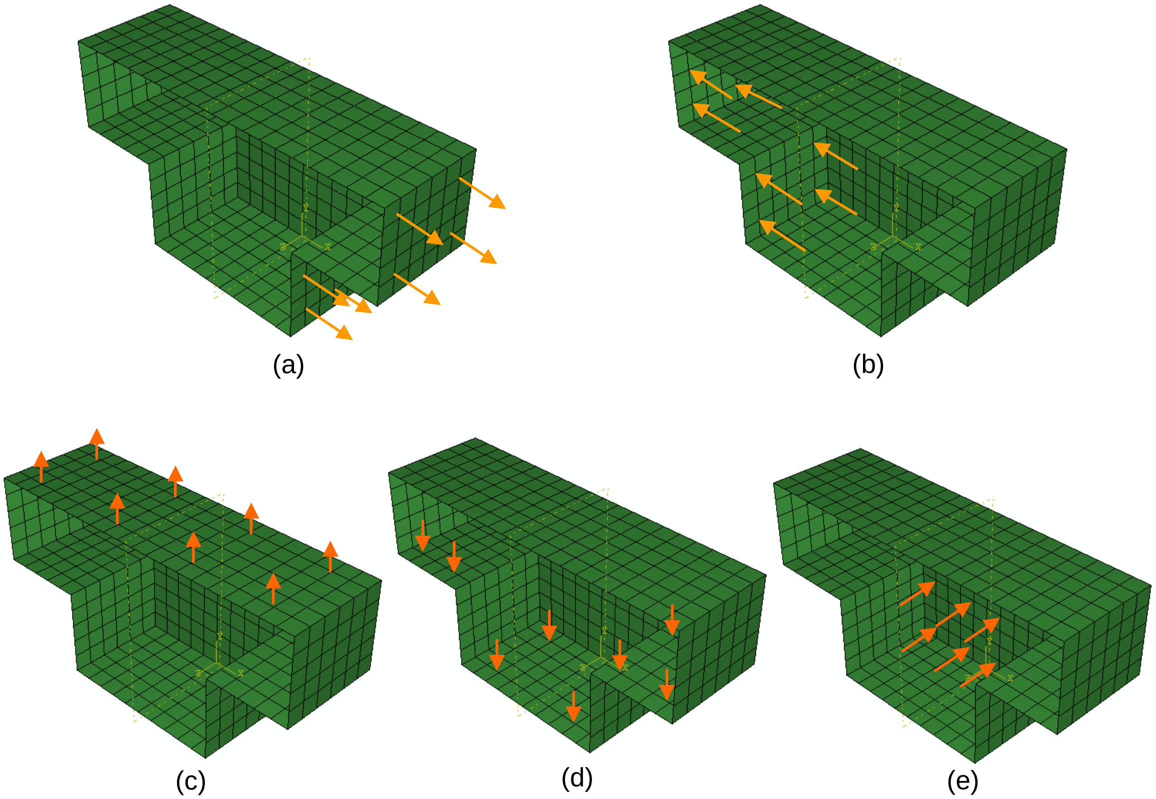
$$\text{max\_disp\_unif}(\boldsymbol{x},\boldsymbol{z},T_u) < h\_$$

$$x_{ij} \in \left\{ nil, L_i, L_i+1, \cdots, U_i \right\};\ i = 1,\cdots,8;\ j = 1,\cdots,d_i$$

$$\boldsymbol{y} \in P_h^4$$

$$\boldsymbol{z} \in F$$

The number of heated nodes is taken as a measure for the heated area to calculate $f_1(\boldsymbol{y})$. $f_2(\boldsymbol{x}, \boldsymbol{z})$ is a measure for the stiffness of the assembly, and is calculated by applying a uniformly distributed load of 20 N to the surface of the enclosure in ±x (Figure 5.6a and b), ±y (Figure 5.6c and d), and $-z$ (Figure 5.6e) directions. The maximum nodal displacement of the mating line for all loadings is, then, obtained. Since there is only one local heating temperature for all the heating areas, T_l is just a single variable and is not a vector.



(a)

(b)

(c)

(d)

(e)

Figure 5.6: Measuring the stiffness of the assembly: (a) uniform force in +x (b) uniform force in -x, (c) uniform force in +y, (d) uniform force in -y, and (e) uniform force in -z.

A penalty is applied if the minimum displacement of all nodes with snaps under local heating at T_l = 200°C is less than h_{lower} = 1 mm. Another penalty is applied if the maximum displacement of all nodes with snaps under uniform heating T_u = 50 ^{o}C is more than h_{-} = 1 mm. The displacement constraints are written as penalty functions as given below:

$$g_1\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},T_l\right)=\max\left(0,h_{lower}-\text{min\_disp\_loc}\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},T_l\right)\right) \tag{5.4}$$

$$g_2\left(\boldsymbol{x},\boldsymbol{z},T_u\right)=\max\left(0,\text{max\_disp\_unif}\left(\boldsymbol{x},\boldsymbol{z},T_u\right)-h_{-}\right) \tag{5.5}$$

Since the DVD player geometry consists of a series of rectangular surfaces, the finite element mesh is made uniform for simplicity. The heating area, therefore, can be defined using discrete coordinate variables instead of using real variables. The optimization problem is, thus, solved using only discrete GA. The parameters used in GA for this case study are shown in Table 5.2. A summary of the snap bounds and the heating temperature values is given in Table 5.3.

Table 5.2: GA parameters used in the case study

Parameter	**Value**
Population size	80
Number of generations	160
Crossover probability	0.95
Mutation probability	0.05

Table 5.3: Summary of the snap bounds and the heating temperature values.

Parameter	Value
h_{lower}	1.0 mm
h_{upper}	∞
h_{-}	1.0 mm
T_l	200 ^{o}C
T_u	50 ^{o}C

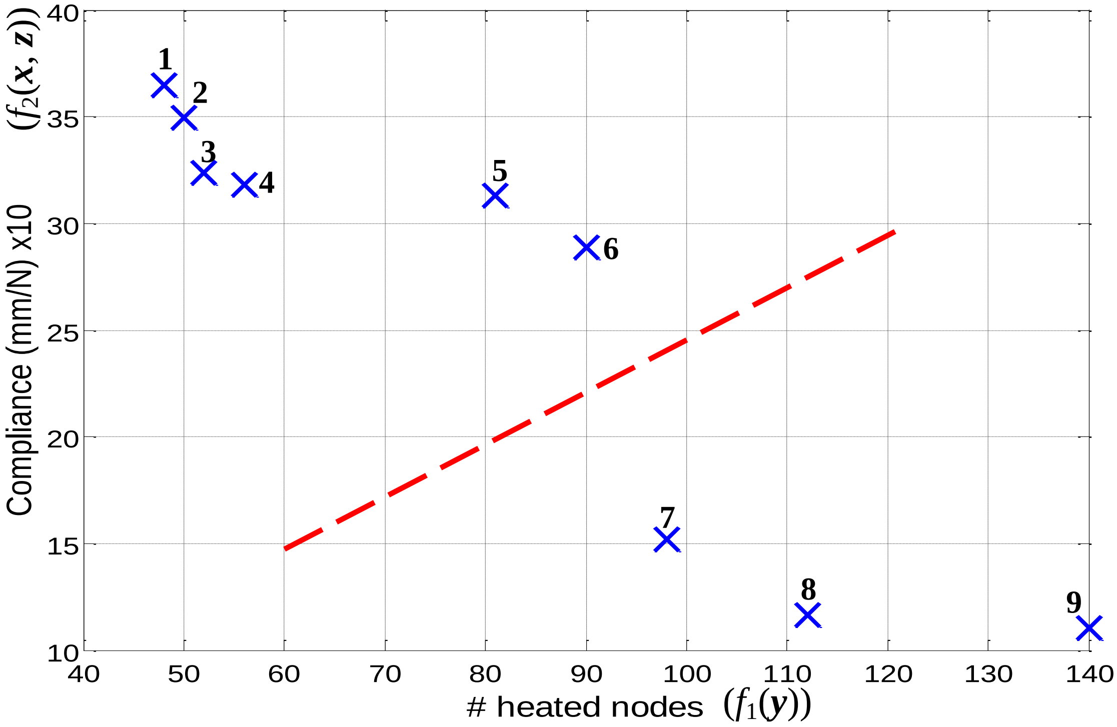


Figure 5.7: Pareto optimal solutions for the case study.

The optimization problem is solved using NSGA-II code running on a Windows workstation with a 3.4 GHz dual core Intel Pentium 4 processor. The time to finish the whole GA run is 3 days. Figure 5.7 shows the Pareto Optimal solutions, which exhibit a trade-off between the part compliance and the amount of heating required (number of heated nodes). The solutions above the dotted line use single heating area on one side of the DVD (t = 0 or 1), while the solutions below the line use symmetric heating (t = 2). Solutions using symmetric heating can be used as a more complex lock-and-key method than heating just one area. To realize so, both areas have to be heated at the same time, or snaps will remain locked. It should be noted that excessive heating can result in accidental opening of the snaps as single-latch snaps are used.

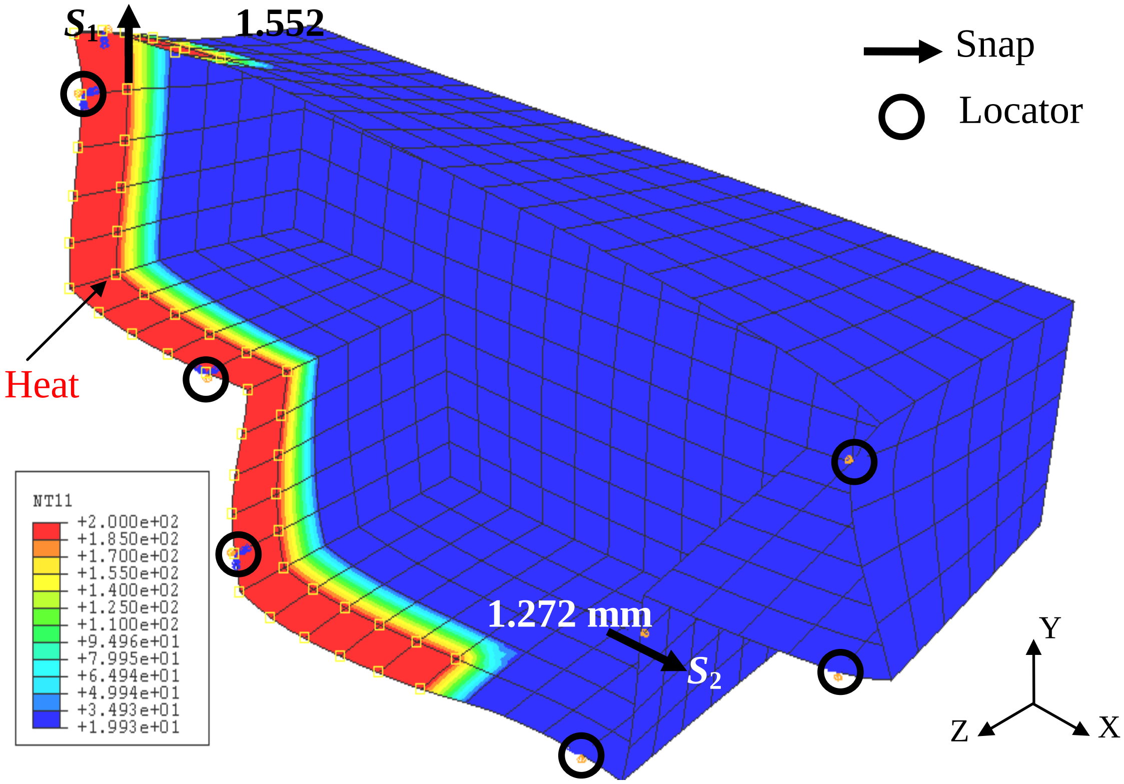


Figure 5.8: Optimum solution with minimum heat area (solution 1)

Figure 5.8 shows the optimal placement of locators and snaps and the response due to local heating of the solution with minimum heating area (solution 1). The heating area is 25 mm × 575 mm (48 nodes). Locator positions are marked with black circles, while snap positions are marked with arrows showing the bulging direction as shown in Figure 5.8. The bulging at both snap locations (s_1 and s_2) is outward and is equal to 1.552 mm and

1.272 mm respectively. Two catches are placed on the shown part at the optimized snap locations; while the actual snaps are placed on the other part (not shown). The maximum deformation at the mating line under the DVD's own weight is due to pressure load in the +*y* direction at the upper surface and is equal to (0.7292 mm). The deformations at s_1 and s_2 due to uniform heating are 0.538 mm and 0.258 mm respectively. Figure 5.9a and Figure 5.9b show CAD drawings of top cover and base part of the final optimized DVD player model for optimum solution 1.

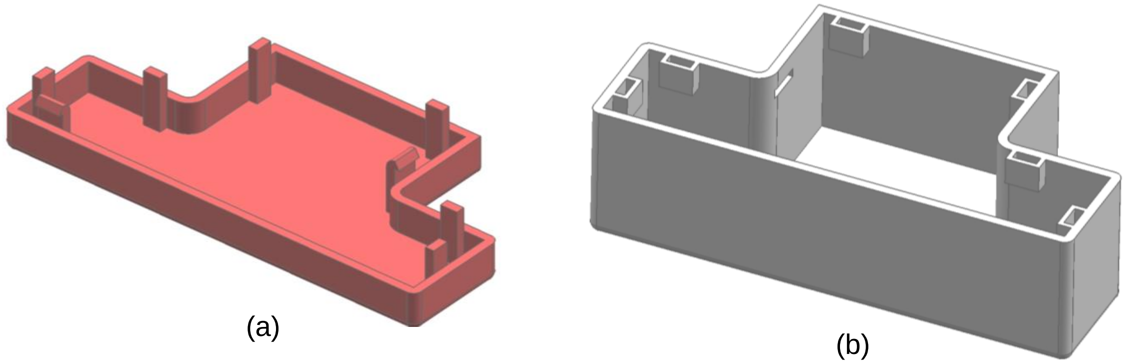


Figure 5.9: CAD drawing for the optimized DVD model (solution 1) (a) top part, and (b) base part.

Figure 5.10 shows the optimal placement of locators and snaps and the response due to local heating of the optimum solution with minimum symmetric heating areas (solution 7). The heating area is 150 mm × 150 mm × 2 (98 nodes). The bulging at snap location (s_1) is outward and is equal to 1.087 mm. A catch is placed on the shown part at; while the actual snap is placed on the other part (not shown). On the other hand, the bulging at snap location (s_2) is inward and is equal to 1.031 mm; thus the actual snap is placed on the shown part at; while the catch is placed on the other part (not shown). The maximum deformation at the mating line under the DVD's own weight is due to pressure load in the

–*z* direction at the bottom surface and is equal (0.3038 mm). The deformations at s_1 and s_2 due to uniform heating are 0.134 mm and 0.550 mm respectively. Figure 5.11a and Figure 5.11b show CAD drawings of top cover and base part of the final optimized DVD player model for optimum solution 7. The remaining Pareto optimum results are shown in Figure 5.13 to Figure 5.19.

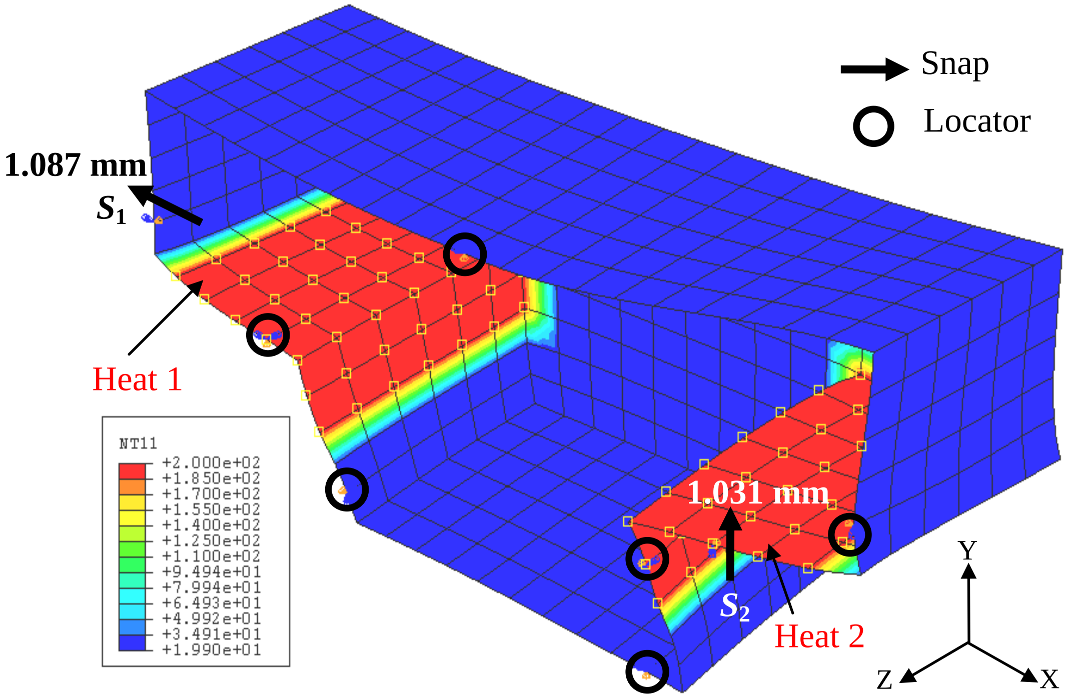


Figure 5.10: Optimum solution with minimum symmetric heat area (solution 7)

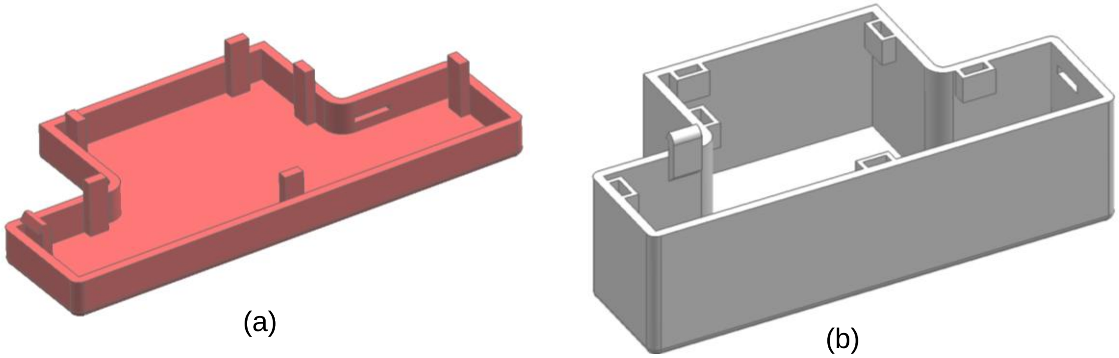
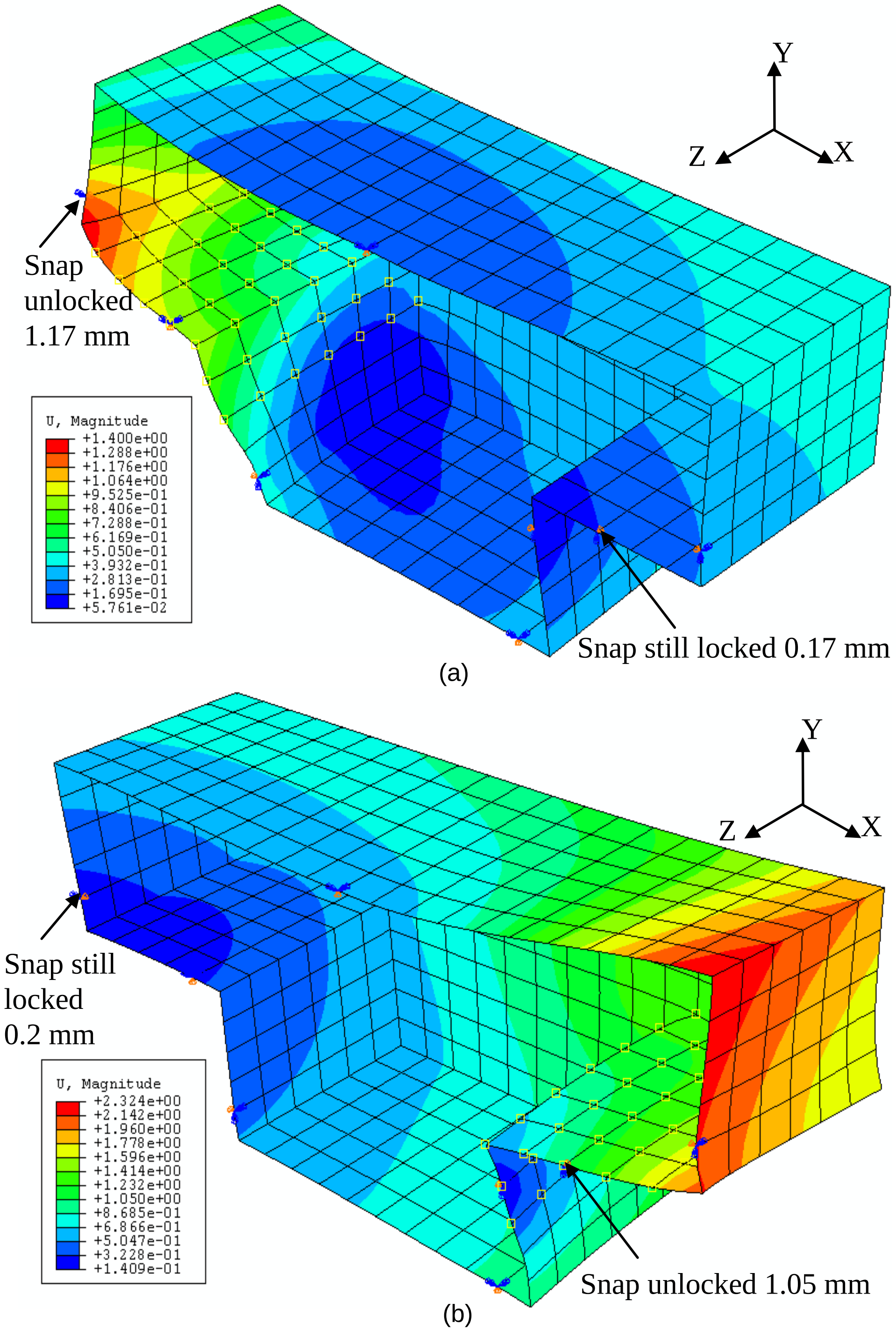


Figure 5.11: CAD drawing for the optimized DVD model (solution 7) (a) top part, and (b) base part.



(a)

(b)

Figure 5.12: Solution 7 response to one sided heating, (a) area 1 is heated, and (b) area 2 is heated.

The use of two heating areas in the solution in Figure 5.10 is considered a more complex lock-and-key method. Single heating area is not enough to unlock the snaps; both areas have to be heated at the same time to ensure snap unlocking. If only the left area is heated, the left snap will unlock while the other snap will remain closed (1.17 mm and 0.17 mm respectively) as shown in Figure 5.12a. Similarly, if only the right area is heated, the right snap will unlock while the other snap will remain closed (1.05 mm and 0.2 mm respectively) as shown in Figure 5.12b. As a result, both sides need to be heated simultaneously to allow unlocking.

5.2.3 Conclusion

This case study proposed the use of the unified method for designing a high-stiffness reversible locator-snap system on DVD player enclosure with a T-shaped mating line. The resulting Pareto-optimal solutions exhibit alternative designs with different trade-offs between structural stiffness during snap engagement and heating area necessary for snap disengagement. Although design for a complex lock-and-key method was not an objective, for simplicity, some results required simultaneous heating of two areas, demonstrating a simple idea of the lock-and-key concept.

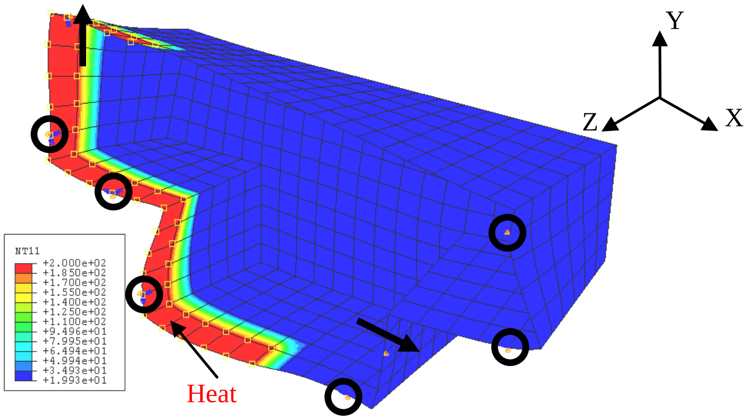


Figure 5.13: Optimum solution #2

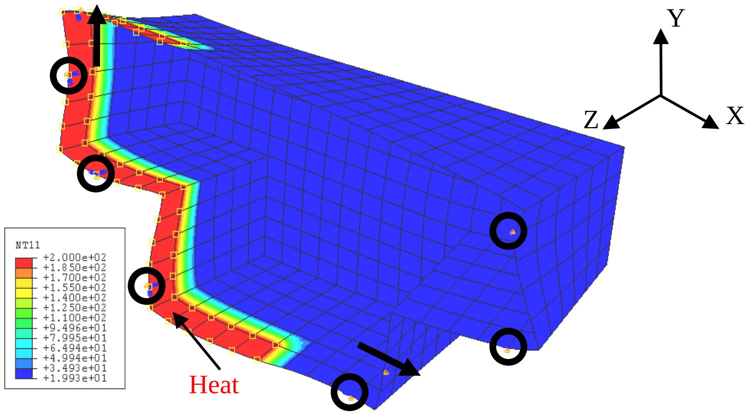


Figure 5.14: Optimum solution #3

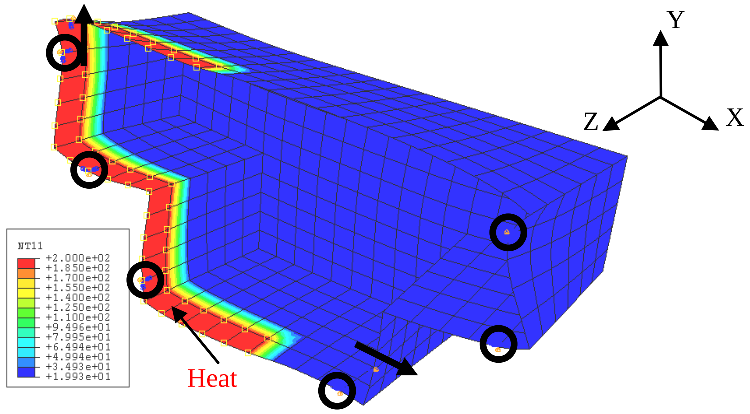


Figure 5.15: Optimum solution #4

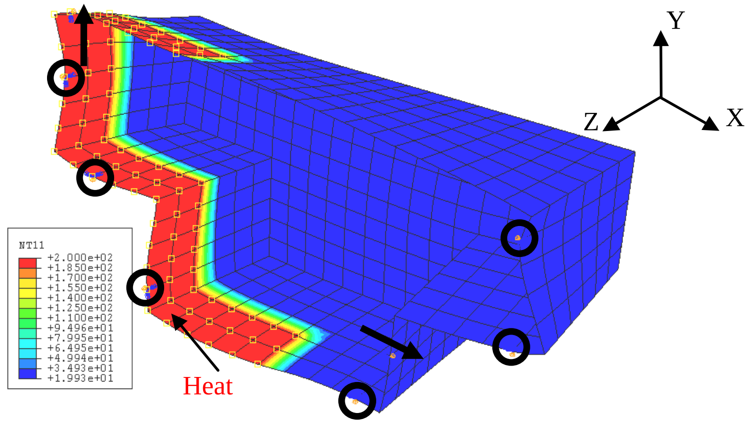


Figure 5.16: Optimum solution #5

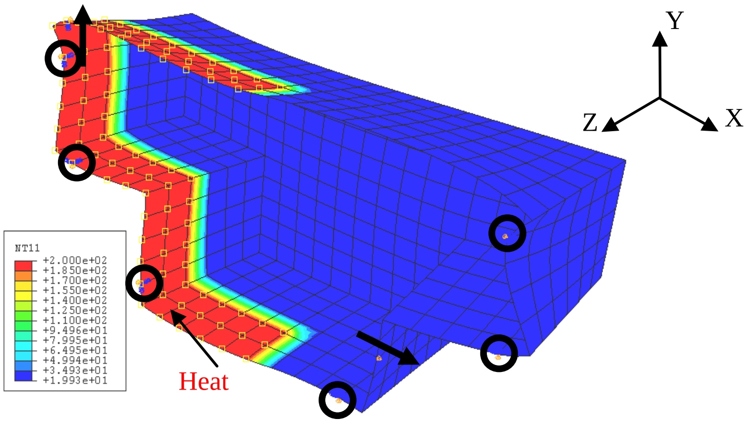


Figure 5.17: Optimum solution #6

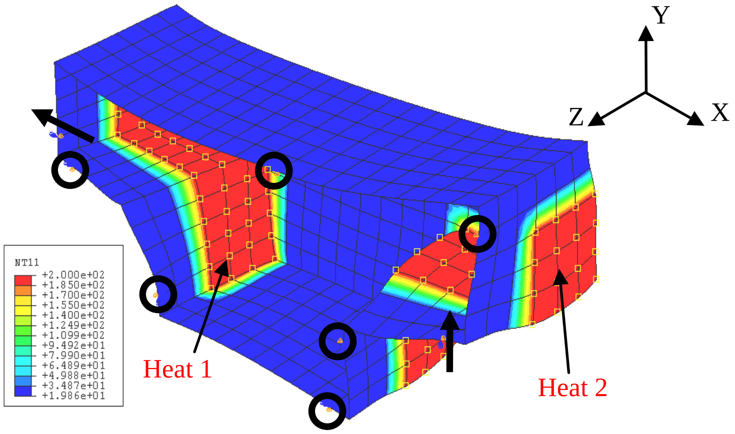


Figure 5.18: Optimum solution #8

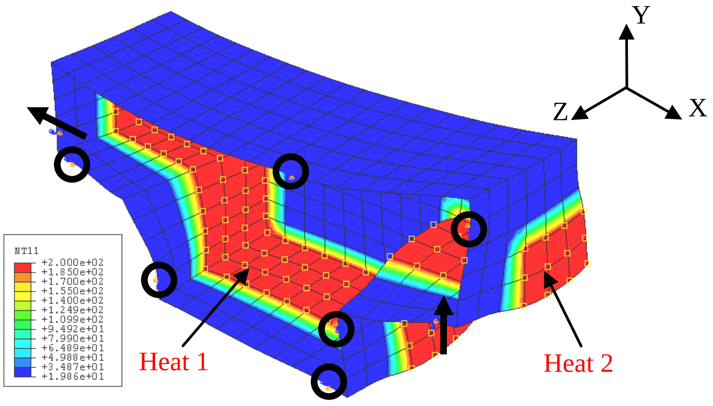


Figure 5.19: Optimum solution #9

5.3 Case Study 2: Enclosure with complex mating line geometry

In this case study, the method is applied to a conceptual rhombus shaped enclosure with a complex mating line to demonstrate the generality of the concept. Similar to the previous case study, the enclosure is assumed to be made of injection molded polystyrene. This case study will include design for the lock-and-key concept using multiple heating locations at two different temperatures and the use of the double-latching snaps. It will also include the maximization of the distance between locators that constrain same DOFs to minimize the effect of tolerance stack-up, as mentioned in Chapter 3. The rhombus geometry is 150×150×125 mm with a wall thickness of 1.5 mm. The material properties are given earlier in Table 5.1. Figure 5.20 shows the simplified model of the enclosure assembly.

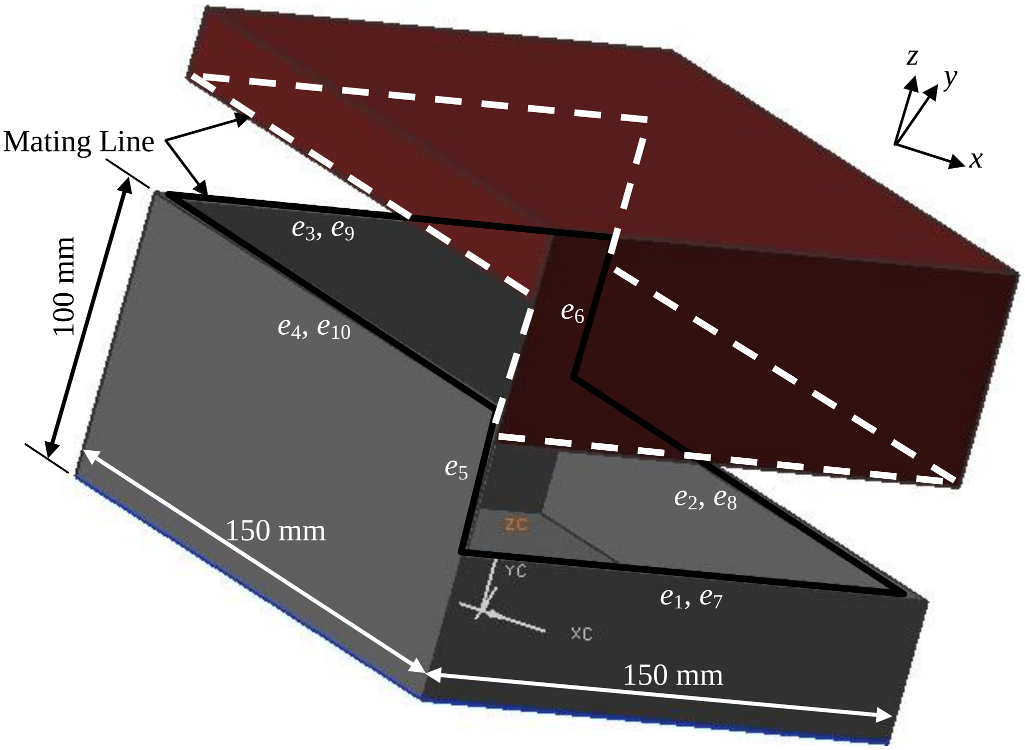


Figure 5.20: Simplified model of case assembly of the rhombus enclosure.

5.3.1 Assumptions and inputs

The series of assumptions and inputs for the given problem are summarized below:

- Each edge of the four edges defining the rhombus (longer edges) can have two types of locators and/or snaps. Whereas, each edge of the vertical edges (shorter edges can have only one type of locator or a snap. This results in a total number of apparent edges (n = 10).
- The dimensions of the enclosure geometry are relatively small, thus the dimensions of the locators and snaps can be large enough to be considered rigid (in-plane and out-of-plane stiffness are infinite).
- The proposed double-latching snaps are used. The upper bound of the snap displacement, h_{upper}, is 1.5 mm, whereas the lower bound of the snap displacement for unlocking, h_{lower}, is 1 mm. h_{-} is also equal to 1 mm.
- The feasible heating region, P_h, is considered as all the 4 surfaces of the lower part, Figure 5.22a except its base surface.
- Two local heating regions are used (p = 2). Each has rectangular shape (v = 4).
- Heating regions are not allowed to overlap.
- One of the local heating regions is heated to T_{l1} = 120^{o}C, while the other local heating region is cooled to T_{l2} = –80^{o}C. Thus, the local temperature vector, $\boldsymbol{T_l}$, has two entries. The uniform heating temperature is T_u=50^{o}C. Room temperature is 20^{o}C.
- During heating, free convection to the air (convection heat transfer coefficient = 8 W/m^{2}.^{o}K) is considered as the only source of heat dissipation.

- The locator and snap library used in this case study consists of only three locators and one snap as shown in Figure 5.21.

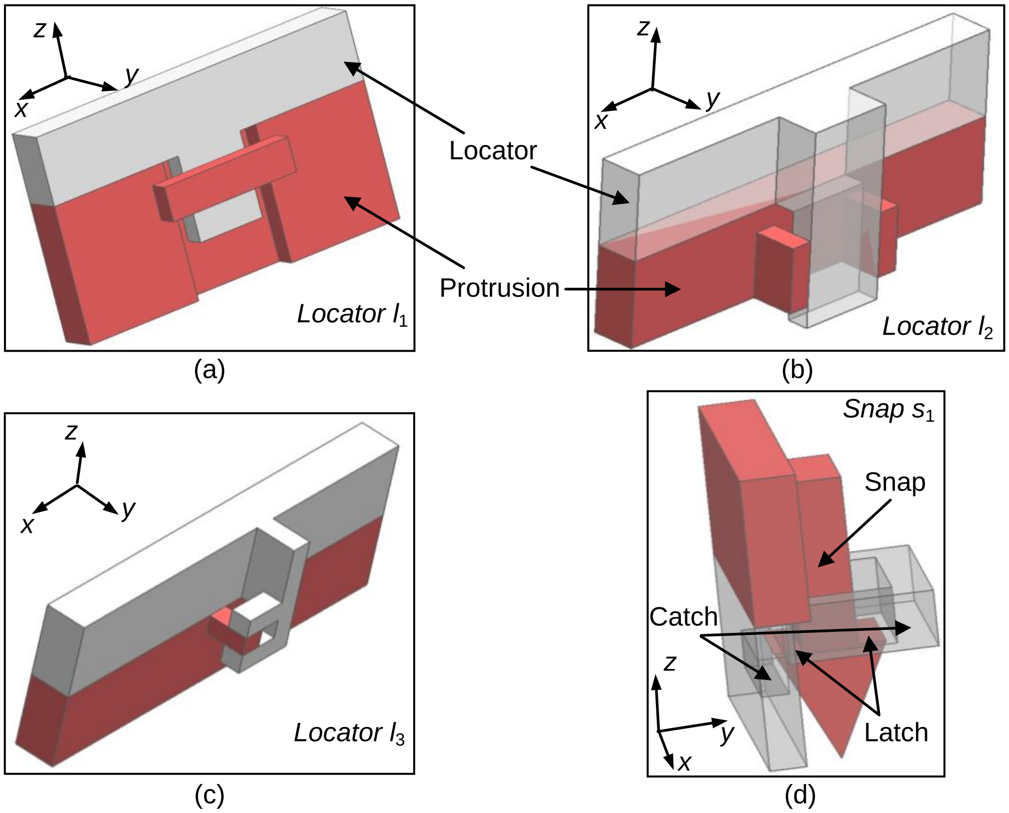


Figure 5.21: Locators and snaps used in the library in the current case study

5.3.1.1 <u>Definition of the heating area</u>

Using a similar mapping to what was done in the previous case study, the feasible heating region, P_h, is transformed from the 3D domain to a simpler 2D domain using the procedure below:

- The rhombus surface has four side surfaces, $S_1,\ldots,S_4$, Figure 5.22a. Each side is flattened to create 2D rectangular regions as shown in Figure 5.22b.

- The coordinates of the heating areas, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are defined in the 2D domain and then transformed to the 3D domain.

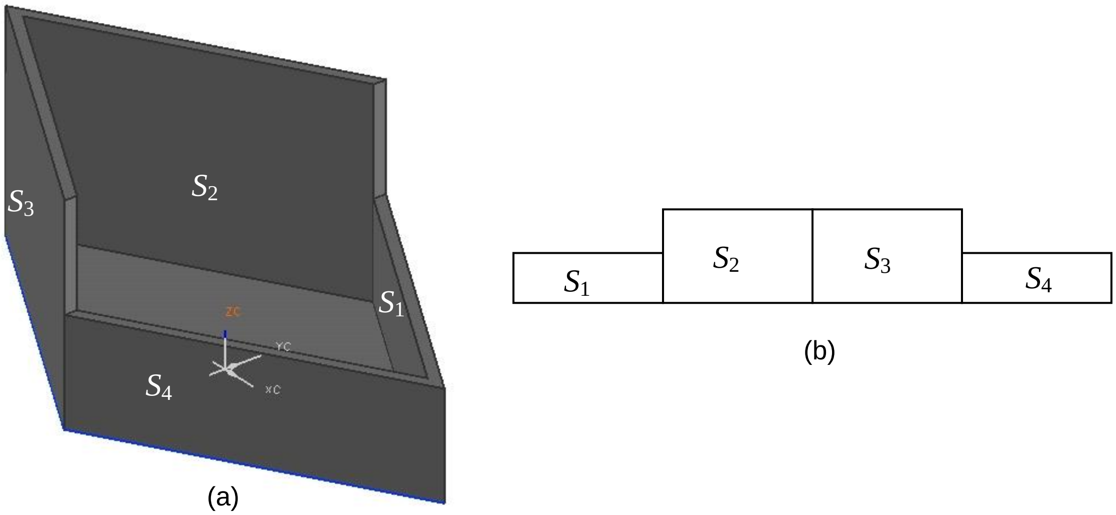


Figure 5.22: (a) CAD model for the lower enclosure only with the 4 sides of the rhombus numbered $S_1,\ldots,S_4$, and (b) flattened 2D surfaces of the rhombus.

5.3.1.2 Generation of the feasible locator and snap orientations

The set of feasible orientations of locators and snaps is pre-calculated as discussed in Section 3.2.3. The wrench matrices in the local coordinate systems are given in Equations (5.6)-(5.9). While the local-global transformation matrices for each edge are given in Equations (5.10)-(5.15). Only the orientations shown in Figure 5.21 are considered, i.e. the locators and snaps are not allowed to change their orientations with respect to the mating line they are attached to.

$$\mathbf{W}_{l_1}^{local} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{5.6}$$

$$\mathbf{W}_{l_2}^{local} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{5.7}$$

$$\mathbf{W}_{l_3}^{local} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \tag{5.8}$$

$$\mathbf{W}_{s_1}^{local} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \tag{5.9}$$

$$\mathbf{R}_{e1,e7} = \begin{pmatrix} 0.866 & 0.5 & 0 \\ -0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.10}$$

$$\mathbf{R}_{e2,e8} = \begin{pmatrix} -0.866 & 0.5 & 0 \\ -0.5 & -0.866 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.11}$$

$$\mathbf{R}_{e3,e9} = \begin{pmatrix} -0.866 & -0.5 & 0 \\ 0.5 & -0.866 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.12}$$

$$\mathbf{R}_{e4,e10} = \begin{pmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.13}$$

$$\mathbf{R}_{e5} = \begin{pmatrix} 0 & 0 & 1 \\ 0.5 & -0.866 & 0 \\ 0.866 & 0.5 & 0 \end{pmatrix} \tag{5.14}$$

$$\mathbf{R}_{e6} = \begin{pmatrix} 0 & 0 & 1 \\ 0.5 & 0.866 & 0 \\ 0.866 & -0.5 & 0 \end{pmatrix} \tag{5.15}$$

Knowing that the total number of edges (n = 10) and the total number of locator and snap types (m = 4), the total number of possible locator and snap combinations is (4^{10} = 1048576). After testing all these combinations using Equations (3.17) and (3.18), only 25254 are feasible and are included in the feasible set F. It is obvious that checking the feasibility prior to running the optimization is really crucial in this case study as only 2.4% of the total possible combinations are feasible; otherwise, a huge part of the design space will be infeasible during the optimization.

5.3.2 Simultaneous optimization of locators/snaps and heating areas

The whole mathematical model is formulated as shown below:

$$\text{minimize } \left\{ f_1(\boldsymbol{y}),\ f_2(\boldsymbol{x},\boldsymbol{z}),\ f_3(\boldsymbol{x},\boldsymbol{z}) \right\}$$

subject to:

$$\text{min\_disp\_loc}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l) \ > \ h_{lower}$$

$$\text{max\_disp\_loc}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l) \ < \ h_{upper}$$

$$\text{max\_disp\_unif}(\boldsymbol{x},\boldsymbol{z},T_u) \ < \ h_{\_}$$

$$x_{ij} \in \left\{ nil, L_i, L_i+1, \cdots, U_i \right\};\ i = 1,\cdots,10;\ j = 1,\cdots,d_i$$

$$\boldsymbol{y}_i \in P_h^4;\ i = 1,2$$

$$\boldsymbol{z} \in F$$

The difference between the formulation in this case study and the formulation in the previous case study is the inclusion of the "max_disp_loc($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{T}_l$)" constraint. This constraint appears in this formulation because of the usage double latching snaps. For the calculation of $f_1(\boldsymbol{y})$, the number of heated nodes is taken as a measure for the heated area. For the calculation of $f_2(\boldsymbol{x}, \boldsymbol{z})$, which is a measure for the stiffness of the assembly, a uniformly distributed load of 20 N is applied to the surfaces of the bottom enclosure part normal the 5 surfaces defining the enclosure, one at a time (Figure 5.23). The maximum displacement of the nodes on the mating line for all loadings is, then, obtained. $f_3(\boldsymbol{x}, \boldsymbol{z})$, is the reciprocal of the minimum the distance between locators/snaps constraining the same DOFs. It is considered as a measure of the tolerance stack-up.

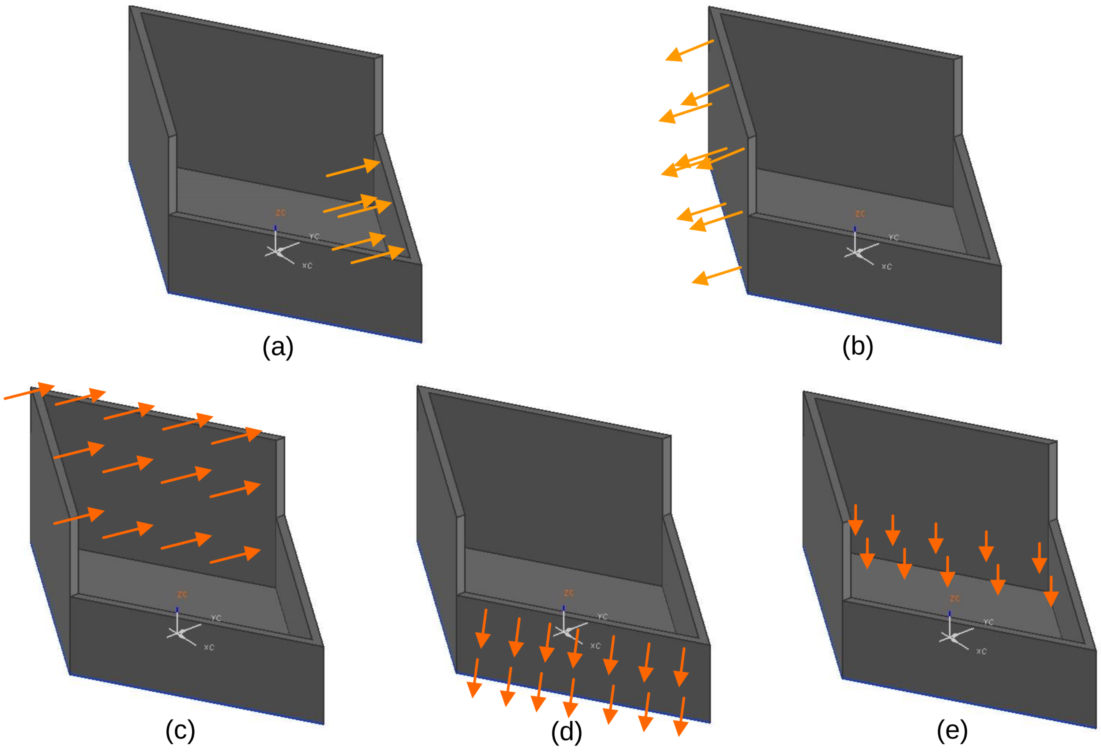


Figure 5.23: Measuring the stiffness of the assembly: (a) uniform force normal to S_1, (b) uniform force normal to S_3, (c) uniform force normal to S_2, (d) uniform force normal to S_4, and (e) uniform force normal to bottom surface.

A penalty is applied if the minimum displacement of all nodes with snaps under local heating at T_{l1} = 120^{o}C and T_{l2} = –80^{o}C is less than h_{+} = 1 mm or more than h_{++} = 1.5 mm. Another penalty is applied if the maximum displacement of all nodes with snaps under uniform heating T_u = 50 ^{o}C is more than h_{-} = 1 mm. The displacement constraints are written as penalty functions as given below:

$$g_1\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l\right)=\max\left(0,h_{lower}-\text{min\_disp\_loc}\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l\right)\right) \tag{5.16}$$

$$g_2\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l\right)=\max\left(0,\text{max\_disp\_loc}\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l\right)-h_{upper}\right) \tag{5.17}$$

$$g_3\left(\boldsymbol{x},\boldsymbol{z},T_u\right)=\max\left(0,\text{max\_disp\_unif }\left(\boldsymbol{x},\boldsymbol{z},T_u\right)-h_{-}\right) \tag{5.18}$$

Similar to the previous case study, since the enclosure geometry consists of a series of rectangular surfaces, the finite element mesh is made uniform, Figure 5.24, for simplicity and accuracy. The heating area, therefore, can be defined using discrete coordinate variables instead of using real variables. The optimization problem is, thus, solved using only discrete GA. The parameters used in GA for this case study are shown in Table 5.4. A summary of the snap bounds and the heating temperature values is given in Table 5.5. The optimization problem is solved using NSGA-II running on a Windows workstation with a 3.4 GHz dual core Intel Pentium 4 processor. The time to finish the whole GA run is 4 days.

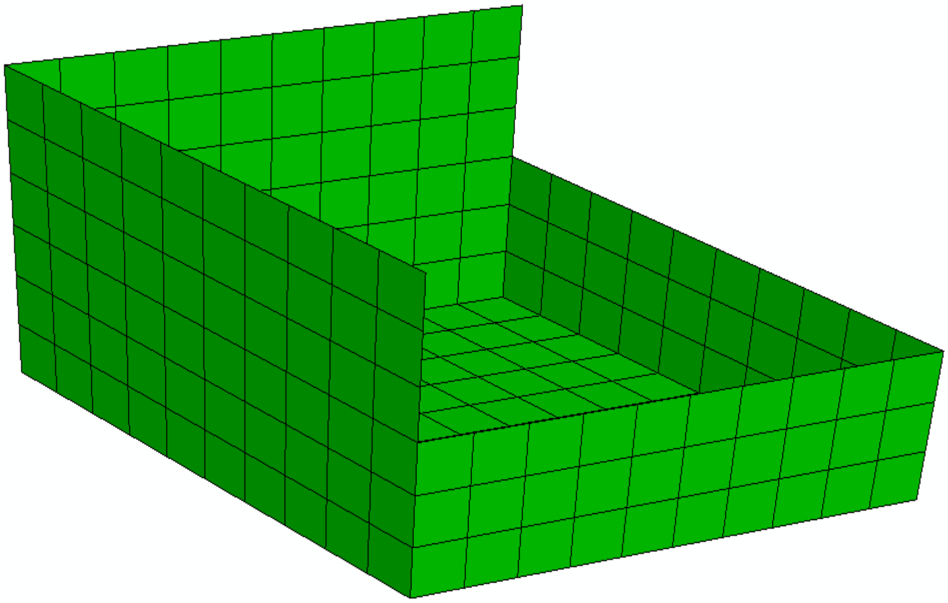


Figure 5.24: FEM mesh for the bottom enclosure part.

Table 5.4: GA parameters used in the case study

Parameter	Value
Population size	200
Number of generations	200
Crossover probability	0.95
Mutation probability	0.05

Table 5.5: Summary of the snap bounds and the heating temperature values.

Parameter	Value
h_{lower}	1.0 mm
h_{upper}	1.5 mm
h_{-}	1.0 mm
T_{l1}	120 ^{o}C
T_{l2}	–80 ^{o}C
T_{u}	50 ^{o}C

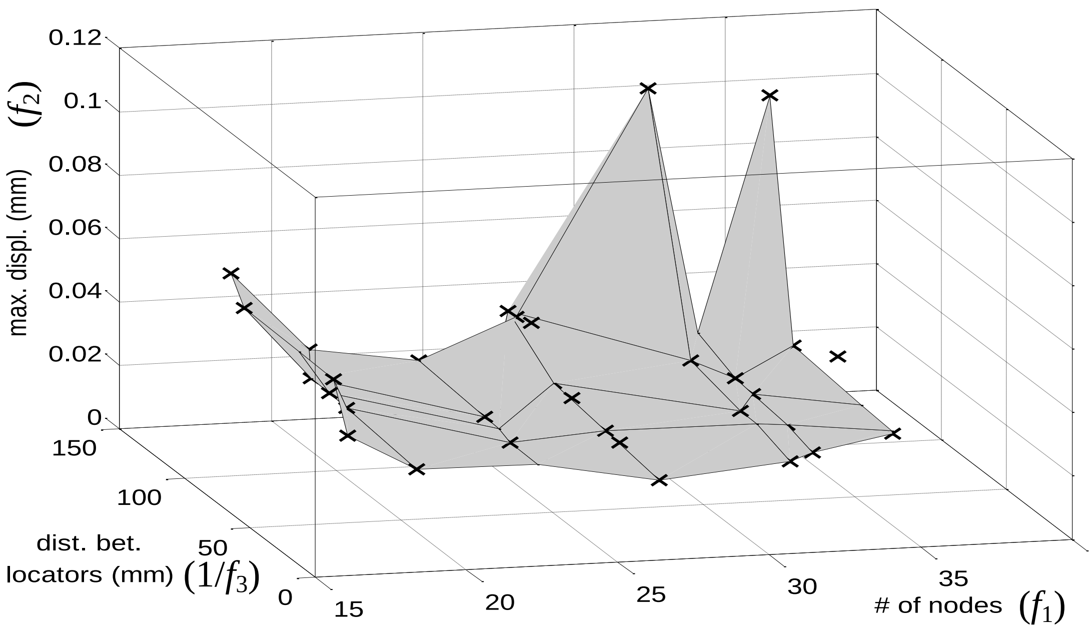


Figure 5.25: 3D Pareto front for the case study.

Figure 5.25 Shows the Pareto Optimal solutions in 3D, which exhibit a trade-off between the part compliance, the amount of heating required (number of heated nodes) and the tolerance stack-up. Figure 5.26 shows the projection of the Pareto front in the heating area size (f_1) verses the distance between locators constraining same DOFs ($1/f_3$) coordinates; whereas, Figure 5.27 shows the projection of the Pareto front in the distance between locators constraining same DOFs ($1/f_3$) verses the displacement at the mating line (f_2) coordinates. Finally, Figure 5.28 shows the projection of the Pareto front in the displacement at the mating line (f_2) verses the heating area size (f_1) coordinates. All the results exhibit multiple heating locations and double acting snaps as tools for the lock-and-key method. Double-latching snaps are more effective lock as they ensure that with excessive heating, the snap will become locked again.

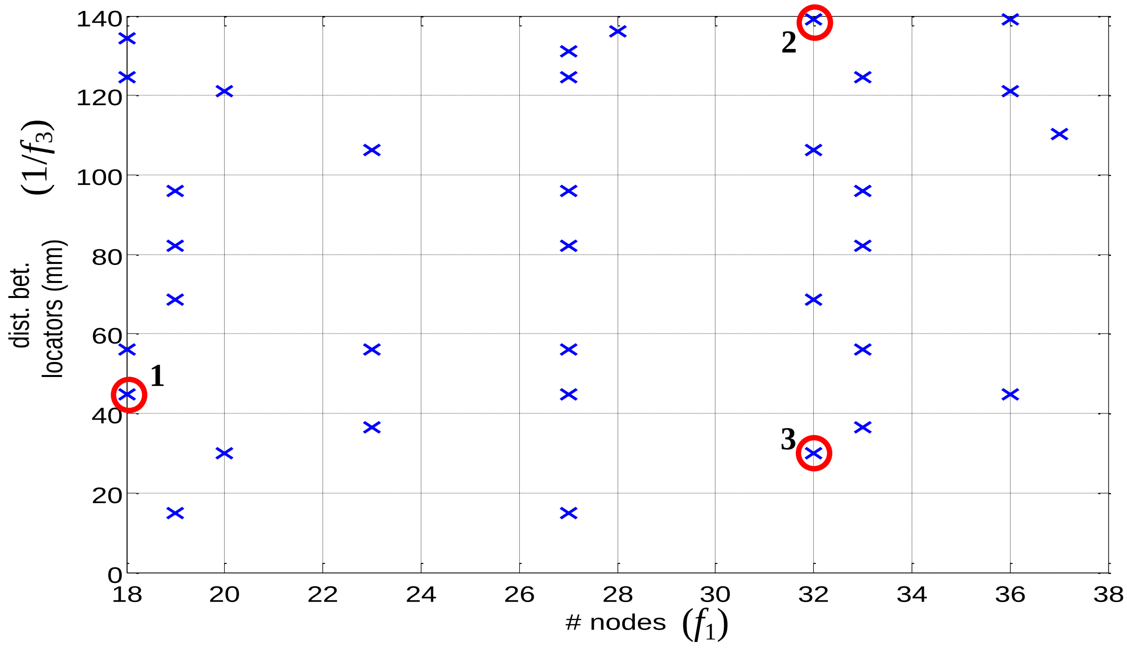


Figure 5.26: 2D Pareto front in the heating area size verses the distance between locators constraining same DOFs coordinates.

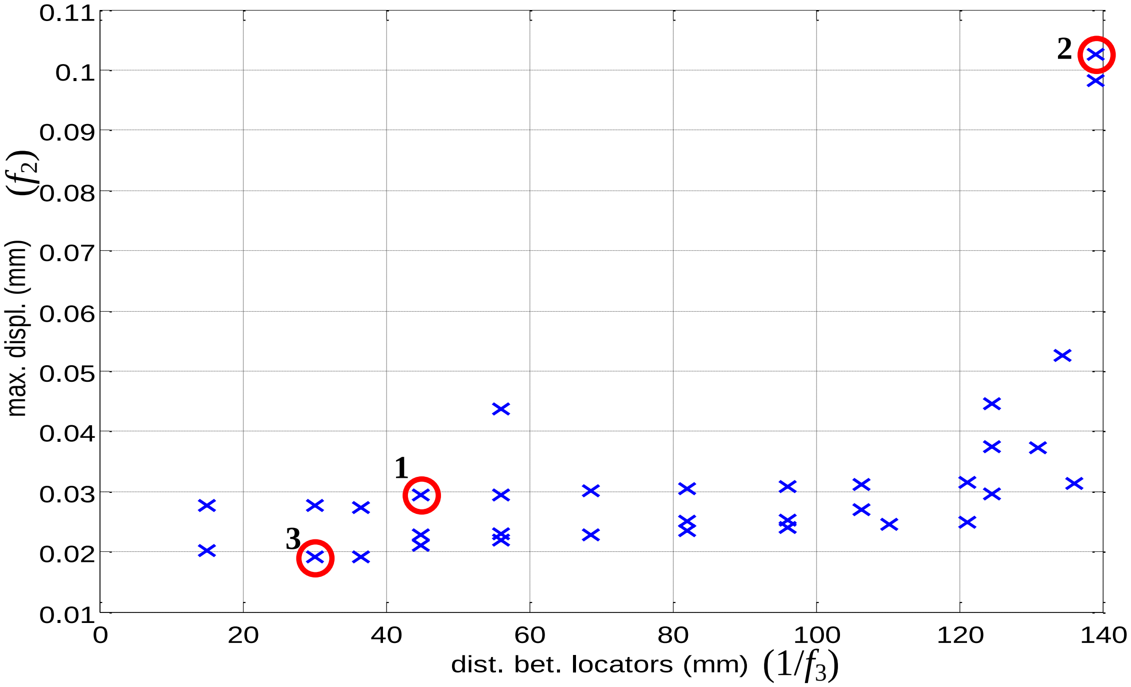


Figure 5.27: 2D Pareto front in the distance between locators constraining same DOFs verses the displacement at the mating line coordinates.

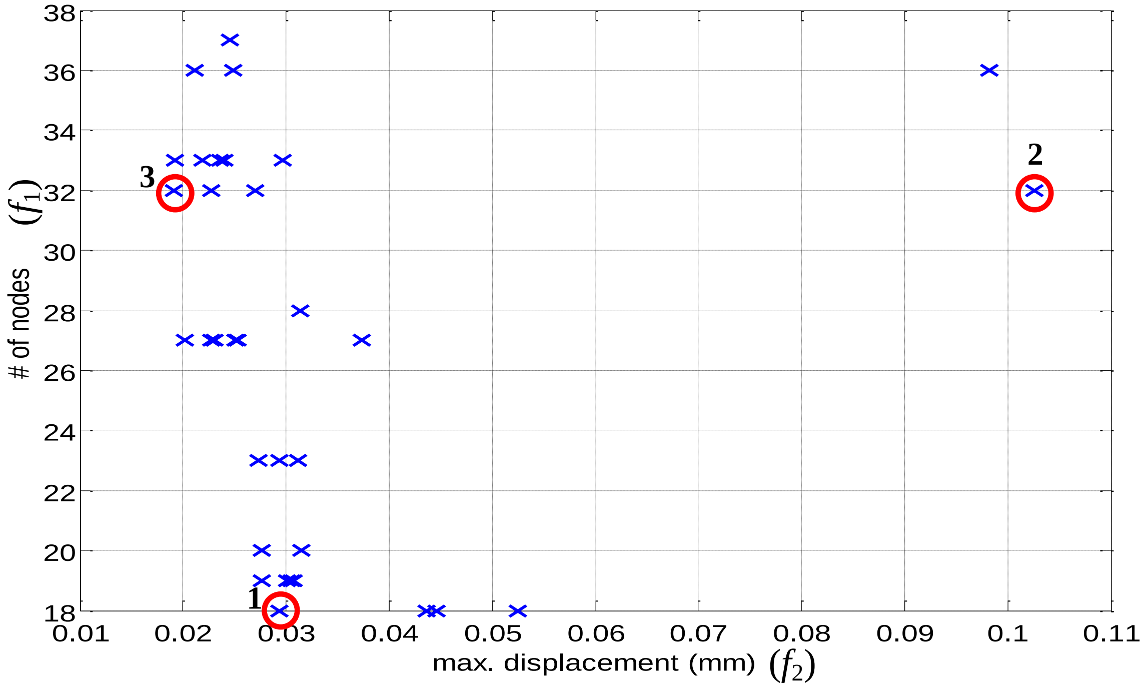


Figure 5.28: 2D Pareto front in the displacement at the mating line verses heating area size coordinates.

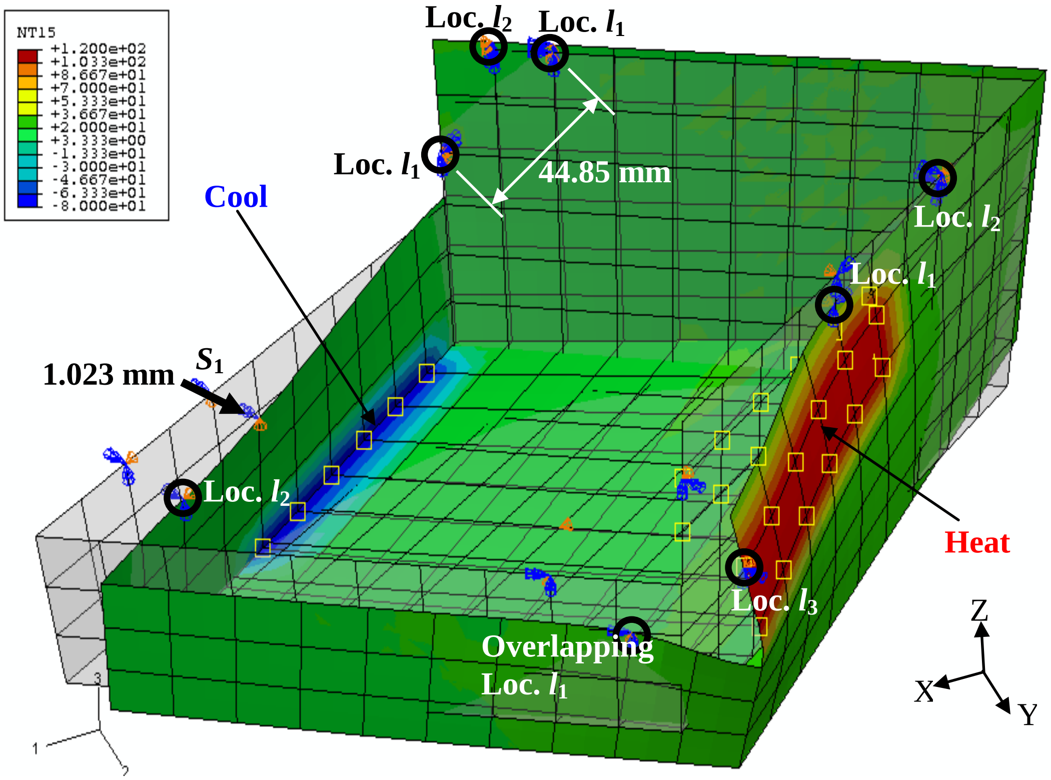


Figure 5.29: Optimum solution with minimum heat area (solution 1)

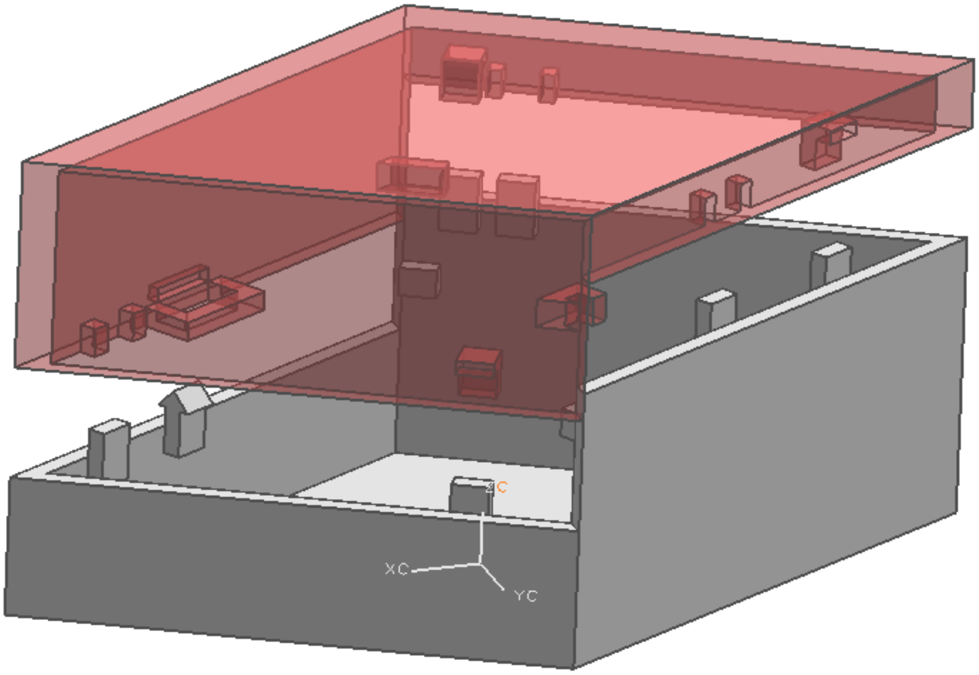


Figure 5.30: CAD drawing for the optimized rhombus enclosure model for optimum solution 1.

Figure 5.29 shows the optimal placement of locators and snaps and the response due to local heating of the solution with minimum heating and cooling areas (solution 1). The total numbers of heated and cooled are 12 nodes and 6 nodes respectively. The maximum deformation at the mating line under the assembly's own weight is due to pressure loading on the bottom surface and is equal to (0.029 mm). The minimum distance between locators constraining same DOFs is 44.85 mm. Locator positions are marked with black circle, while the snap position is marked with an arrow showing the bulging direction. The locator types are written on the Figure. Figure 5.30 shows a CAD drawing for the rhombus enclosure model for optimum solution 1. The optimization resulted in one pair of overlapping locators (two locators at the same location) and of the same type. Those locators are, therefore, treated as one locator. The bulging at the snap location is inward and is equal to 1.023 mm. The snap is placed on the shown part at the optimized snap locations; while the catch is placed on the other part (not shown). The deformation at the snap location due to uniform heating at 50°C is 0.341 mm.

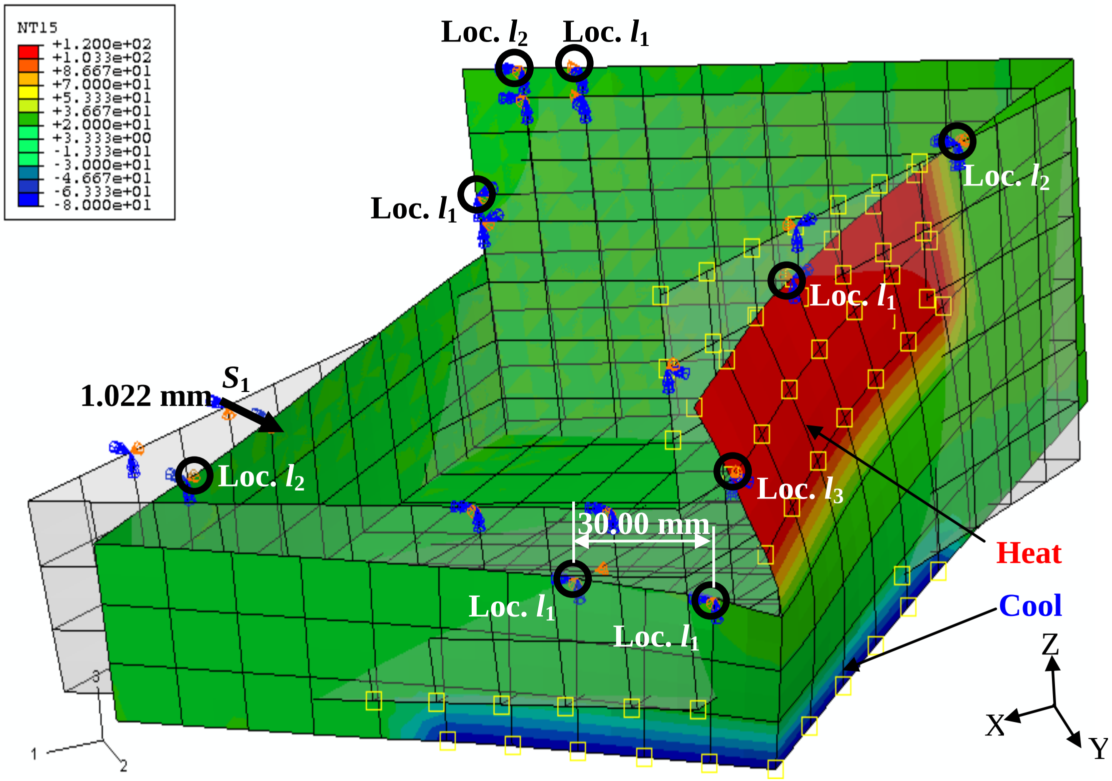


Figure 5.31: Optimum solution with minimum compliance (solution 2).

Figure 5.31 shows the optimal placement of locators and snaps and the response due to local heating of the solution with minimum deformation at the mating line due to the assembly's own weight (minimum compliance) (solution 2). The total numbers of heated and cooled areas are 21 nodes and 11 nodes respectively. The maximum deformation at the mating line under the assembly's own weight is due to pressure loading on the bottom surface and is equal to (0.019 mm). The minimum distance between locators constraining same DOFs is 30.00 mm. Locator positions are marked with black circle, while the snap position is marked with an arrow showing the bulging direction. The locator types are written on the Figure. Figure 5.32 shows a CAD drawing for the rhombus enclosure model for optimum solution 2. The bulging at the snap location is inward and is equal to 1.022 mm. The snap is placed on the shown part at the optimized snap locations; while the catch is placed on the other part (not shown). The deformation at the snap location due to uniform heating at 50^{o}C is 0.233 mm.

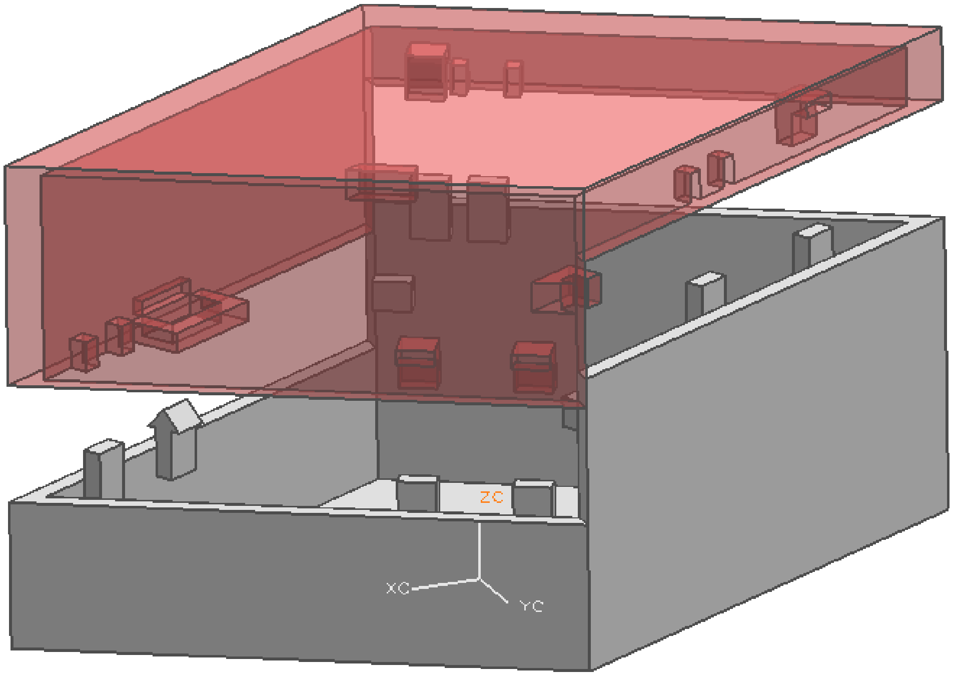


Figure 5.32: CAD drawing for the optimized rhombus enclosure model for optimum solution 2.

Figure 5.33 shows the optimal placement of locators and snaps and the response due to local heating of the solution with maximum distance between locators constraining same DOFs (minimum susceptibility to tolerance stack-up) (solution 2). The total numbers of heated and cooled areas are 21 nodes and 11 nodes respectively. The maximum deformation at the mating line under the assembly's own weight is due to pressure loading on the bottom surface and is equal to (0.103 mm). The minimum distance between locators constraining same DOFs is 139.05 mm. Locator positions are marked with black circle, while the snap position is marked with an arrow showing the bulging direction. The locator types are written on the Figure. Figure 5.34 shows a CAD drawing for the rhombus enclosure model for optimum solution 3. The optimization resulted in two pairs of overlapping locators (two locators at the same location) and of the same type. Each pair of these locators is treated as one locator. The bulging at the snap location is inward and is equal to 1.001 mm. The snap is placed on the shown part at the optimized snap locations; while the catch is placed on the other part (not shown). The deformation at the snap location due to uniform heating at 50°C is 0.543 mm.

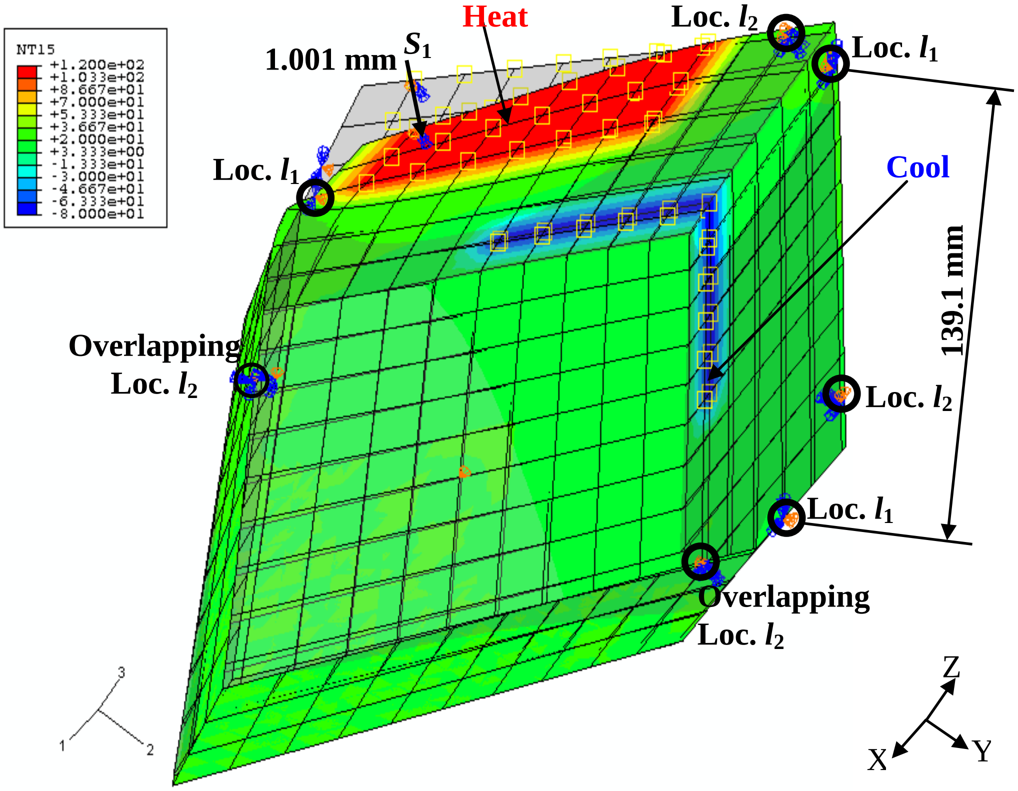


Figure 5.33: Optimum solution with maximum distance between locators constraining same DOFs (solution 3).

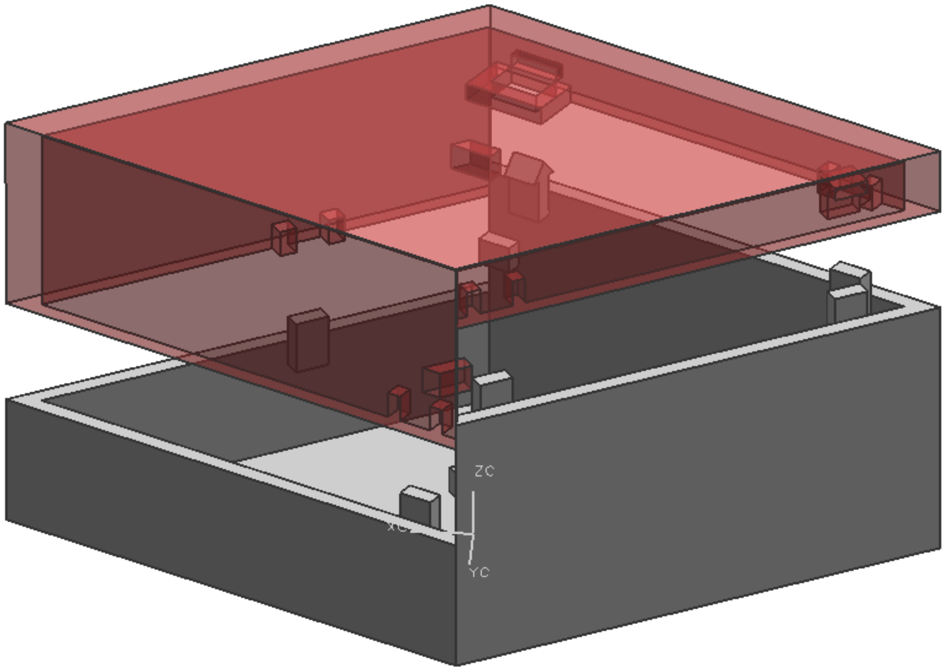


Figure 5.34: CAD drawing for the optimized rhombus enclosure model for optimum solution 3.

The use of two heating locations (one for heating and one for cooling) in this case study ensures the complexity of the lock-and-key concept. If one area (either heating or cooling) is activated, the snap will remain locked. Both areas have to be activated at the same time to the desired temperatures to ensure snap unlocking. The use of double acting snap ensures that in case of excessive heating, the snap will lock again.

5.3.3 Conclusion

This case study proposed the use of the unified method developed in Chapter 3 on an enclosure with complex mating line geometry to demonstrate its effectiveness. The resulting Pareto-optimal solutions exhibit alternative design with different trade-offs between structural stiffness during snap engagement, heating area necessary for snap disengagement, and minimum effect of tolerance stack-up due to locators fixing the same degree of freedom more than once. The use of two heating regions at different temperatures and the use of double-latching snaps ensure the complexity of the snap lock. The right procedure (right key) has to be followed or snaps will remain locked. The difference between h_{upper} and h_{lower} is only 0.5 mm. Any minor variation (noise) in the heating temperature values or heating area locations can cause the results to violate the desired requirements and fail to disengage. Robust optimization can be used in this case to make sure that the resulting designs are less prone to noise factors (e.g. environmental variations during the product’s usage, manufacturing variations, and component deterioration.

Chapter 6: Parallel Genetic Algorithms

6.1 Introduction

Genetic Algorithm (GA) was first introduced by John Holland in 1975 [101]. GA is a heuristic optimization method that mimics the evolution of living creatures by implementing the idea of the survival of the fittest. Different Multi Objective Genetic Algorithm (MOGA) versions were introduced in the last decades to solve problems with multiple competing objectives. Solving such problems with evolutionary algorithms, like MOGA, has an advantage over other optimization methods. Due to the population-based nature of MOGA, it can capture several points on the Pareto optimal set in just one run. Generally speaking MOGA is a favored optimization method over derivative based methods due to its ability to deal with large numbers of continuous and discrete variables simultaneously; it is also capable of dealing with highly multi modal and discontinuous objective functions to find near global optimal solutions. On the other hand, as with all heuristic based methods, GA is highly time-consuming. In some applications, as in electric appliances, the design process is very short. Designers cannot wait until an optimum solution is reached. Parallelization of optimization algorithms can help reduce the time needed to reach an optimum solution; thus optimization can be included in the design phase.

Although a number of researchers implemented different versions of parallel GA, load balancing — dividing the computational work on all processors so that the idle time of any processor is minimized — did not receive much attention. The reason behind this is probably because load balancing may not be needed when the time to evaluate the objective functions does not vary with respect to the design variables. In this research, the objective function evaluation time can vary significantly. You can notice from the previous case studies that the objective functions evaluations depend on running multiple Finite Element Analyses (FEA) that easily vary in time as they depend on iterations to converge to the solution. Actually, the time to evaluate the objective functions can vary from milliseconds, in case of an infeasible design where the FEA is not even initialized, to few minutes in some feasible cases where the convergence of the FEA requires multiple iterations. As a result, this chapter focuses on presenting an active load balancing scheme to a parallel implementation of the Multi objective Genetic Algorithm used in this research (NSGA-II) to improve its efficiency.

One method to parallelize MOGA is to dedicate a processor for each function that needs evaluation, while there is a manager processor that does the GA overhead work (selection, mutation and cross-over to generate the new population design variables (chromosomes)) and evaluates the fitness function. Here, since objective function evaluations may vary significantly in time, there may be high load imbalance. In addition, the communication time (number of sends and receives between the manager and workers processors) can be of very high significance compared to the GA overhead work and the objective function evaluation time. Another method is to dedicate each processor to solve a number of fitness functions, evaluate all objectives and constraints for a single set of variables, then send these values to the manager to do the GA overhead work. Such a model can be seen as Manager-worker model, reviewed in Section 2.4. The parallelization of GA in this way can be seen as using a serial GA code but in less time, which is the focus of our research. If the time to evaluate the fitness functions can vary based on the design variables, e.g. if it involves time varying finite element simulations, load imbalance may occur. A third method is to use distributed manager-worker model (island model), reviewed in Section 2.4, each processor works on its own share of the population. Migration between processors occurs after a number of generations. Cantù-Paz [132] has further details on different parallel paradigms; though, they are analyzed in a serial sense.

Another major cause of load imbalance is the use of variable processor speeds of the parallel computing grids. Currently, parallel computing grids continue to grow with newer processors while keeping older processors running. This results in grids with processors with varying speeds. Jobs submitted to parallel computing grids can be assigned to processors with different speeds leading to continuous load imbalance. As a result, it is becoming important to always include load balancing to codes that will run in parallel.

6.2 Parallelization of the GA algorithm

The proposed parallel code assumes distributed memory parallel computing grid and uses Message Passing Interface (MPI) for communication between processors. The code uses the manager-worker model. The overview of the proposed parallelization method can be summarized as follows:

- The manager processor does the GA overhead work.
- The manager divides the population over the other processors to evaluate the objective function values and constraints; then, the workers return the objective function values back to the manager; the manager may/may not take a share of the population to be evaluated.
- During the process of generating the new population, if a new chromosome is present in the old population, this chromosome need not be evaluated; its function values are passed to the new population. As a result, the number of chromosomes that need to be evaluated vary from one generation to the other; yet, it cannot be greater than the population size.
- Worker processors are initialized to receive the whole population and find the exact number received using the MPI_Getcount() function.

Although, it may seem obvious to divide each generation evenly among multiple processors, a problem of load imbalance may occur when the objective function evaluation depends on time varying simulations, e.g. crash simulations or contact mechanics, or if a simple constraint check is done to determine solution feasibility prior to evaluating the time-consuming objectives, or even if there are some variations in processors' performance.

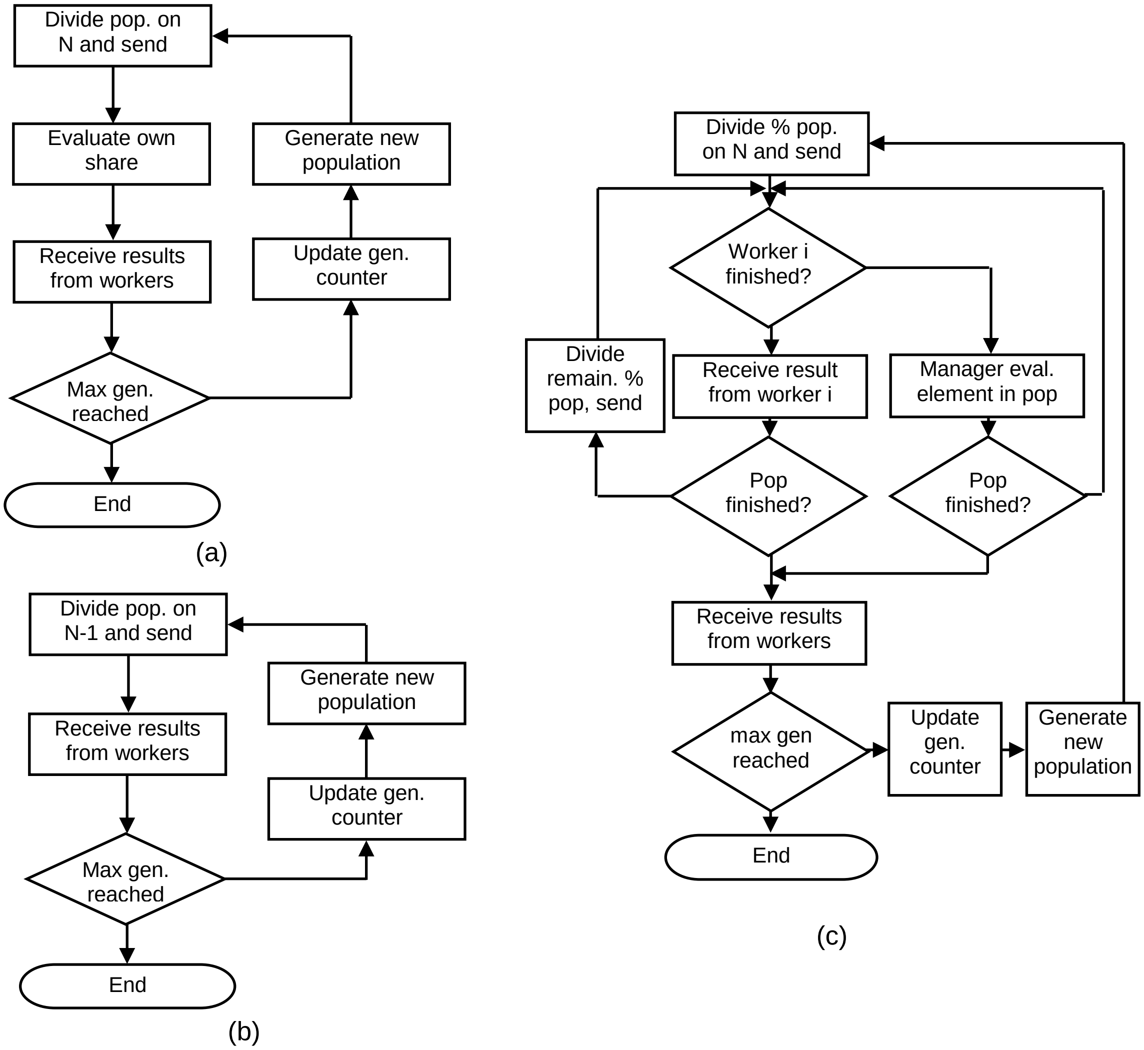


Figure 6.1: Flow charts for the developed parallel schemes (a) Scheme 1 (manager works), (b) Scheme 2 (manager doesn't do work), and (c) Scheme 3 (active load balancing)

Three different load balancing schemes are proposed (Figure 6.1). 1) Unevaluated chromosomes are divided evenly over all the processors including the manager. This scheme is helpful if the manager is not overworked. 2) Unevaluated chromosomes are divided evenly over the worker processors only, leaving the manager to do only the managerial GA work. This scheme is helpful if the manager is overworked to the extent that the performance of the first scheme is reduced. It should be noted that this scheme works only for a total number of processors > 2, if the number of processors =2, the scheme will use the same procedure as scheme 1. If the objective function evaluation time is inconsistent, schemes 1 and 2 may result in much idle time in some processors as they may finish their shares earlier than other processors. 3) Only a certain percentage of the population is evenly divided over the given processors. After sending the chromosomes to all worker processors, the manager tests if any processor has sent back the resulting objective function values, if so, the manager sends this processor a part of the remaining population, otherwise, the manager starts evaluating the remaining population until any processor finishes. This process is repeated until the whole population is evaluated. This scheme is helpful for problems with highly inconsistent objective function evaluation times.

The time to evaluate the objective functions is calculated each time the objective function is called. The mean and variance are evaluated after each population. Depending on the mean and variance, the code will use one of the three load-balancing schemes described earlier. To determine when to use each load balancing scheme, the experiments presented in Section 6.3 were done to see the effect of the mean objective functions evaluation times and the standard deviation on the choice of the scheme used.

6.3 Analysis of the parallel Genetic Algorithm

All the parallel codes were written in C++ using Message Passing Interface (MPI). First, a test was done to evaluate the send and receive times taken to transfer data between different processors, Section 6.3.1. This test is needed to compare the communication time to the objective function evaluation time and see how the communication time can affect the performance of the parallel code. In the second experiment, Section 6.3.2, we implement the three schemes on a test optimization function and compare the performance of the 3 schemes. In the last experiment, Section 6.3.3, we study the effect of the mean and standard deviation of the objective function evaluation times on the percentage of the population the manager processor should send to the worker processors, X%. It should be noted that the results of these experiments are specific to the parallel computing cluster used. The codes were executed on the parallel computing cluster, *nyx*, at the Center of Advanced Computing (CAC) at the University of Michigan. The highlights of *nyx* are given in Table 6.1.

Table 6.1: nyx cluster highlights

Criteria	**Value**
Type	Distributed Memory cluster
Number of nodes	586
Types of nodes	single CPU/node and dual CPU/node
Processor Types	AMD Opteron (single core and dual core)
Processor Speeds	Opt240 (1.8 GHz) – Opt2220 (2.8 GHz)
Number of Cores	1762
Memory per Node	2 GB – 10 GB
Storage Space	80 GB/user
Node and core selection	Based on availability

6.3.1 Test the send/receive times

A simple "*ping-pong*" code was written to send an array of variable type "*double*", with an increasing size from 1 to 5000, from one processor to another. The send and receive times were recorded and the best fit straight line that represents these data is obtained using linear regression. The data had very high variance, but in general the expected increasing trend could be captured. Figure 6.2a, and Figure 6.2b show the send and receive plots for the whole range, while Figure 6.3a, and Figure 6.3b show the send and receive plots for array sizes till 500 and the least squares fitting line.

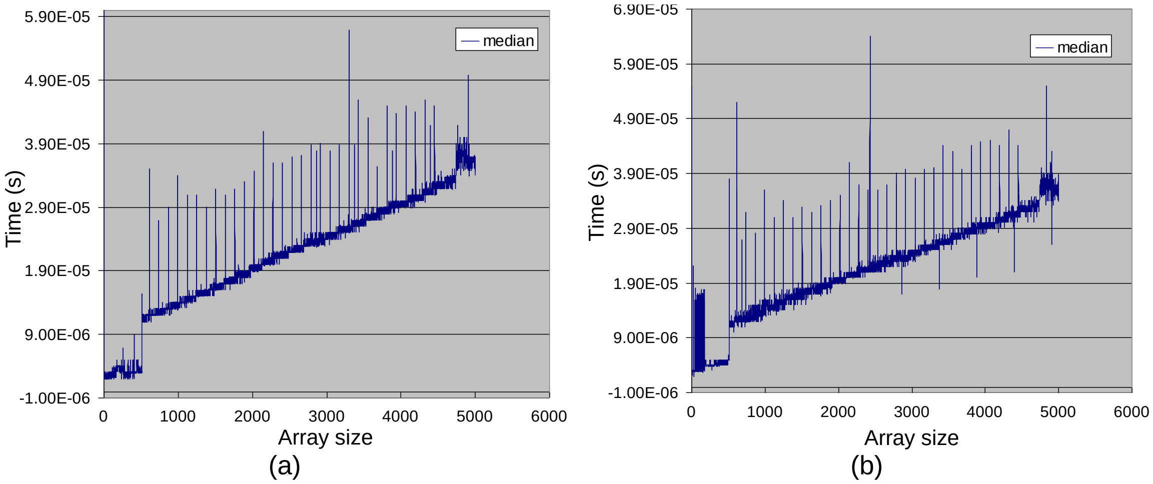
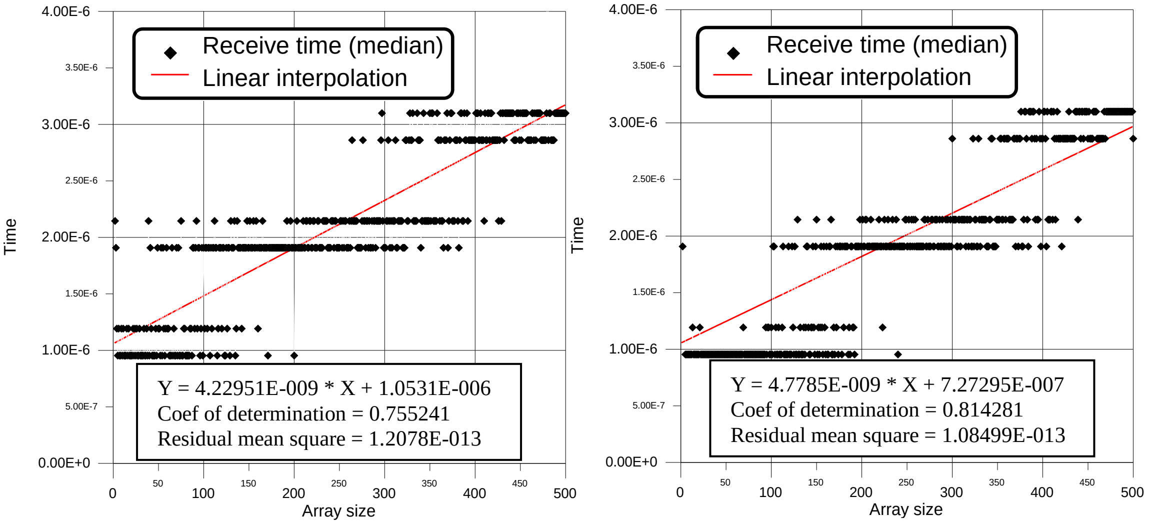


Figure 6.2: Receive (a) and send (b) plots for an array of size 1 to 5000

The results in Figure 6.2 and Figure 6.3 are realistic to what is mentioned in the literature [132]. Assuming the linear interpolation functions for the send and receive times to be $(\alpha + \beta \times \textit{data size})$; the value of α is about 1000 times that of β. Since the average number of data sent/received during a single communication in the GA code is relatively small, one can ignore $\beta \times$ (*data size*) use α as an estimate to the average time to send/receive a packet. $\alpha_{av} = \alpha_{send} + \alpha_{recv} = 0.888\text{e-}6$ seconds. It is important to know the average time taken for sending and receiving a packet as this is a major overhead affecting the performance of a parallel code. In the case of parallel GA if the objective function evaluation time is very short and is comparable to α_{av}, using a parallel version of GA may not be of significant advantage over its serial version.



(a) (b)

Figure 6.3: Receive (a) and send (b) plots for an array of size 1 to 500 and the least squares linear fit for each data

6.3.2 Testing the 3 schemes on fixed time functions

In this experiment the three parallelization schemes were implemented on a test optimization function to compare their performance. The experiment was repeated four times with different objective function evaluation times to study the effect of different objective function evaluation times on the performance of the three schemes.

During this process, the mathematical objective function given in Equation (6.1) is used as a test function:

$$f = e^{-0.05x}\cos(x) + e^{-0.05y}\cos(y). \tag{6.1}$$

The following pseudo code is used to evaluate the objective function. Each time the objective function in Equation (6.1) is called, it is evaluated for a number of times to simulate different objective function evaluation times:

```
- Max loop index ← int( random number( $loop_{mean}$, $loop_{sd}$) )
- index ← 0
While (index < Max loop index)
    - evaluate $f$
end
```

loop$_{mean}$, *loop*$_{sd}$ are the mean and standard deviation of the time to evaluate the objective function to simulate variable objective function evaluation times. The random number generator seed is fixed to minimize inconstancy in the GA operations during different runs.

In this experiment, the objective function is evaluated for (100, 1000, 10000 and 100000) times with a standard deviation value of zero. All runs were done with the same population size and number of generations. The runs were done using 1, 2, …, 10 processors on *nyx*. Table 6.2 shows the results (time to finish a GA run with population size 50 and a number of generations of 40). The time row in Table 6.2 shows the average time to evaluate a single objective function for 100, 1000, 10 000 and 100 000 times. The efficiency of parallel code is evaluated using Equation (6.2).

$$efficiency = \frac{\text{time to finish GA run using 1 processor}}{n \times \text{time to finish GA run using } n \text{ processors}} \qquad (6.2)$$

Table 6.2: Different schemes' performance with the change in number of processors and objective functions evaluation time

	Scheme 1 (manager does work)				Scheme 2 (manager does no work)				Scheme 3 (load balancing)			
Loop	100	1000	10000	100000	100	1000	10000	100000	100	1000	10000	100000
Time	2.61e-5	2.56e-4	2.54e-3	2.54e-2	2.61e-5	2.56e-4	2.54e-3	2.54e-2	2.61e-5	2.56e-4	2.54e-3	2.54e-2
1	0.1207	0.3238	2.3603	22.6774	0.1216	0.3316	2.3578	22.8102	0.1199	0.3249	2.3586	22.7101
2	0.1649	0.2727	1.3082	11.6444	0.1679	0.2706	1.2771	11.6191	0.1763	0.2845	1.3314	12.1044
3	0.1631	0.2358	0.9312	7.8982	0.1655	0.2781	1.3176	11.6191	0.1761	0.2580	1.0072	8.1608
4	0.1654	0.2254	0.7996	6.4730	0.1672	0.2382	0.9330	7.8897	0.1750	0.2402	0.8226	6.4771
5	0.1738	0.2186	0.6998	5.5279	0.1658	0.2209	0.7781	6.3459	0.1835	0.2292	0.6884	5.2079
6	0.1924	0.2149	0.6723	5.2662	0.1633	0.2223	0.7279	5.8072	0.1877	0.2228	0.5954	4.6693
7	0.2554	0.2081	0.6021	4.5269	0.2554	0.2111	0.6359	4.8703	0.2091	0.2202	0.5456	3.9036
8	0.1642	0.2048	0.5826	4.3467	0.1618	0.2046	0.5629	4.1651	0.1985	0.2210	0.5166	3.4811
9	0.1812	0.2075	0.6159	4.6954	0.1784	0.1865	0.4908	3.5365	0.2073	0.2303	0.4881	3.0848
10	0.1785	0.2038	0.5398	3.9538	0.1698	0.2017	0.5954	4.5713	0.1953	0.2393	0.4800	2.9912
E	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
F	0.3661	0.5937	0.9021	0.9737	0.3621	0.6127	0.9232	0.9816	0.3401	0.5709	0.8858	0.9381
F	0.2466	0.4578	0.8449	0.9571	0.2448	0.3975	0.5965	0.6544	0.2270	0.4197	0.7806	0.9276
I	0.1824	0.3591	0.7380	0.8758	0.1818	0.3480	0.6318	0.7228	0.1713	0.3382	0.7168	0.8766
C	0.1389	0.2963	0.6746	0.8205	0.1466	0.3002	0.6061	0.7189	0.1307	0.2836	0.6853	0.8721
I	0.1046	0.2512	0.5851	0.7177	0.1241	0.2486	0.5398	0.6546	0.1065	0.2431	0.6602	0.8106
E	0.0675	0.2223	0.5600	0.7156	0.0680	0.2244	0.5297	0.6691	0.0819	0.2108	0.6176	0.8311
N	0.0919	0.1977	0.5064	0.6521	0.0939	0.2026	0.5236	0.6846	0.0755	0.1837	0.5707	0.8155
C	0.0740	0.1734	0.4258	0.5366	0.0757	0.1975	0.5338	0.7167	0.0643	0.1568	0.5369	0.8180
Y	0.0582	0.1589	0.4372	0.5736	0.0652	0.1644	0.3960	0.4990	0.0614	0.1358	0.4914	0.7592

It can be deduced from Table 6.2 that when the objective function evaluation time is short and with a small population size, *i.e.* when the average time to evaluate the whole population is < 0.01 seconds, parallelization is not favored. The serial overheads and the communication time (send and receive time) are of high significance that worsens the efficiency. With relatively brief objective function evaluation time and relatively large number of processors, scheme 2 showed better efficiency than scheme 1; yet, this occurs with very low efficiency due to the large overhead from the serial part. In other words, although scheme 2 has slightly better efficiency than scheme 1, the efficiency gained by scaling remains very low. Finally, as the time to evaluate the objective function increases, (> 2.0e-3 seconds) even with almost zero variance, the load balancing scheme (scheme 3) performs better. The efficiency of the load balanced scheme is much more consistent, while the efficiency of scheme 1 deteriorated to 71% with 6 processors, the efficiency of scheme 3 was still higher and only reached 76% with 10 processors. This can be because the number of chromosomes is not divisible by the number of processors, so one of the processors gets overloaded; in addition, the manager processor may be overloaded. It is important as well to note that with the varying speeds of the CPUs on the cluster, a job may run on two processors with different speeds. This can easily result in lower performance for schemes 1 and 2 than scheme 3.

6.3.3 Load balancing using scheme 3

In the load balancing scheme, scheme 3 proposed in this chapter, only a percentage (X %) of the population is sent to the worker processors. The manager processor starts evaluating the remainder of the population until any of the workers finishes its job. At this time, the manager sends another percentage of the remaining population to that worker. The manager keeps doing this procedure until the whole population is evaluated. The following pseudo code describes how the load balancing scheme works:

```
- Processor share = ceiling(number of chromosomes × X / number of processors)
- Send processor share to all worker processors
- index ← Processor share × number of processors
While (index < number of chromosomes)
    - If a processor finishes
            - evaluate the processor share from the remaining chromosomes
            - send the new processor share to the processor that finished
            - index ← index + processor share
    - else
            - Manager Processor evaluates a chromosome,
            - index++
    end
 end
```

In the above procedure, it is important to find a reasonable value for X that minimizes the time needed to finish the GA population. As a result, this experiment was done to study the effect of the mean and/or variance of the time to evaluate the objective functions on the value of X. Table 6.3 shows the times to finish a whole GA run with ($loop_{mean}$ =

100000) and standard deviation factors (*sdf* = 0.3, 0.1, 0.05, and 0.01). The value of the loop's standard deviation is therefore ($loop_{sd} = sdf \times loop_{mean}$). The experiment was done using scheme 3 with values of (X = 0.5, 0.6, 0.7, 0.8) and scheme 1, and number of processors equal to 4, 6 and 8.

Table 6.3: Different schemes' computation times with the change in standard deviation and the value of X.

# processor	X	Standard dev. factor of the mean value of loop iteration			
		0.01	0.05	0.1	0.3
4	Scheme 1	7.047	6.517	6.601	7.178
	Scheme 3, x=0.5	9.226	6.885	8.773	7.010
	Scheme 3, x=0.6	7.528	7.520	6.817	6.978
	Scheme 3, x=0.7	6.663	6.711	7.152	7.287
	Scheme 3, x=0.8	*6.494*	*6.667*	*6.598*	*6.578*
6	Scheme 1	5.171	4.9147	5.189	5.348
	Scheme 3, x=0.5	5.029	5.048	5.109	5.197
	Scheme 3, x=0.6	4.935	4.939	4.993	5.266
	Scheme 3, x=0.7	4.928	4.899	4.895	*4.928*
	Scheme 3, x=0.8	*4.583*	*4.707*	*4.629*	4.946
8	Scheme 1	5.047	5.066	4.782	4.837
	Scheme 3, x=0.5	3.884	4.058	4.038	4.169
	Scheme 3, x=0.6	3.880	4.026	3.963	2.288
	Scheme 3, x=0.7	*3.497*	*3.701*	4.037	4.027
	Scheme 3, x=0.8	3.674	3.756	*3.773*	*3.943*

In the above table, the shaded cells show the minimum time to finish the whole GA run for a given number of processors and standard deviation factor. It can be deduced from the above table that, although the variance in the objective function evaluation time affects the time to solve the problem, as the variance increase, the total time to finish the GA increases too; on the other hand, there is no clear correlation between the value of X and the time variance. What can be deduced is that an X value around 0.7 – 0.8 does fine for the various variance values.

6.3.4 Active load balancing

From the discussion in the two previous subsections, it can be figured out that schemes 1 and 3 have better performance over scheme 2. As a result, the active load balancing scheme, proposed in this chapter, selects only from schemes 1 and 3 as shown in the pseudo code below.

```
While (index < number of populations)
        - evaluate the average objective function evaluation time ($f_{eval\_av}$)
        If ($f_{eval\_av}$ < Threshold)
            - use scheme 1 (divide population evenly over all the processors
        - else
            - use scheme 3 (Load balancing with X = 0.8)
        end
        - index++
end
```

It can be deduced, as well, that load balancing scheme 3 can perform better than scheme 1 even for consistent objective function times, as long as the time to evaluate the objective function is long enough. The reason behind that is the fact that a single GA job can run on different processors with different speeds. As a result, although the processors are evaluating the same objective function, the variation in the processor speeds affects the time taken to evaluate objective function; thus, load balancing using scheme 3 can lead to better performance. The above results, based on *nyx* performance, show that if the average time to evaluate the objective function is less than (*Threshold* = 2.5e-3 seconds), it is better to use scheme 1; otherwise, use scheme 3. For scheme 3, the percentage of the total number of chromosomes that need to be sent for evaluation by other processors each time, X, is set as 80%. It should be noted that these values are specific for *nyx*. If the proposed code is to run on a different parallel computing grid, the code can still perform well; yet, for optimum performance, the above experiments need to be repeated to adjust the threshold and X values. It should also be noted that even for *nyx*, if the grid structure changes, by adding newer processors and/or removing old processors, these parameter values may need to change as well.

6.4 Test the algorithm

The proposed active load balancing algorithm was finally applied on the same test function where each time the function is called, the function is evaluated for a random number of times that is randomly selected from a uniform distribution between (100 000, 500 000). The GA elapsed time, speed up and efficiency are recorded in Table 6.4 and Figure 6.4. In addition a plot for the elapsed time and the relative speed up is shown. The results showed almost a linear speed up except for 16 processors. This is because as the number of processors increases, the load on each worker processor decreases and the serial overhead load becomes more and more significant. In addition, due to the large variance in the objective function evaluation times, some processors may still be overworked leading to slight load imbalance between processors.

Table 6.4: Proposed GA elapsed time, speed up and efficiency with the change in the number of processors

# of processors	Elapsed time (s)	Relative speed up	Efficiency
1	69.310	1	1
2	34.906	1.986	0.993
4	19.113	3.626	0.907
6	13.783	5.029	0.838
8	10.746	6.450	0.806
12	7.709	8.990	0.749
16	6.638	10.441	0.653

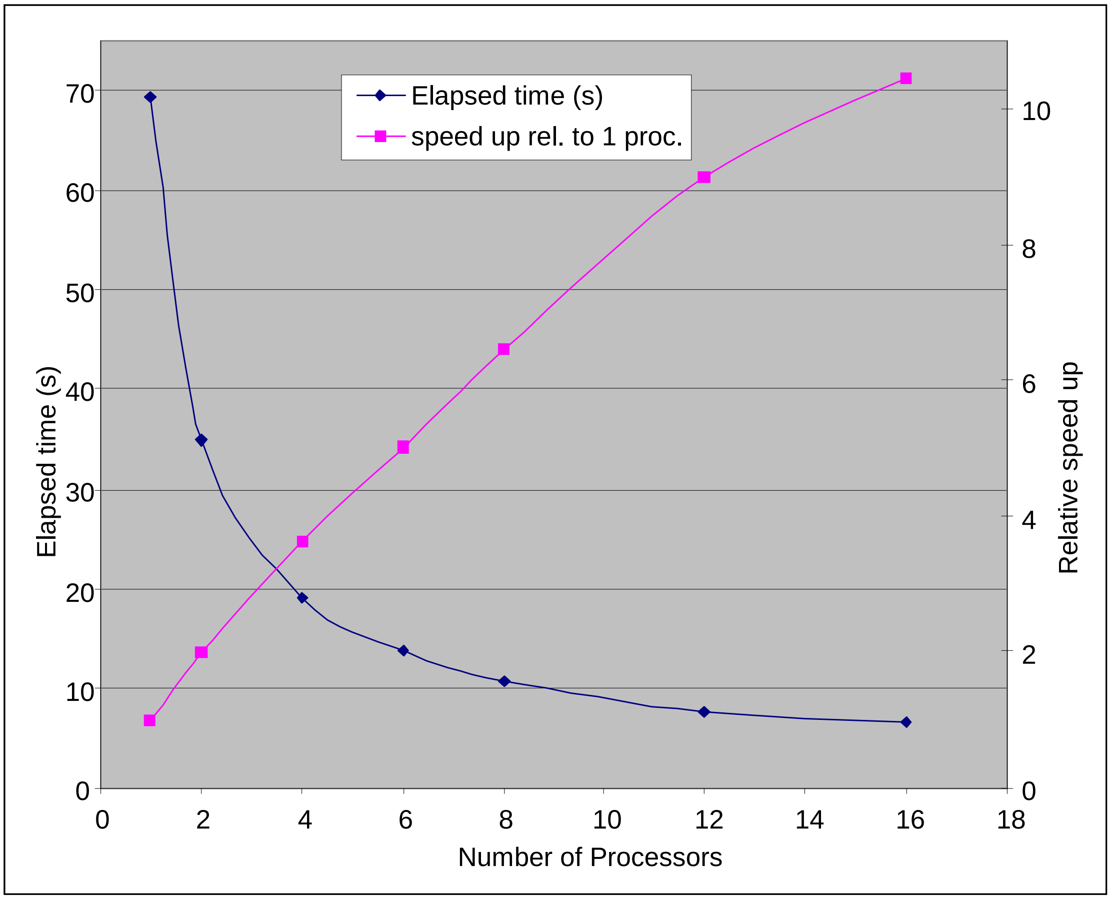


Figure 6.4: Elapsed time and relative speed up vs. number of processors

6.5 Case Study: Flat panel TV enclosure

In this case study, the proposed heat-reversible locator-snap joints are applied on an engineering example of a flat panel TV enclosure. The optimization problem is solved using serial and load balanced NSGA-II to test the proposed algorithm's performance.

Liquid Cristal Displays (LCDs) can be considered part of the modern life. They are currently appearing on everything from calculators, mobile phones to wide screen TVs. For instance, LCD TVs are of increasing demand. There was about 40 million flat panel TVs sold worldwide in 2005 and it is expected to rise to 100 million in 2009 [134]. At the same time, the technological advances are so rapid that millions of LCD screens are discarded each year. Unfortunately, Liquid crystals are potentially hazardous and it is unsafe to discard them without recycling. LCDs of a surface larger than 100 cm^2 are considered hazardous and have to be removed from WEEE [135] for proper disposal and removal of hazardous material. The European directive on WEEE requires LCDs and electrolyte capacitors present in the EOL electric and electronic equipment to be removed and treated separately [136]. In the USA, California's electronic waste recycling act [137] considered the following devices hazardous and cannot be discarded in the regular trash. Instead, they have to be sent to recycling plants for recycling, reuse and disposal [138]:

- Cathode Ray Tube (CRT) monitors and televisions.
- LCD desktop monitors and LCD televisions.
- Laptop computers and portable DVD players with LCD screens.
- Plasma televisions.

It is, therefore, important to have the disassembly process for such equipment economic by using joints that can disengage with less time and energy. In addition, LCD screens are good candidates for plastic recycling as their housings are made of large pieces of homogenous material. Using joints that allow clean separation of these housings is, as well, significant to get high quality plastics from recycling. Finally, using joints that can disengage non-destructively is essential to allow fast and economic service and repair of LCD screens.

As a result, a heat-reversible locator-snap system is used to join the front TV Bezel to the TV assembly in this case study. The bezel is assumed to be injection molded from Polystyrene, material properties are in Table 6.5. This case study also includes design for the lock-and-key concept using multiple heating locations at two different temperatures and the use of the double-latching snaps. It will also include the maximization of the distance between locators that constrain the same DOF to minimize the effect of tolerance stack-up, as mentioned in Chapter 3. The bezel geometry is 1100×660×50 mm with a wall thickness of 3 mm. Figure 6.5 shows the simplified model of the TV assembly.

Table 6.5: Material properties of polystyrene.

Property Name (units)	Value
Density (g/cm3)	1.050
Elasticity modulus (MPa) / Poisson Ratio	3600 / 0.353
Melting point (^{o}C)	240.0
Thermal expansion coefficient (μm/m.^{o}C)	70.00
Conductivity (W/m.^{o}K)	0.140

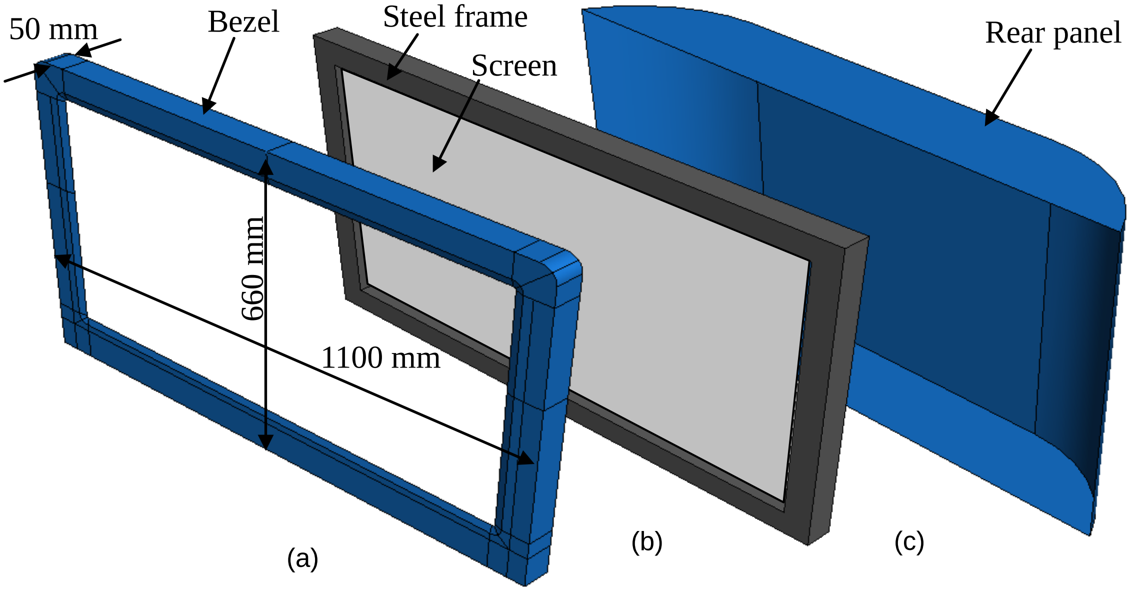


Figure 6.5: Simplified model of a flat panel TV: (a) front bezel, (b) steel frame with LCD screen, and (c) rear panel.

6.5.1 Assumptions and inputs

The series of assumptions and inputs given in this case study are summarized below:

- The optimization process will focus only on the Bezel geometry. Figure 6.6 shows the bezel geometry, the mating lines and the feasible heating region.
- Each edge of edges e_1, e_3, e_5 and e_7 can have up to three locators or snaps all of the same type. Whereas, each edge of edges e_2, e_4, e_6 and e_8 can have up to four locators or snaps all of the same type. This results in a total number of actual edges (=7) while the total number of apparent edges (n = 8).
- The locators and snaps are considered compliant; the stiffness values of the locators and snaps are given in the next sub-sections.
- The proposed double-latching snaps are used.

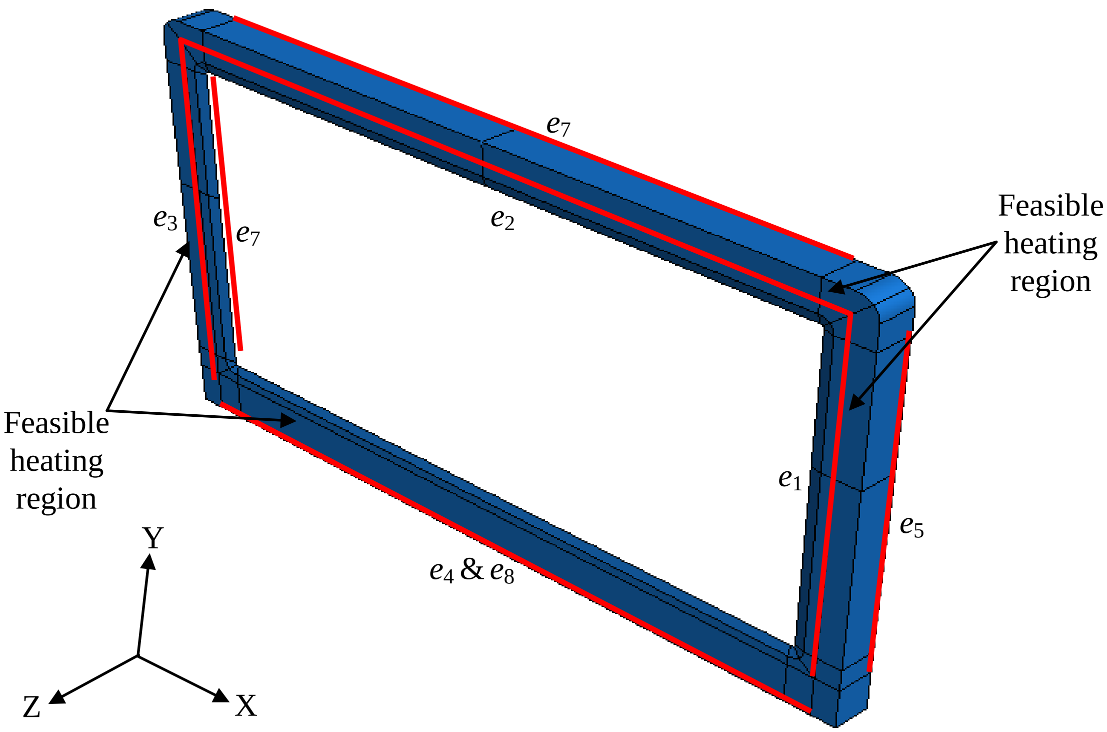


Figure 6.6: Front TV bezel with mating lines and feasible heating region (P_h) defined.

- The feasible heating region, P_h, is considered as the front surface of the bezel. All the three sides are not heated.
- Two local heating regions are used (m = 2). Each has a rectangular shape.
- Heating regions are not allowed to overlap.
- Local heating regions are heated or cooled to different temperatures. One rectangle is heated to T_{l1} = 120°C, while the other rectangle is cooled to T_{l2} = –80°C. The uniform heating temperature is T_u = 50°C. Room temperature is 20°C.
- During heating, free convection to the air (convection heat transfer coefficient = 8 W/m^2.°K) is considered as the only source of heat dissipation.
- The values of h_-, h_{lower} and h_{upper} are 1.5, 1.55 and 2.0 mm respectively.

- The locator and snap library used in this case study consists of only three locators (x-direction, y-direction and z-direction constraining locators) and one z-direction constraining snap as shown in Figure 6.7 (all in local coordinates).

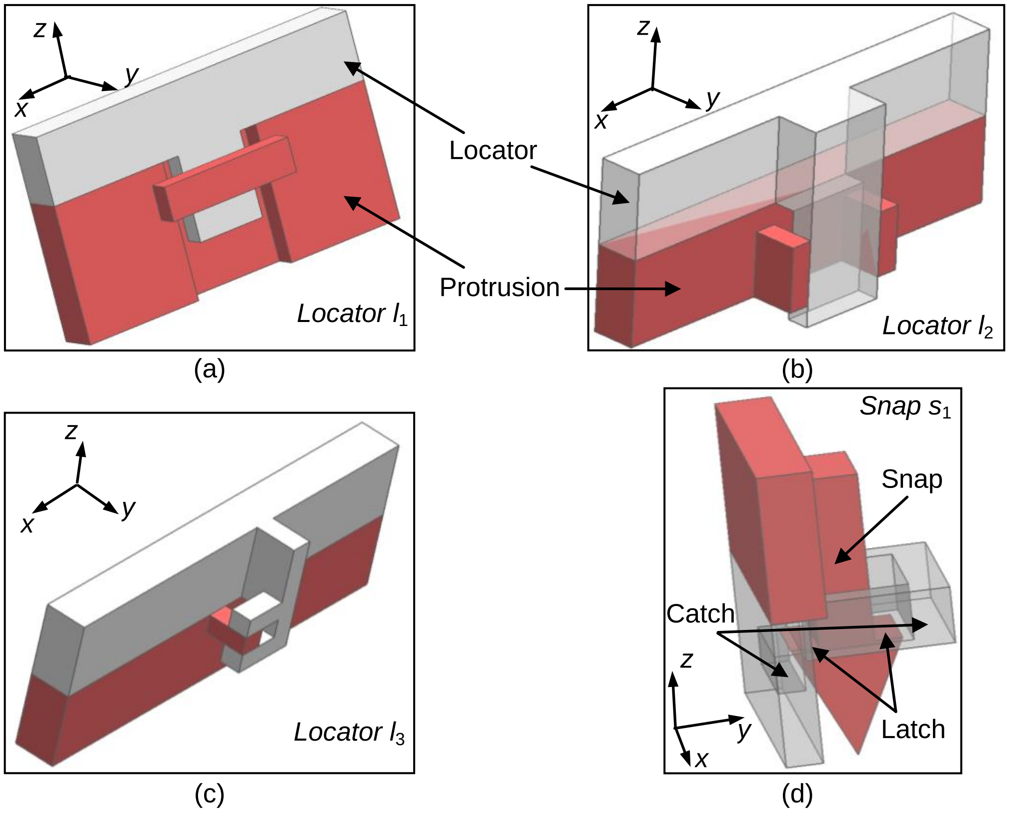


Figure 6.7: Locators and snaps used in the library in the current case study

6.5.1.1 Definition of the heating area

The feasible heating region, P_h, is a hollow rectangle. Using Cartesian coordinates, as in the previous case studies, will lead to large infeasible space, where zero nodes are heated. As a result, using polar coordinates to define the heating area would be more useful. Figure 6.8 shows an example of two local heating areas defined using the polar coordinates.

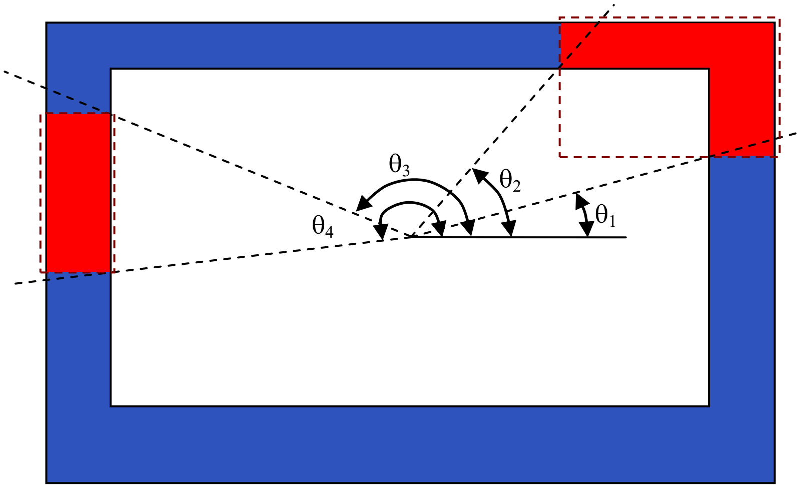


Figure 6.8: Front view of the front TV bezel two heating areas identified using polar coordinates.

6.5.1.2 Evaluation of the locator and snap stiffness

Since the bezel is large compared to the snap and locator sizes, the locators are not considered as rigid. Considering that the panel is injection-molded, the thicknesses of locators are kept the same as the panel thickness to avoid undesired injection molding defects such as sink marks on the external side. Figure 6.9 shows the dimensions of the locators and the snaps, and Table 6.6 shows the values of the equivalent stiffness for the locators and snaps, measured as described in Section 3.2.2

Table 6.6: Snap and locator stiffness values.

Locator Type	**Stiffness**
z-direction constraining locator or snap, Figure 3d	2703 N/mm
y-direction constraining locator, Figure 3c	303 N/mm
x-direction constraining locator, Figure 3b	3226 N/mm

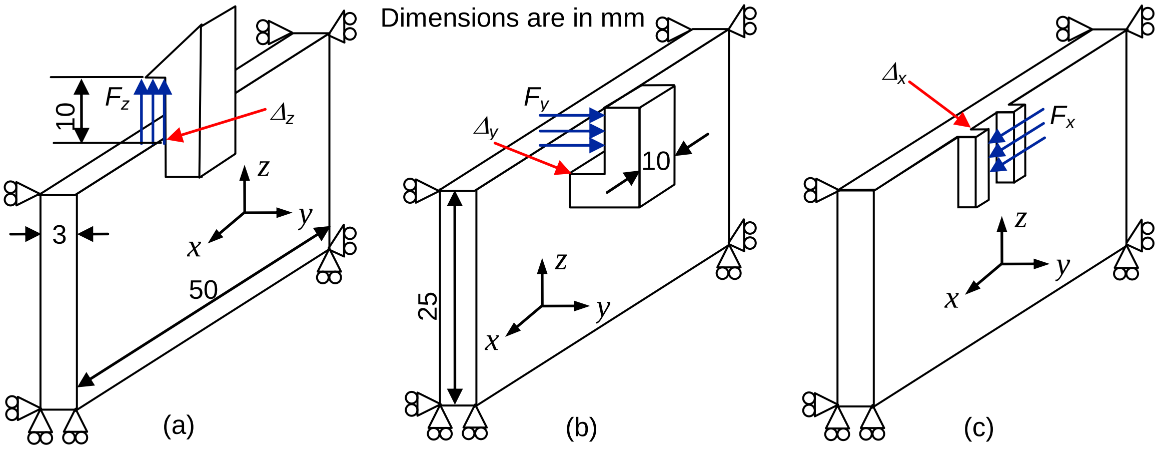


Figure 6.9: Measuring locator and snap stiffness: (a) *z*-direction constraining locator and snap, (b) *y*-direction constraining locator, and (c) *x*-direction constraining locator

6.5.1.3 Generation of the feasible locator and snap orientations

The set of feasible orientations of locator and snaps is pre-calculated as discussed in Section 3.2.3. The wrench matrices for the locators library, shown in Figure 6.7, in the local coordinate system are given in Equations (6.3) to (6.6); while the local-global transformation matrices for each edge are given in Equations (6.7) to (6.12). Only the orientations shown in Figure 6.7 are considered, i.e. the locators and snaps are not allowed to change their orientations with respect to the mating line they are attached to.

$$\mathbf{W}_{l_1}^{local} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{6.3}$$

$$\mathbf{W}_{l_2}^{local} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{6.4}$$

$$\mathbf{W}_{l_3}^{local} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \tag{6.5}$$

$$\mathbf{W}_{s_1}^{local} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \tag{6.6}$$

$$\mathbf{R}_{e1} = \begin{pmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \tag{6.7}$$

$$\mathbf{R}_{e2,e4,e8} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} \tag{6.8}$$

$$\mathbf{R}_{e3} = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \tag{6.9}$$

$$\mathbf{R}_{e5} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{6.10}$$

$$\mathbf{R}_{e6} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{6.11}$$

$$\mathbf{R}_{e7} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{6.12}$$

Knowing that the total number of edges (n = 8) and the total number of locator and snap types (m = 4), the total number of possible locator and snap combinations is (4^8 = 65536).

After testing all these combinations using Equations (3.17) and (3.18), only 1517 are feasible and are included in the feasible set F. It is obvious that checking the feasibility prior to running the optimization is crucial in this case study, as well, as only 2.2% of the total possible combinations are feasible.

6.5.2 Simultaneous optimization of locators/snaps and heating areas

The whole mathematical model is formulated as shown below:

$$\text{minimize } \left\{ f_1(\boldsymbol{y}),\ f_2(\boldsymbol{x},\boldsymbol{z}),\ f_3(\boldsymbol{x},\boldsymbol{z}),\ f_4(\boldsymbol{x},\boldsymbol{z}) \right\}$$

subject to:

$$\text{min\_disp\_loc}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l) \ > \ h_{lower}$$

$$\text{max\_disp\_loc}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l) \ < \ h_{upper}$$

$$\text{max\_disp\_unif}\,(\boldsymbol{x},nil,\boldsymbol{z},T_u) \ < \ h_{\_}$$

$$x_{ij} \in \left\{nil, L_i, L_i+1,\cdots,U_i\right\};\ i=1,\cdots,8;\ j=1,\cdots,d_i$$

$$\boldsymbol{y}_i = \left\{\theta_1,\theta_1\right\} \in \left\{0,360\right\}^2;\ i=1,2$$

$$\boldsymbol{z} \in F$$

For calculation of $f_1(\boldsymbol{y})$, the number of heated nodes is taken as a measure of the heated area. For calculation of $f_2(\boldsymbol{x}, \boldsymbol{z})$, which is a measure of the stiffness of the assembly, a uniformly distributed load of 10 N is applied normal to the TV screen in the +z direction, and the maximum displacement of the nodes on the mating line is obtained. $f_3(\boldsymbol{x}, \boldsymbol{z})$, is the reciprocal of the minimum the distance between locators/snaps that constrain the same DOF. It is used as a measure of the tolerance stack-up. $f_4(\boldsymbol{x}, \boldsymbol{z})$, is the measure of the maximum thermal stresses on the bezel due to the normal operation of the TV. When the TV is turned on the internal components start to heat up from room temperature until the whole assembly reaches steady state. As a result, the bezel tries to expand; yet, it is constrained by the locators and snaps. This results in thermal stresses that have to be minimized to avoid the bezel stress failure. Since the electronics inside the TV are usually asymmetrical, the temperature distribution along the bezel is also asymmetrical. It peaks at the top right corner with a value around 50 °C, and the minimum is at the bottom left with a value around 30 °C. The Temperature distribution of the bezel under normal working conditions is shown in Figure 6.10.

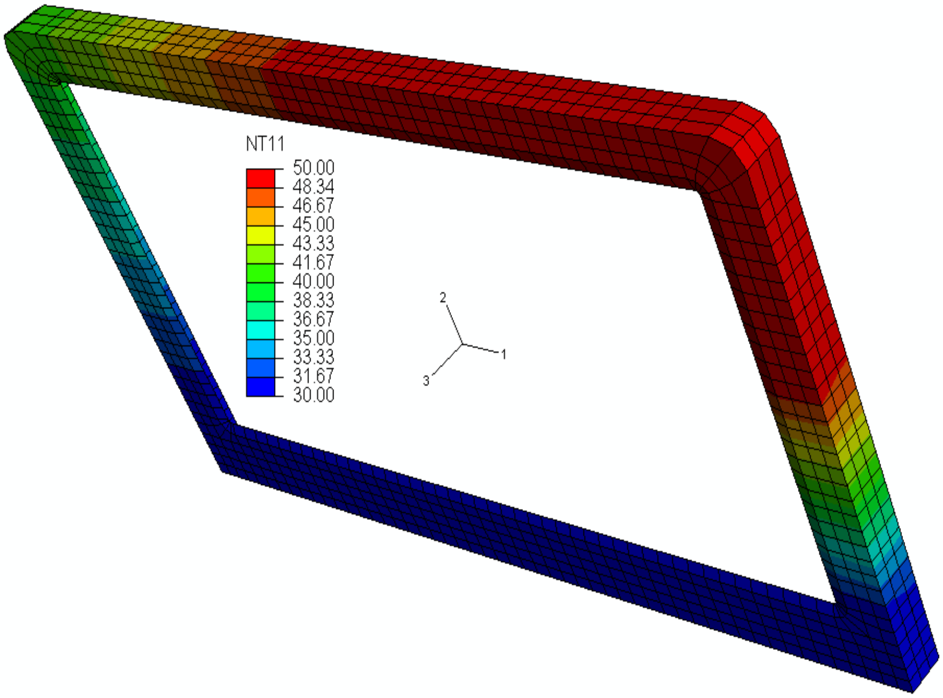


Figure 6.10: Temperature distribution at the bezel while the TV is turned on.

A penalty is applied to the objectives if the minimum displacement of all nodes with snaps under local heating at T_{l1} = 120°C and T_{l2} = –80°C is less than h_{+} = 1.55 mm or more than h_{++} = 2.0 mm. Another penalty is applied if the maximum displacement of all nodes with snaps under uniform heating T_u = 50 °C is more than h_{-} = 1.5 mm. The displacement constraints are written as penalty functions as given below:

$$g_1\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l\right)=\max\left(0,h_{lower}-\text{min\_disp\_loc}\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l\right)\right) \tag{6.13}$$

$$g_2\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l\right)=\max\left(0,\text{max\_disp\_loc}\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{T}_l\right)-h_{upper}\right) \tag{6.14}$$

$$g_3\left(\boldsymbol{x},\boldsymbol{z},T_u\right)=\max\left(0,\text{max\_disp\_unif}\left(\boldsymbol{x},\boldsymbol{z},T_u\right)-h_{-}\right) \tag{6.15}$$

Table 6.7: GA parameters used in the case study

Parameter	**Value**
Population size	200
Number of generations	200
Crossover probability	0.95
Mutation probability	0.05

The design variables ***x*** and ***z*** are treated as vectors with discrete variables, while ***y*** is treated as a vector with real variables. Thus, the optimization problem is solved using mixed, discrete and continuous, genetic algorithm. The parameters used in GA for this case study are shown in Table 6.7. A summary of the snap bounds and the heating temperature values is given in Table 6.8. The optimization problem was solved using a serial GA code running on a Windows workstation with a 3.4 GHz dual core Intel Pentium 4 processor. The time to finish the whole GA run was 6-7 days. The optimization problem was also solved using the proposed parallel GA code with and without load balancing on *nyx* using 1, 2, 4, 8 and 16 processors. The comparison between the runtimes of the optimization runs will be given later in this chapter.

Table 6.8: Summary of the snap bounds and the heating temperature values.

Parameter	Value
h_{lower}	1.55 mm
h_{upper}	2.00 mm
h_{-}	1.50 mm
T_{l1}	120 ^{o}C
T_{l2}	–80 ^{o}C
T_{u}	50 ^{o}C

Figure 6.11 Shows the extreme points for the 4-dimensional Pareto Optimal surface which exhibit a trade-off between the part compliance (mating line deformation), the amount of heating required (number of heated nodes), the thermal working stress and the distance between locators that constrain the same DOF (tolerance stack-up). All the results exhibit multiple heating locations and double-latch snaps as tools for the lock-and-key method. Double-latch snaps are more effective lock as they ensure that with excessive heating, the snap will become locked again.

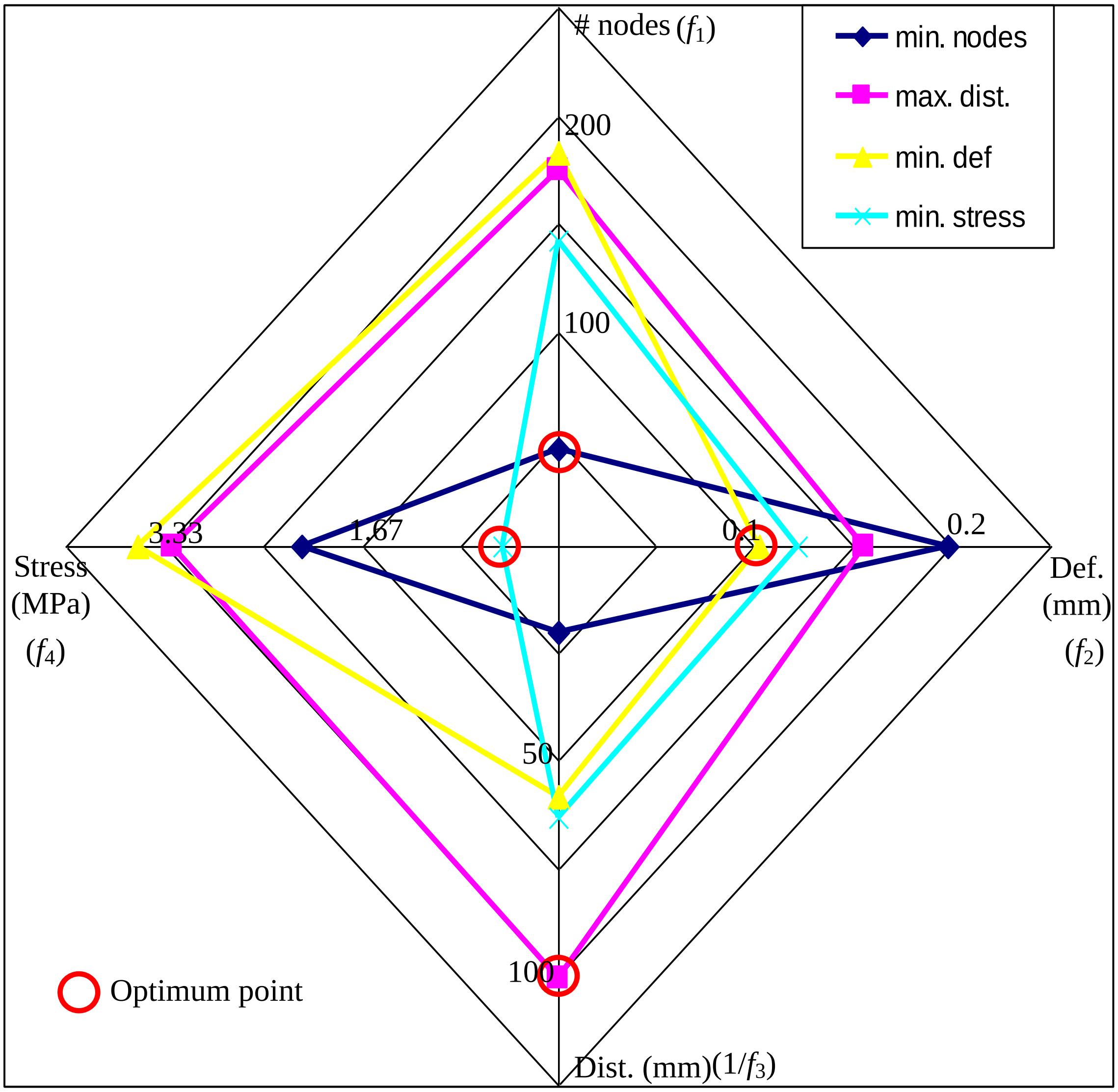


Figure 6.11: Spider-web diagram for the extreme points in the Pareto Optimal Solution.

Figure 6.12 shows the heating and cooling regions for the solution with the minimum local heat/cool area. The total number of heated and cooled nodes is 45 nodes, 20 are heated and 25 are cooled. Figure 6.13 shows optimal placement of locators and snaps and the response due to the local heating and cooling at the locations given in Figure 6.12. The maximum deformation at the mating line under a distributed load of 10 N that is applied normal to the TV screen in the +z direction is equal to (0.20 mm). The minimum distance between snaps that constrain the same DOF is 20.0 mm. The locator and snap positions are marked with solid squares. The lower edge has two different types of locators; one type is highlighted with hollow circle while the other type is highlighted by a hollow square. Figure 6.14 shows a schematic CAD drawing for the bezel. The bulging at the snap locations is inward and is equal to 1.78, 1.73 and 1.57 mm. The catches are placed on the shown part at the optimized snap locations; while the snaps are placed on the other part (not shown). The thermal stress is 2.17 MPa. The deformation at the snap locations due to uniform heating at 50°C is actually opposite to disassembly deformation; thus, the snaps will remain closed. If only the cooling area is activated, all snaps will remain closed (maximum deformation is 0.05 mm); whereas, if only the heating area is activated, one snap will remain closed with deformation (=1.53 mm).

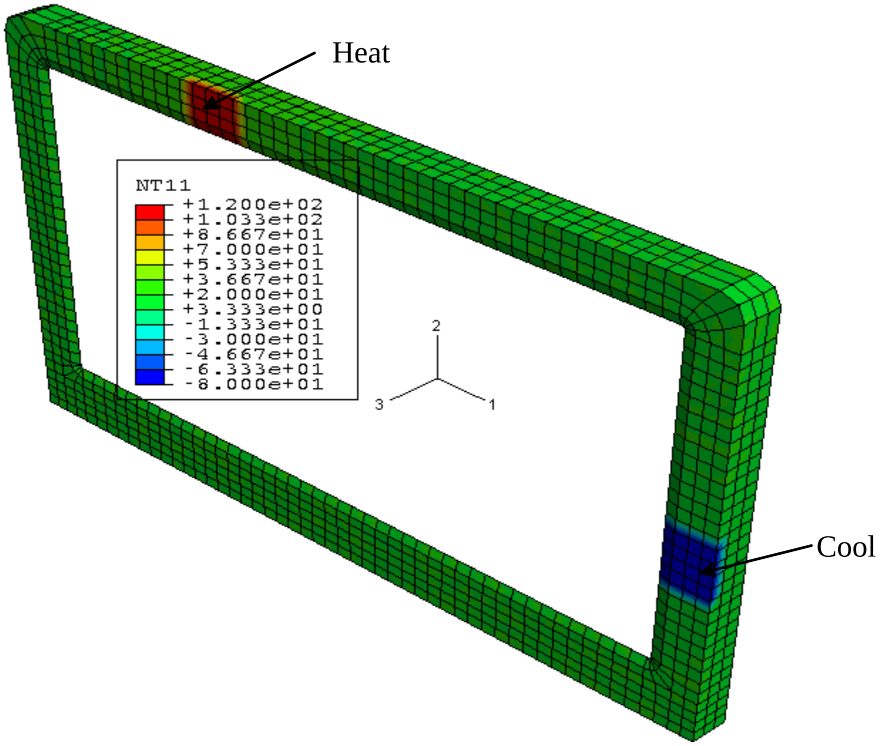


Figure 6.12: The heating and cooling regions for the solution with minimum local heat/cool area.

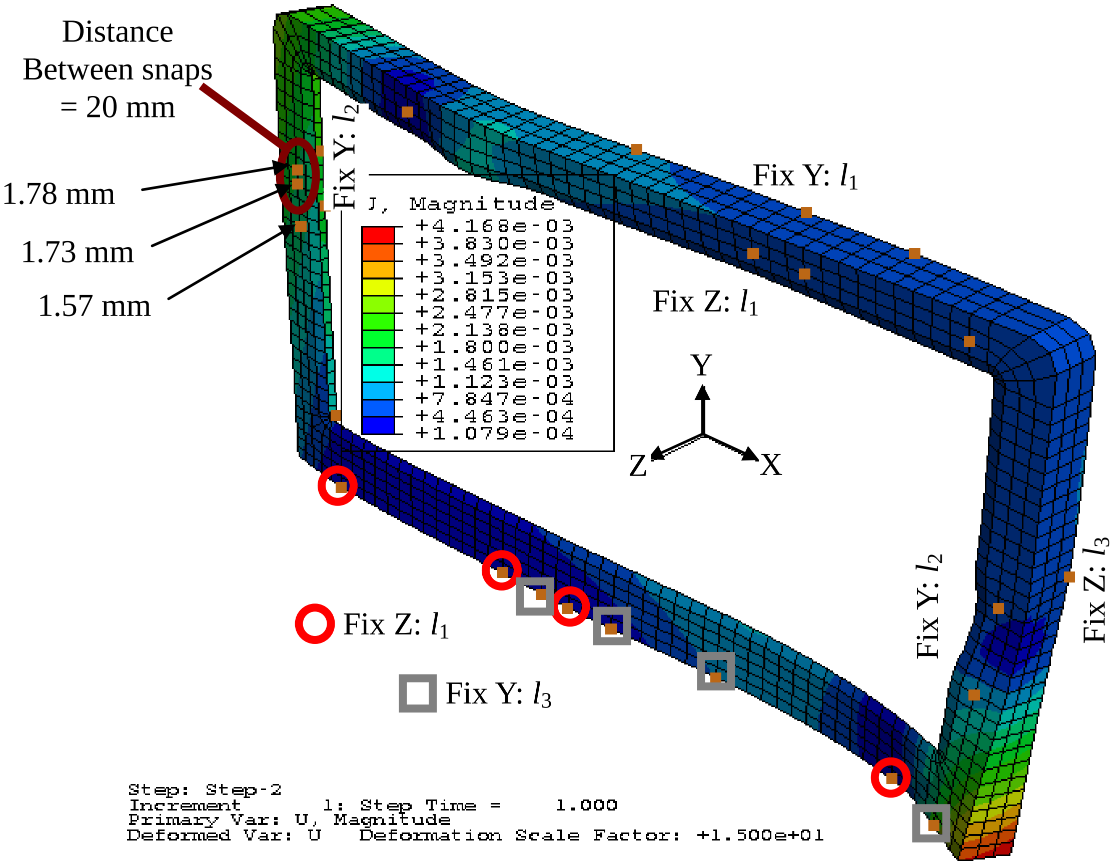


Figure 6.13: Locator and snap locations and the bezel deformation due to local heating/cooling for the solution with minimum local heat/cool area.

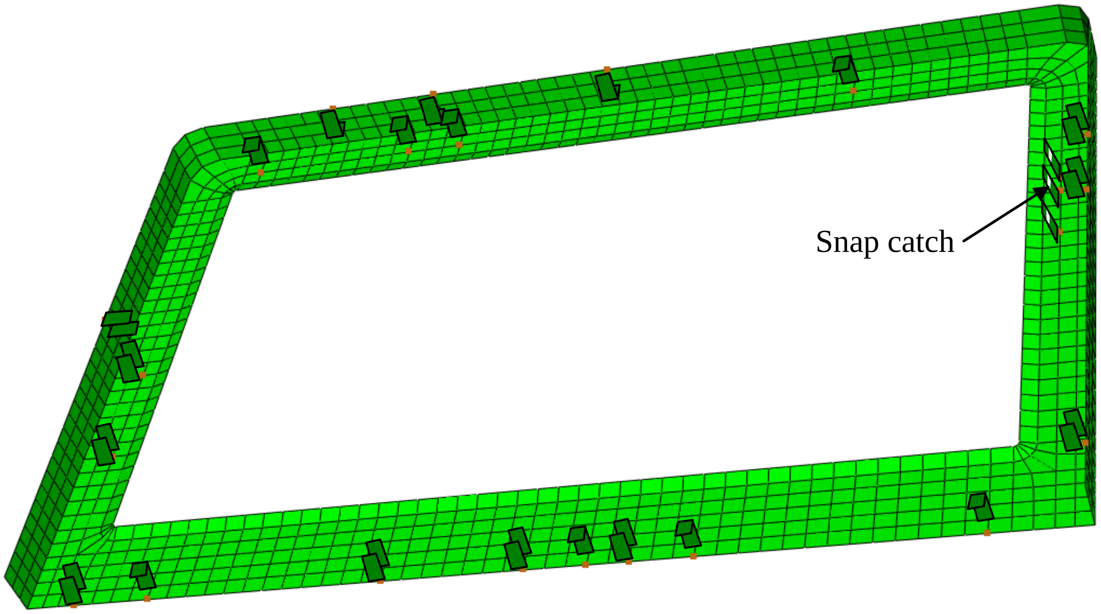


Figure 6.14: Schematic CAD drawing for the optimum bezel with minimum local heat/cool area.

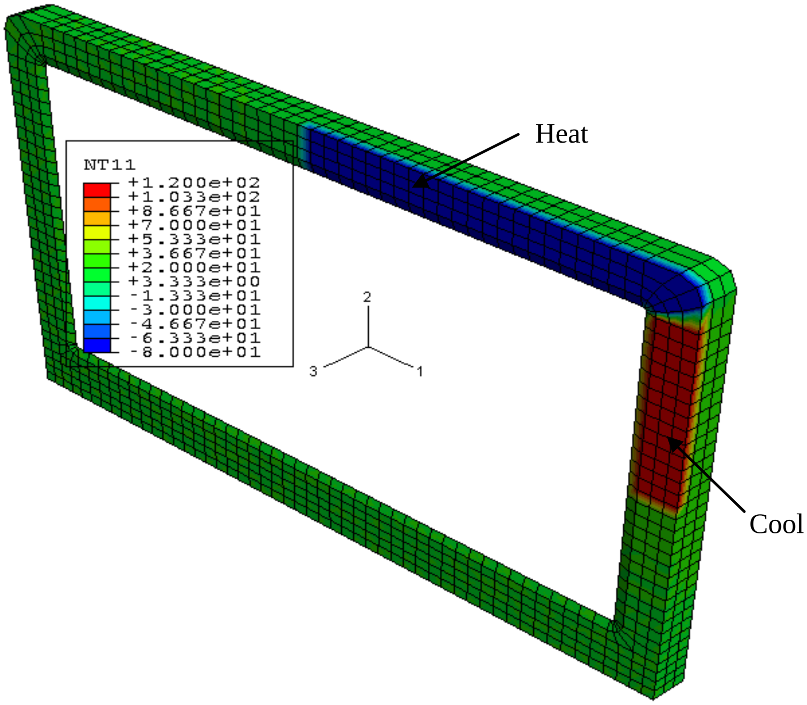


Figure 6.15: The heating and cooling regions for the solution with minimum deformation at the mating line.

Figure 6.15 shows the heating and cooling regions for the solution with the minimum deformation at the mating line. The total number of heated and cooled nodes is 183 nodes, 56 are heated and 127 are cooled. Figure 6.16 shows optimal placement of locators and snaps and the response due to the local heating and cooling at the locations given in Figure 6.15. The maximum deformation at the mating line under a distributed load of 10N that is applied normal to the TV screen in the +*z* direction is equal to (0.10 mm). The minimum distance between locators that constrain the same DOF is 58.0 mm. The locator and snap positions are marked with solid squares. The lower edge has snaps and locators; the snaps are highlighted with hollow circle while the locators are highlighted by a hollow square. Figure 6.17 shows a schematic CAD drawing for the bezel. The bulging at the four snap locations is inward and is equal to 1.87, 1.96, 1.98 and 1.99 mm. The catches are then placed on the shown part at the optimized snap locations; while the snaps are placed on the other part (not shown). The thermal stress is 3.56 MPa. The deformation at the snap locations due to uniform heating at 50°C is actually outwards (opposite to disassembly deformation); thus, the snaps will remain closed. If only the cooling area is activated, all snaps will remain closed (maximum deformation is 1.00mm); whereas, if only the heating area is activated, all snaps will remain closed (maximum deformation is 1.37 mm).

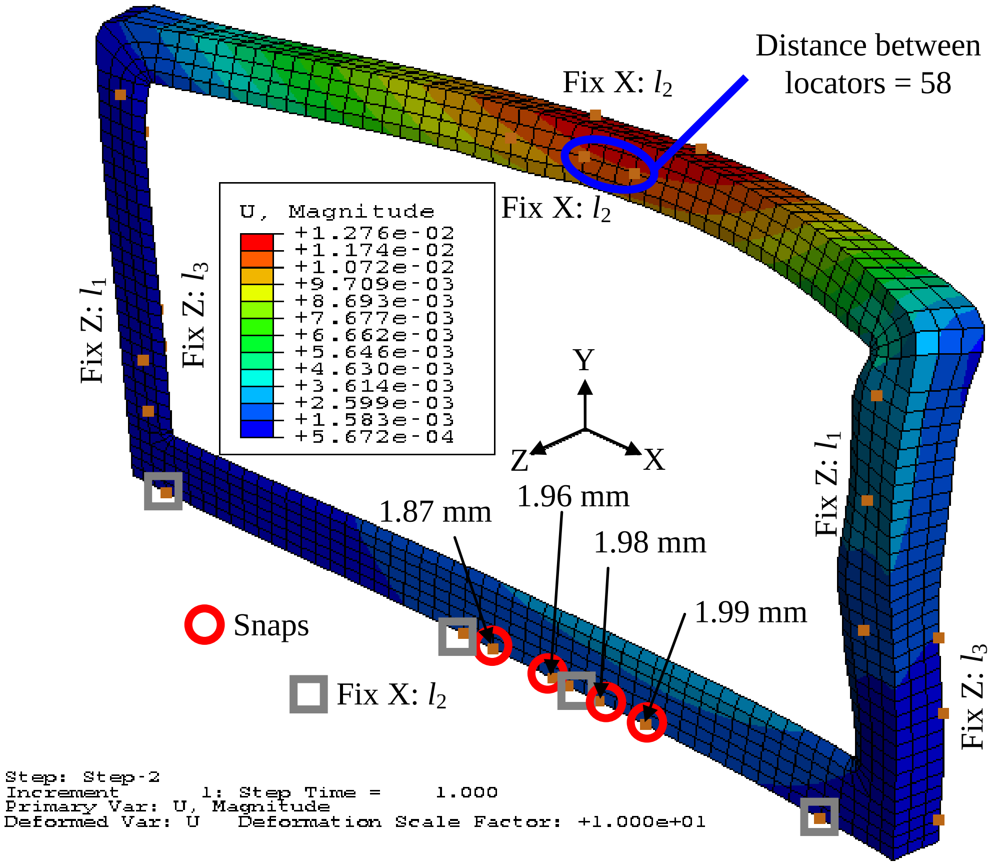


Figure 6.16: Locator and snap locations and the bezel deformation due to local heating/cooling for the solution with minimum deformation at the mating line.

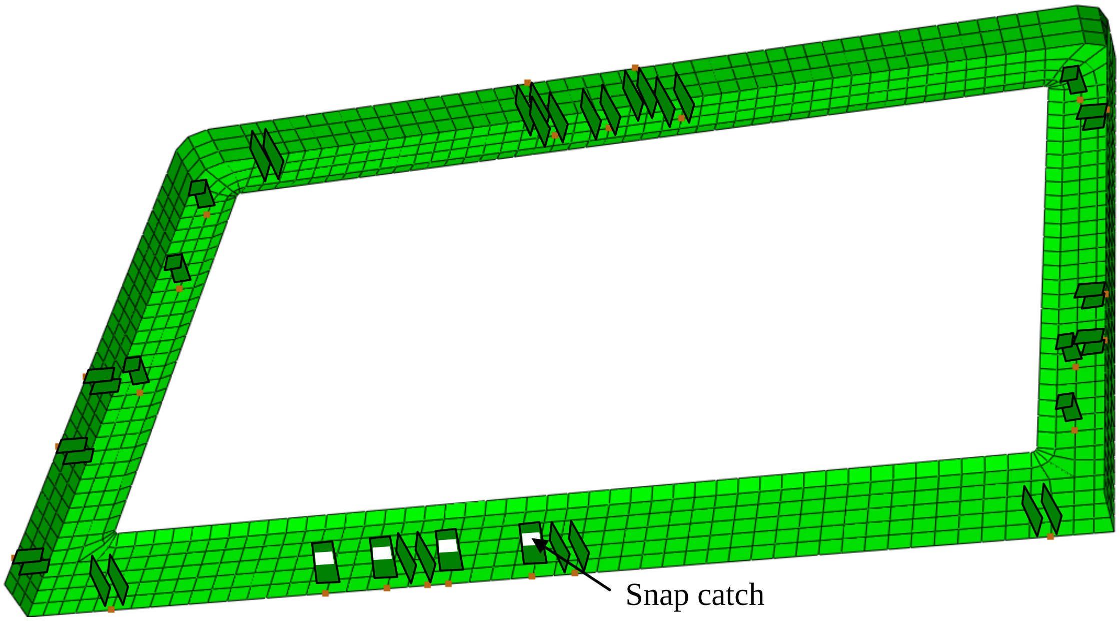


Figure 6.17: Schematic CAD drawing for the solution with minimum mating line deformation.

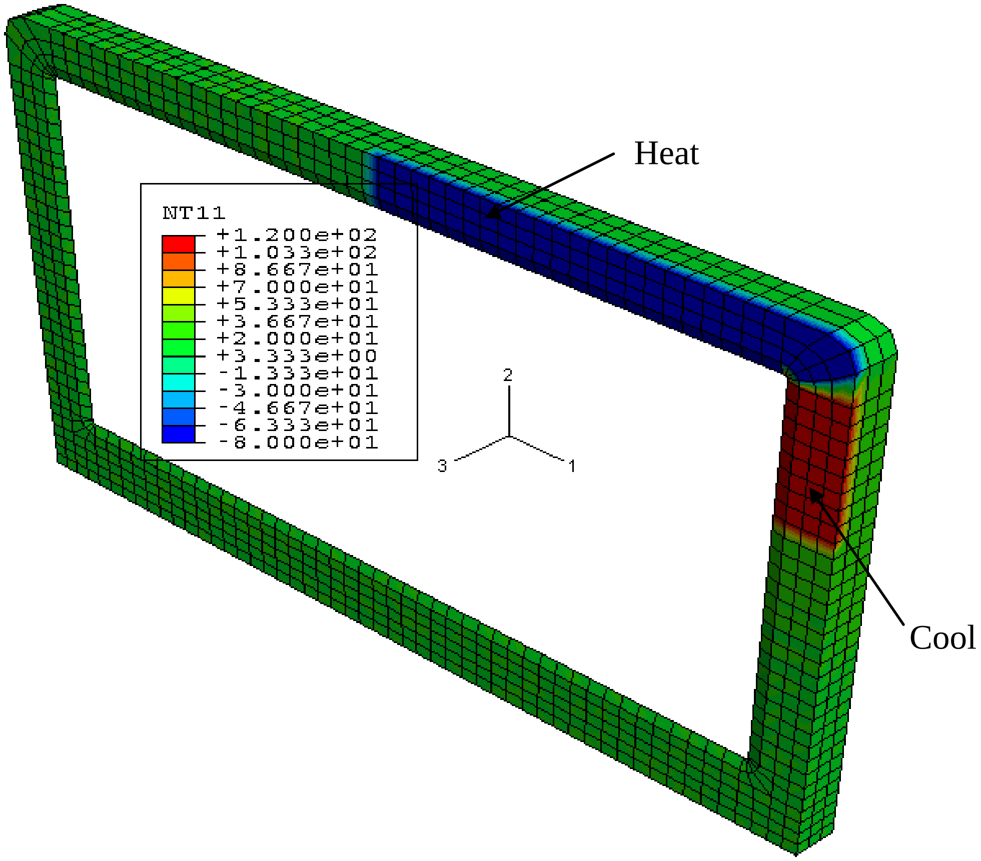


Figure 6.18: Heating and cooling regions for the solution with maximum distance between locators that constrain the same DOF.

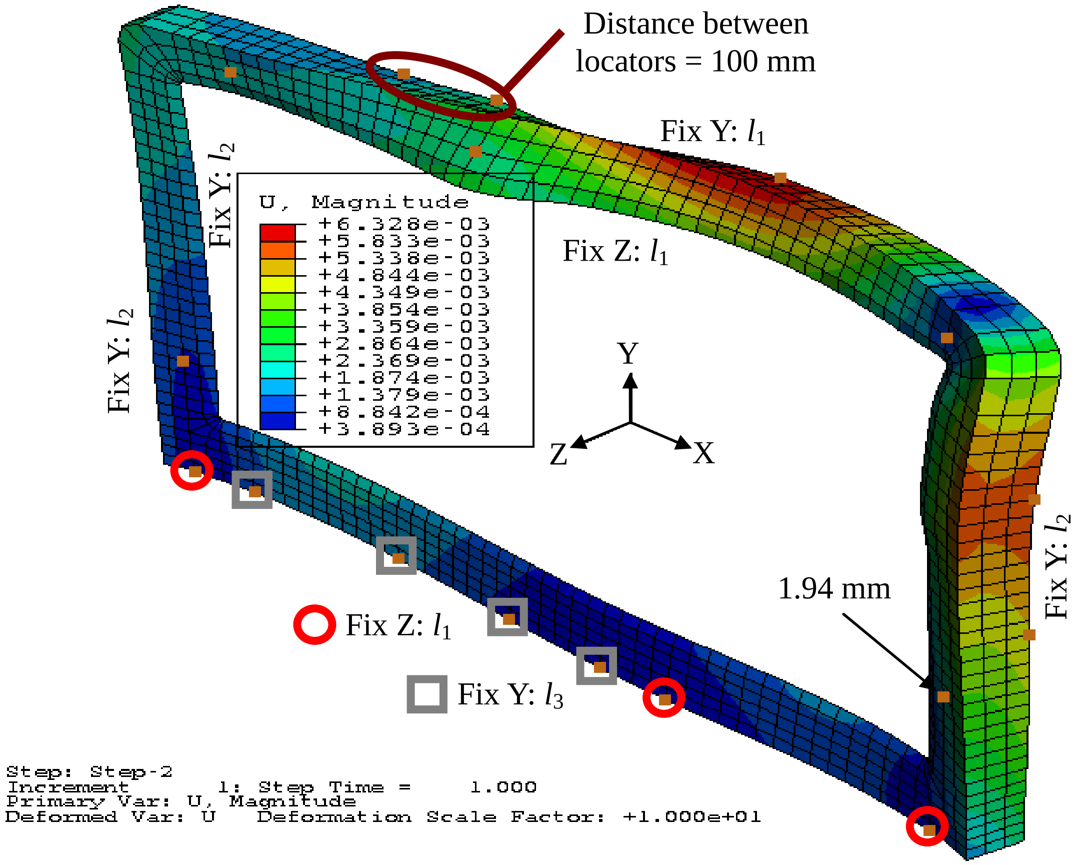


Figure 6.19: Locator and snap locations and the bezel deformation due to local heating/cooling for the solution with maximum distance between locators that constrain the same DOF.

Figure 6.18 show the heating and cooling regions for the solution with the maximum distance between conflicting locators. The total number of heated and cooled nodes is 175 nodes, 48 are heated and 127 are cooled. Figure 6.19 shows optimal placement of locators and snaps and the response due to the local heating and cooling at the locations given in Figure 6.18. The maximum deformation at the mating line under a distributed load of 10 N that is applied normal to the TV screen in the +*z* direction is equal to (0.15 mm). The minimum distance between locators that constrain the same DOF is 100.0 mm. The locator and snap positions are marked with solid squares. The lower edge has two different types of locators; one type is highlighted with hollow circle while the other type is highlighted by a hollow square. Figure 6.20 shows a schematic CAD drawing for the bezel. The bulging at the snap locations is outward and is equal to 1.94 mm. The catches are placed on the shown part at the optimized snap locations; while the snaps are placed on the other part (not shown). The thermal stress is 3.26 MPa. The deformation at the snap location due to uniform heating at 50^{o}C is 0.56 mm; thus, the snaps will remain closed. If only the cooling area or the heating area is activated, the snap will remain closed. Snap deformations in these cases are 0.74 mm and 1.2 mm respectively.

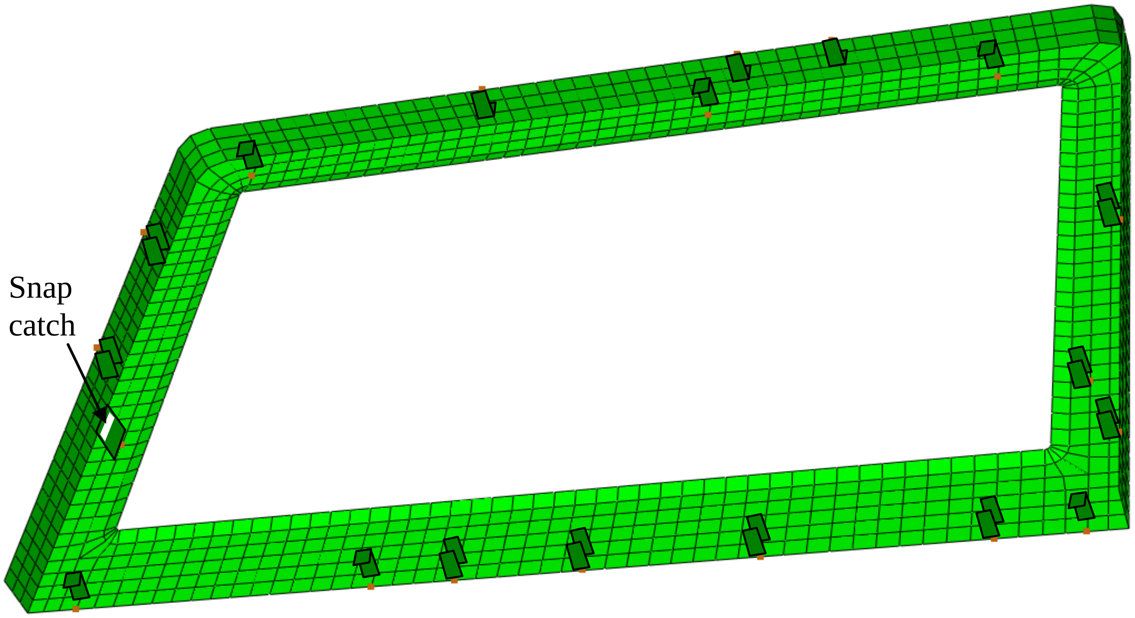


Figure 6.20: Schematic CAD drawing for the solution with maximum distance between locators that constrain the same DOF.

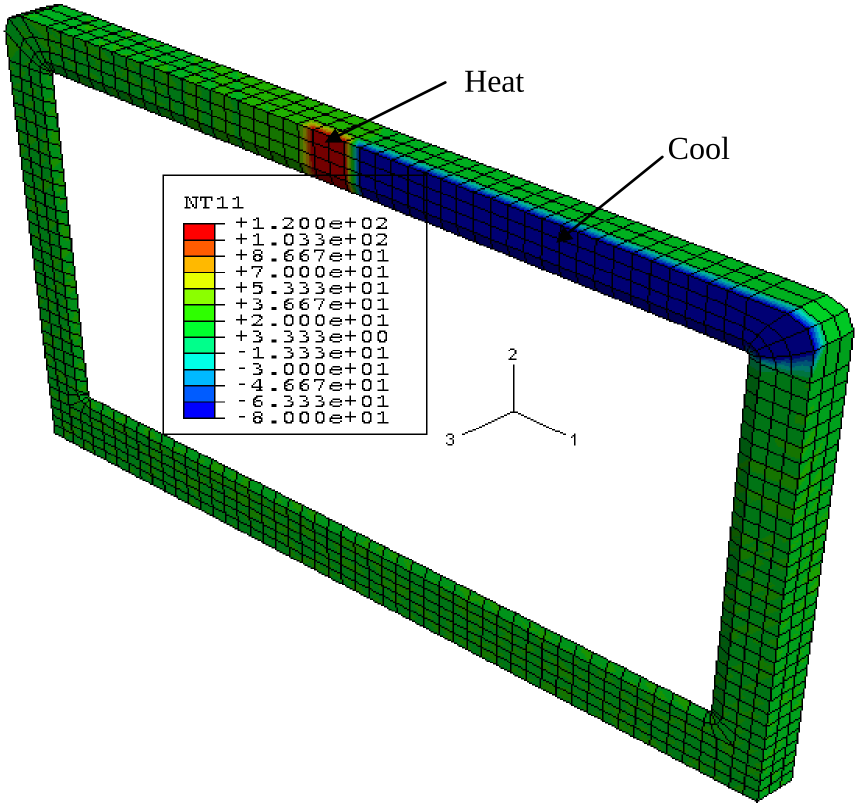


Figure 6.21: Heating and cooling regions for the solution with minimum thermal stress during the TV operation.

Figure 6.21 show the heating and cooling regions for the solution with the minimum thermal stress during the TV operation. The total number of heated and cooled nodes is 142 nodes, 15 are heated and 127 are cooled. Figure 6.22 shows optimal placement of locators and snaps and the response due to the local heating and cooling at the locations given in Figure 6.21. The maximum deformation at the mating line under a distributed load of 10 N that is applied normal to the TV screen in the +z direction is equal to (0.12 mm). The minimum distance between locators that constrain the same DOF is 63.1 mm. The locator and snap positions are marked with solid squares. The lower edge has snaps and locators; the snaps are highlighted with hollow circle while the locators are highlighted by a hollow square. Figure 6.23 shows a schematic CAD drawing for the bezel. The bulging at the snap locations is inward and is equal to 1.70 and 1.82 mm. The catches are placed on the shown part at the optimized snap locations; while the snaps are placed on the other part (not shown). The thermal stress is 0.47 MPa. The deformation at the snap locations due to uniform heating at 50°C is actually outwards (opposite to disassembly deformation); thus, the snaps will remain closed. If only the cooling area is activated, one snap will remain closed (snap deformation = 2.1 mm > 2 mm); whereas, if only the heating area is activated, all snaps will remain closed (maximum deformation is 0.25 mm).

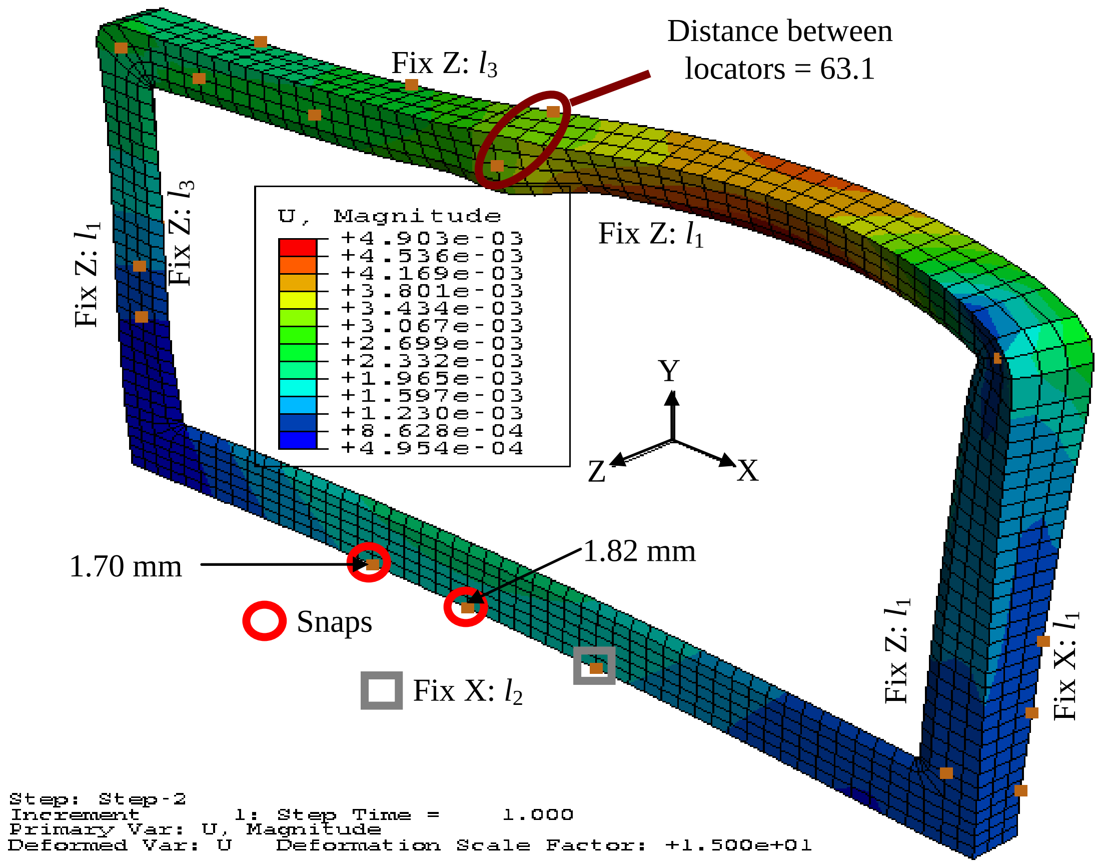


Figure 6.22: Locator and snap locations and the bezel deformation due to local heating/cooling for the solution with minimum thermal stress during the TV operation.

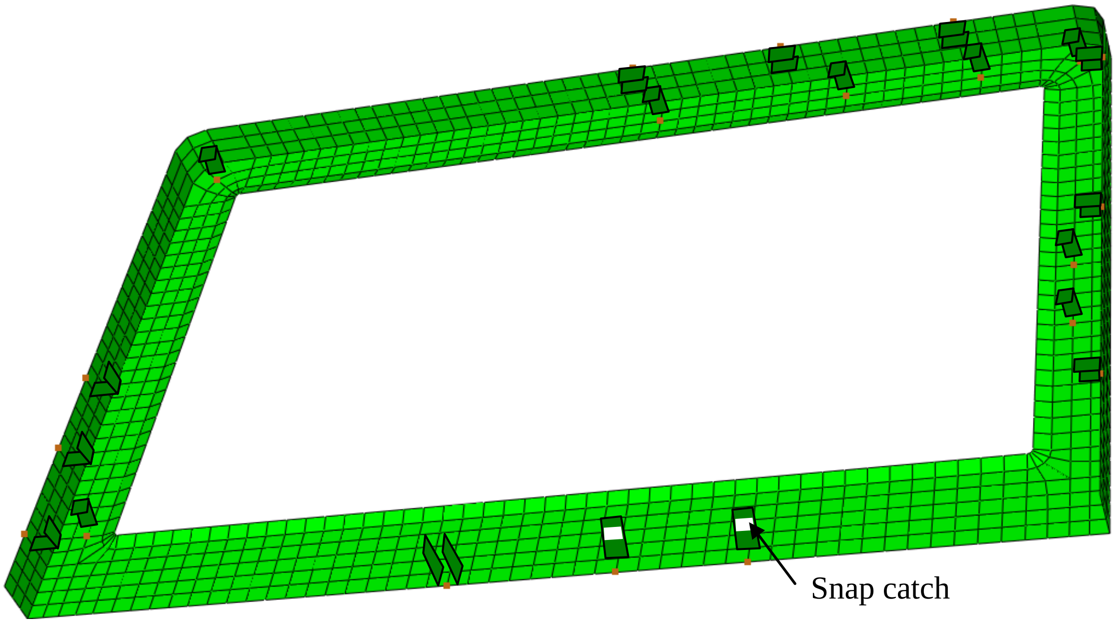


Figure 6.23: Schematic CAD drawing for the solution with minimum thermal stress during the TV operation

6.5.3 Parallel GA performance

The optimization problem for the case study was solved on the parallel computing grid, *nyx*, using 1, 2, 4, 8 and 16 processors. Two parallel GA codes were used, using equal distribution of the population over all the processors, and using the proposed active load-balancing code. The GA parameters used in these experiments are given in Table 6.9.

Table 6.9: GA parameters used in the case study

Parameter	**Value**
Population size	120
Number of generations	120
Crossover probability	0.90
Mutation probability	0.05

Table 6.10 and Table 6.11 have the elapsed times, speed up and efficiencies with respect to the number of processors using parallel GA with equal distribution of the population and with the proposed load balancing scheme. Both results are also summarized in the graph in Figure 6.24. Although it is unfair to simply compare these results as the used processor speeds during each GA run is unknown; the results seem very practical with linear speed-up with respect to the number of processors. In addition, using load balancing in this case study demonstrated better efficiency than simply dividing the whole population over the number of working processors evenly. Finally, the efficiency values of the proposed load balancing scheme in this case study is consistent with the efficiency values obtained in Table 6.4 although the objective function evaluation times and the population sizes vary significantly. This can be taken as a proof of the robustness of the proposed algorithm.

Table 6.10: Parallel GA with equal distribution of population elapsed time, speed up and efficiency w.r.t the number of processors

# of processors	Elapsed time (hrs)	Relative speed up	Efficiency
1	225	1	1
2	202	1.11	0.56
4	112	2.01	0.5
8	53	4.25	0.53
16	29	7.76	0.48

Table 6.11: Parallel GA with load balancing elapsed time, speed up and efficiency w.r.t the number of processors

# of processors	Elapsed time (hrs)	Relative speed up	Efficiency
1	225	1	1
2	115	1.96	0.98
4	68	3.31	0.83
8	40	5.63	0.7
16	23	9.78	0.61

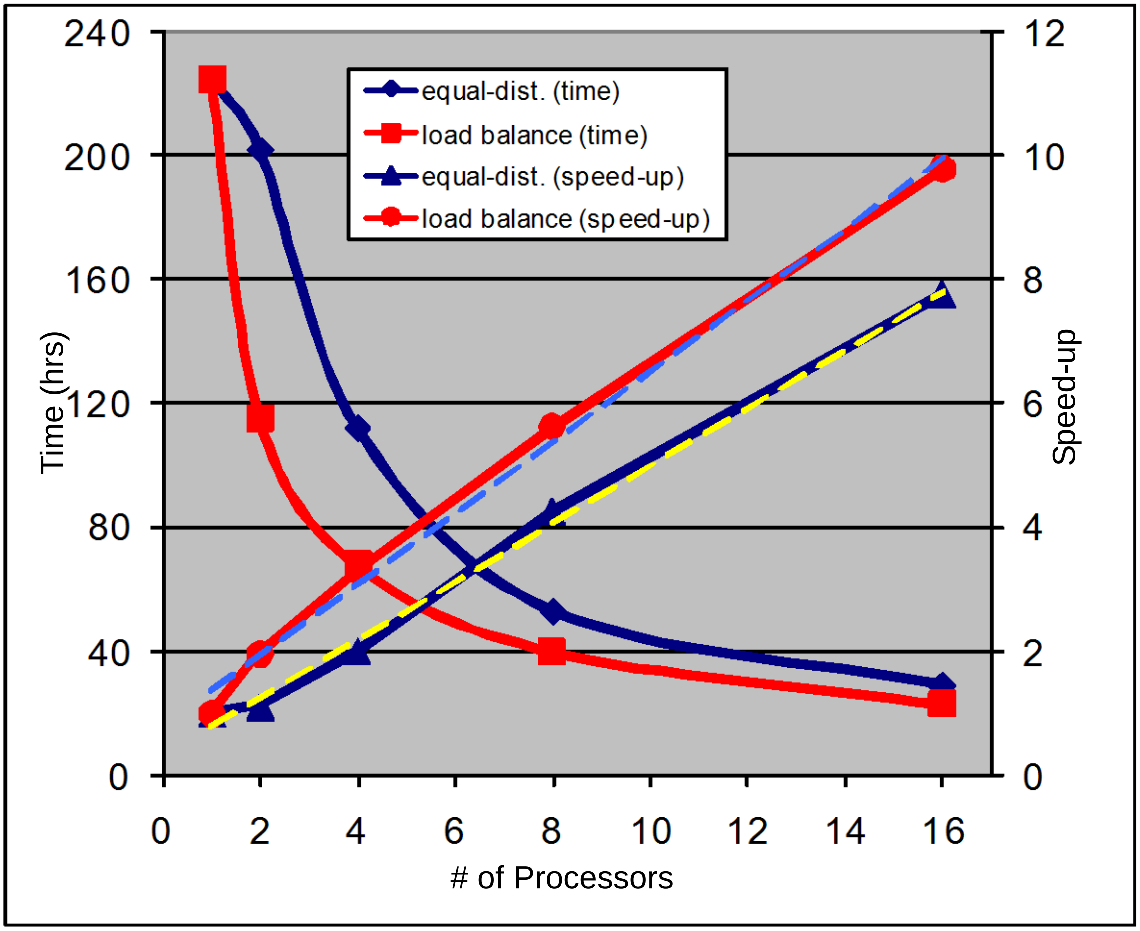


Figure 6.24: Elapsed times and relative speed-ups vs. the number of processors.

It should be noted that all the parallel GA runs converged to near optimum solutions; yet, the results presented in Section 6.5.2 are better. This is easily justifiable as the population size and the number of generations were higher in Section 6.5.2 than in the parallel GA runs (200 and 200 vs. 120 and 120).

6.6 Conclusion

A parallel version of MOGA based on non dominated sorted genetic algorithms (NSGA-II) was developed with dynamic load balancing for general purpose optimization problems. Two parallelization schemes were embedded in the code to choose from, 1) manager works its share of evaluating the objective function and on the GA serial overhead work too, 2) load balanced manager-worker scheme where the manager keeps sending small portions of the population to the worker processors until the whole population is evaluated. The code takes the advantage of knowing the behavior in the previous generation to decide which of the 2 schemes to use to maximize efficiency. The proposed code can, therefore, reduce the design cycle time for various consumer products. It can also allow designers to include optimization in the design process of electric products that suffer from very short design cycles.

A practical case study for joining a flat panel TV bezel to the TV assembly using the proposed heat-reversible locator-snap system, developed in Chapter 3, was analyzed using serial GA and the proposed parallel GA to demonstrate the proposed code’s effectiveness. The resulting Pareto-optimal solutions exhibit alternative designs with different trade-offs between structural stiffness and thermal stresses during snap engagement, and the heating area necessary for snap disengagement. In this case study, the feasible heating region was considered as the whole bezel surface. This surface is easily accessible; yet, the heating and the cooling temperatures were selected without verifying their effects on the components inside the TV, for instance the LCD screen performance may be affected. It is recommended to verify the feasibility of these temperatures and modify them if needed. The modified bezel designs can be implemented in reality provided that the social, environmental and financial benefits and the low disassembly costs exceed the increase in the manufacturing costs due the use of more complex dies. Similar to the previous case study, robust optimization can be used to make sure that the resulting designs are less prone to noise factors such as environmental variations during the product's usage, manufacturing variations, and component deterioration. The proposed load balanced parallel GA proved to be more efficient than just parallelizing the GA algorithm by dividing the population evenly over the number of processors.

Chapter 7: Closure

7.1 Summary

The use of joints that can disengage with minimum labor, part damage, and material contamination is critical to ensure effective service, part reuse, and material recycling. In this dissertation, we proposed the concept of heat-reversible locator-snap systems as joints that can satisfy the aforementioned requirements. The assembly process is analogous to a regular locator-snap system. During disassembly, snaps can be released by the application of localized heat at specific locations. The in-plane thermal expansion – constrained by locators – and the temperature gradient along the wall thickness are exploited to realize the out-of-plane bulging of the enclosure wall that releases the snaps.

In this dissertation, a general computational method for designing the fail-safe lock-and-key heat-reversible locator-snap systems that allows easy, clean and non-destructive disassembly was developed. The lock-and-key concept is realized by double-latch snaps that require force within a certain range to disengage, and multiple snaps that require heating multiple locations at different temperatures to disengage. The lock-and-key concept guarantees that the joints are fail safe; i.e. it ensures that the components do not disengage accidentally during use or by unauthorized personnel, but disengage easily when the right procedure is followed. Based on the proposed method, a generic optimization problem was posed to find the orientations, numbers, and locations of locators and snaps, and the numbers, locations, and sizes of heating areas, which realize the release of snaps only when the desired procedure is followed with minimum heating and maximum stiffness, while satisfying motion and structural requirements. Screw Theory was utilized as a tool to perform the motion and constraint analyses required to pre-calculate the set of feasible orientations of locators and snaps that are examined during optimization. A Multi-Objective Genetic Algorithm based on the Fast Elitist Non-dominated Sorted Genetic Algorithm (NSGA-II) was used an optimizer to solve the posed generic optimization problem.

The evaluation of objective functions requires Finite Element Analyses (FEA) that are time-consuming by nature; in addition, FEA is solved using iterative procedures that do not necessarily converge at the same time. As a result we developed a parallel version, using a manager-worker scheme of NSGA-II with dynamic load balancing, to solve the generic optimization problem efficiently. The proposed algorithm takes the advantage of knowing the behavior in the previous generations to decide which of the two schemes to use to evaluate the objective functions in the current generation; it thereby maximizes efficiency. In Scheme 1, the population is divided evenly over all the processors, including the manager. In scheme 2, the population is divided into patches that are sent to worker processors based on their availability.

Since the proposed heat-reversible locator-snap systems do not require any special tools or actuators to implement, they can be applied on various products ranging from automotive bodies to consumer electronics as in the given case studies. The products should have relatively large thermal expansion coefficient and has relatively large surfaces. This ensures that heating can induce enough deflection to release snaps. The method may not applicable to products that have dimensions in the order of few centimeters. In the first case study, heat-reversible snap joints were proposed as a joining method between internal frames and external panels in automotive bodies. Next, the proposed locator-snap system was applied on two case studies for consumer electric products, namely, a T-shaped DVD player enclosure and a flat panel TV enclosure. In the latter, the developed parallel GA code was used. Finally, to demonstrate the generality of method, a case study was done on a model with complex mating line geometry. In all case studies, the resulting Pareto-optimal solutions resulted in alternative designs with different trade-offs between the design objectives while satisfying all constraints.

7.2 Contributions

The contributions of this research can be summarized as:

- The introduction of heat-reversible locator-snap systems which are similar to regular locators/snaps in literature, yet they disengage non-destructively due to heat application at certain locations. Unlike the previous work on heat-reversible cantilever snaps, the heat-reversible locator-snap system focuses on the interactions between the locators, snaps, mating line and enclosure geometry as a whole, not just limited to the design of the snap in isolation.

- The development of the lock-and-key concept that is realized by the use of the proposed double latching snaps and the use of multiple heating locations with different temperatures. This makes the disassembly process hard for unauthorized personnel.
- The development of a general computational method for designing the fail-safe lock-and-key heat-reversible locator-snap systems that allows easy, clean and non-destructive disassembly.
- The development of an active load balanced Multi-Objective Genetic Algorithm as a powerful optimization tool that will help reduce the computational time during optimization extensively, especially with the increasing use of parallel computing.

7.3 Future work

Based on the results of this research, the following topics can be considered to expand the scope of the proposed method:

- More work can be done on the lock-and-key concept. For instance, the development of a sequential disassembly process, like heating two locations sequentially, or application of force at certain location after heating, *etc*.
- Due to the complexity of including tolerance analysis within the optimization algorithm, we suggested maximizing the distance between conflicting locators in

this research. Any work to include more realistic tolerance analysis is recommended. For instance, developing a method to generate simplified sensitivity matrices based on the orientation of the locators and snaps that can be generated offline prior to optimization (similar to the generation of the feasible orientation set).

- Robust optimization can be included to make sure that the resulting designs are less prone to noise factors such as environmental variations during the product's usage, manufacturing variations, and component deterioration.
- Work on using variable mating lines can be done. Different mating lines can be examined prior to optimization and the best mating line is selected.
- This work can be integrated with the work in the area of product-embedded disassembly, so that the removal of the product enclosures becomes the triggering action for the product disassembly.
- In the field of load balanced genetic algorithms, the following work is suggested:

    - Generate rules to actively adjust the load balancing parameters during optimization. For instance, the load balancing parameters can be a function of the ratio of the number of feasible solutions to the number of infeasible solutions as this ratio varies as the optimization algorithm evolves.
    - Study the effect of the location of the design variables in the design space on the objective functions evaluations time.

Appendices

Appendix A: Screw Theory

Screw Theory is a way to express velocities and forces in three dimensional space, combining both rotational and translational parts. Any given displacement of a rigid body can be defined by a rotation about an axis and a translation parallel to that axis. Definitions of some of the screw theory terminology used in this thesis are given below and are summarized from [93], [100], [139] and [140]:

- **Screw:** A screw is a straight line with which a definite linear magnitude, termed pitch, is associated. A screw is commonly represented by *screw coordinates* (Figure A.1) as a pair of two row vectors $\boldsymbol{S} = (\boldsymbol{s}, \boldsymbol{s}_0)$ in 3D Cartesian coordinates where $\boldsymbol{s}$ is a unit vector parallel to the screw axis and $\boldsymbol{s}_0$ is given as:

$$\boldsymbol{s}_0 = \boldsymbol{r} \times \boldsymbol{s} + p\boldsymbol{s} \tag{A.1}$$

Where $\boldsymbol{r}$ is the position vector of a point on the screw axis and p is the pitch. p can be obtained using the following equation:

$$p = \frac{\boldsymbol{s}.\boldsymbol{s}_0}{\boldsymbol{s}.\boldsymbol{s}} \tag{A.2}$$

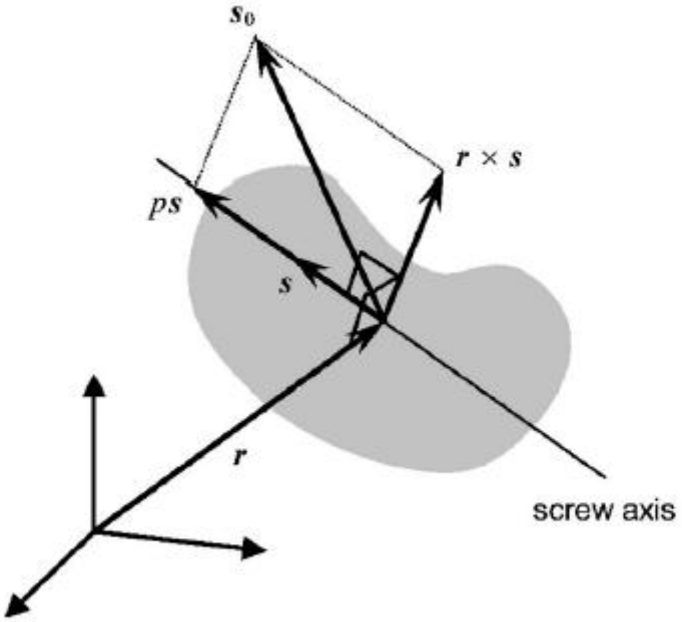


Figure A.1: Screw representation using screw coordinates.

- **Pure Rotation Screw:** If the pitch of the screw is zero, the motion is pure rotation. Pure rotation screws follow Equation (A.1).
- **Pure Translation Screw:** If the pitch approaches infinity, the motion is pure translation. Pure translation screws do not follow Equation (A.1), instead it is denoted by a zero vector to represent $\boldsymbol{s}$, while $\boldsymbol{s}_0$ represents the unit vector parallel to the screw axis.
- **Twist:** A body is said to receive a twist about a screw when it is uniformly rotated about the screw through a distance equal to the pitch and an angular rotation, Figure A.2. Using screw coordinates, a twist is denoted as $\boldsymbol{T} = (\boldsymbol{\omega}, \boldsymbol{v})$, where $\boldsymbol{\omega}$ is the angular velocity and $\boldsymbol{v}$ is the linear velocity of a point on the body located at the origin of the global reference frame.

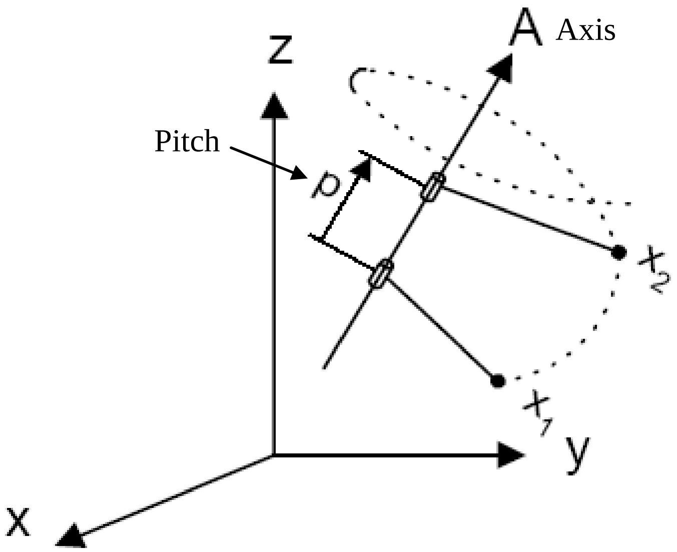


Figure A.2: Geometric example of twist.

- **Wrench:** A wrench is the canonical form to represent the system of forces on a rigid body on a screw. Using screw coordinates, a wrench can be represented as ***W*** = (***f***, ***m***), where ***f*** and *m* are force and moment vectors, respectively, that a point on the body located at the origin of the global reference frame should resist.
- **Reciprocal Screws:** if a wrench acts on a rigid body in such a way that it produces no work while the body is undergoing an infinitesimal twist, the two screws are, then, said to be reciprocal screws. Mathematically, two screws, $\boldsymbol{S}_1 = (\boldsymbol{s}_1, \boldsymbol{s}_{01})$ and $\boldsymbol{S}_2 = (\boldsymbol{s}_2, \boldsymbol{s}_{02})$ are reciprocal to each other, if and only if, they satisfy the following equation:

$$\boldsymbol{s}_1 \cdot \boldsymbol{s}_{02} + \boldsymbol{s}_2 \cdot \boldsymbol{s}_{01} = 0 \tag{A.3}$$

- **Reciprocal function (**Reciprocal(***S***)**):** The reciprocal function returns a screw matrix which includes all the reciprocal screws to the screws contained in ***S***. It can be obtained by exchanging the first three columns and the last three columns of the null space of ***S***.

- **Union of Screw matrices:** This represents the sum of the screw spaces. It can be obtained by simply stacking them on top of each other as shown in Equation (A.4).

$$\bigcup_{i=1}^{n} \boldsymbol{S}_i = \begin{pmatrix} \boldsymbol{S}_1 \\ \boldsymbol{S}_2 \\ \vdots \\ \boldsymbol{S}_n \end{pmatrix} \tag{A.4}$$

- **Intersection of Screw matrices:** This represents the set of screws common to all the screw matrices of interest. It can be obtained by applying double reciprocals as shown in Equation (A.5).

$$\bigcap_{i=1}^{n} \boldsymbol{S}_i = \text{reciprocal}\left(\bigcup_{i=1}^{n} \text{reciprocal}\left(\boldsymbol{S}_i\right)\right) \tag{A.5}$$

Appendix B: Non Dominated Sorted Genetic Algorithm (NSGA-II)

This appendix describes the definitions related to multi objective optimization problems and the operators used in the NSGA-II code used in this dissertation. The codes and terminologies given below are summarized from [34] and [35].

B.1 Multi objective optimization Definitions

The General Multi Objective optimization Problem:

Find the vector $\mathbf{x}^* = \left[x_1^*, x_2^*, \cdots, x_n^*\right]^T$ which satisfies

$$g_i\left(\mathbf{x}\right) \geq 0,\ i = 1, 2, ..., m$$

$$h_i\left(\mathbf{x}\right) = 0,\ i = 1, 2, ..., p$$

And optimizes the vector function

$$\mathbf{f}\left(\mathbf{x}\right) = \left[\, f_1\left(\mathbf{x}\right),\, f_2\left(\mathbf{x}\right), \cdots, f_k\left(\mathbf{x}\right)\right]^T$$

Pareto Optimality:

A point $\mathbf{x}^* \in \Omega$ is Pareto optimal if for every $\mathbf{x}^* \in \Omega$ and $I = \{1,2,\ldots,k\}$ either,

$$\forall_{i \in I}\left(f_i\left(\mathbf{x}\right) = f_i\left(\mathbf{x}^*\right)\right)$$

Or there is at least one $i \in I$ such that

$$f_i(\mathbf{x}) > f_i(\mathbf{x}^*)$$

Pareto dominance:

A vector $\mathbf{u} = [u_1, u_2, \cdots, u_k]^T$ dominates a vector $\mathbf{v} = [v_1, v_2, \cdots, v_k]^T$ (*denoted by* $\mathbf{u} \preceq \mathbf{v}$) if and only if *u* is partially less than *v*; i.e.:

$$\forall i \in \{1, 2, ..., k\}, u_i \le v_i \wedge \exists i \in \{1, 2, ..., k\} : u_i < v_i$$

Pareto Optimal Set ($\mathcal{P}^*$):

$$\mathcal{P}^* := \{x \in \Omega \mid \neg\exists x' \in \Omega \;\; \mathbf{f}(x') \preceq \mathbf{f}(x)\}$$

Pareto Front ($\mathcal{PF}^*$):

$$\mathcal{PF}^* := \{\mathbf{u} = \mathbf{f} = (f_1(x), f_2(x), ..., f_k(x)) \mid x \in \mathcal{P}^*\}$$

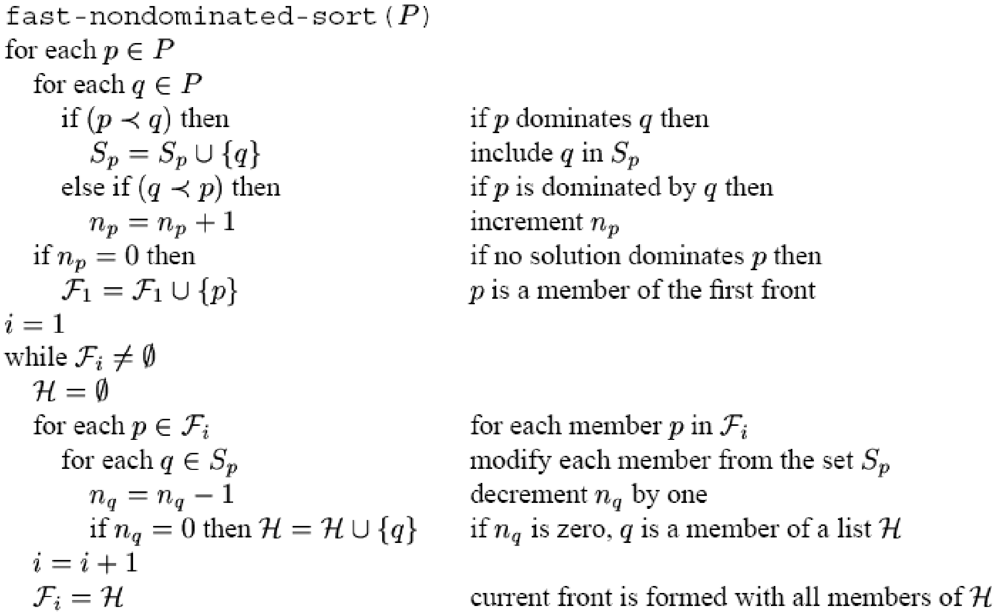
B.2 NSGA-II

The following sub-sections describe the operators used in the NSGA-II code used in this dissertation.

B.2.1 Fast non-dominating sorting approach

The algorithm described below has requires $O(mN^2)$ computations at the most. This requires $O(mN)$ comparisons, where m is the number of objectives.

First, for each solution two entities are calculated: (i) n_i, the number of solutions which dominate the solution *i*, and (ii) S_i, a set of solutions which the solution i dominates. These two entities require $O(mN^2)$ comparisons to calculate. All points which have n_i=0 are identified and saved in a list $\mathcal{F}_1$, called the current front. For each solution in the current front we visit each member (*j*) in its set S_i and reduce its n_j count by one. If for any member *i* the count becomes zero, we put it in a separate list $\mathcal{H}$. When all members of the current front have been checked, we declare the members in $\mathcal{F}_1$ as members of the first front. We then continue this process using the newly identified front $\mathcal{H}$ as our current front. Each such iteration requires $O(N)$ computations. This process continues till all the fronts are identified. The worst case complexity of this loop is $O(N^2)$. Thus, the overall complexity of the algorithm $O(mN^2) + O(N^2)$ or $O(mN^2)$

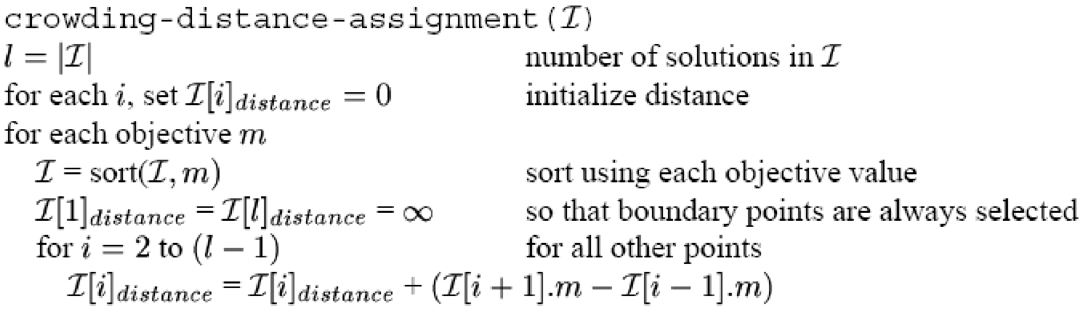


```
fast-nondominated-sort($P$)
for each $p \in P$
  for each $q \in P$
    if $(p \prec q)$ then                        if $p$ dominates $q$ then
      $S_p = S_p \cup \{q\}$                     include $q$ in $S_p$
    else if $(q \prec p)$ then                   if $p$ is dominated by $q$ then
      $n_p = n_p + 1$                            increment $n_p$
  if $n_p = 0$ then                              if no solution dominates $p$ then
    $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$   $p$ is a member of the first front
$i = 1$
while $\mathcal{F}_i \neq \emptyset$
  $\mathcal{H} = \emptyset$
  for each $p \in \mathcal{F}_i$                 for each member $p$ in $\mathcal{F}_i$
    for each $q \in S_p$                         modify each member from the set $S_p$
      $n_q = n_q - 1$                            decrement $n_q$ by one
      if $n_q = 0$ then $\mathcal{H} = \mathcal{H} \cup \{q\}$   if $n_q$ is zero, $q$ is a member of a list $\mathcal{H}$
  $i = i + 1$
  $\mathcal{F}_i = \mathcal{H}$                  current front is formed with all members of $\mathcal{H}$
```

Figure B.1: Pseudo code for non dominated sorting.

B.2.2 Density estimation

The pseudo code in Figure B.1 is to estimate the density of the solutions surrounding a particular point. This is done by measuring the *crowding distance*, which serves as an estimate of the size of the largest cuboid enclosing a point I without including any other point, as shown in Figure B.2.



```
crowding-distance-assignment($\mathcal{I}$)
$l = |\mathcal{I}|$                                          number of solutions in $\mathcal{I}$
for each $i$, set $\mathcal{I}[i]_{distance} = 0$            initialize distance
for each objective $m$
   $\mathcal{I} = \text{sort}(\mathcal{I}, m)$               sort using each objective value
   $\mathcal{I}[1]_{distance} = \mathcal{I}[l]_{distance} = \infty$     so that boundary points are always selected
   for $i = 2$ to $(l - 1)$                                  for all other points
      $\mathcal{I}[i]_{distance} = \mathcal{I}[i]_{distance} + (\mathcal{I}[i+1].m - \mathcal{I}[i-1].m)$
```

Figure B.2: Pseudo code for measuring the crowding distance.

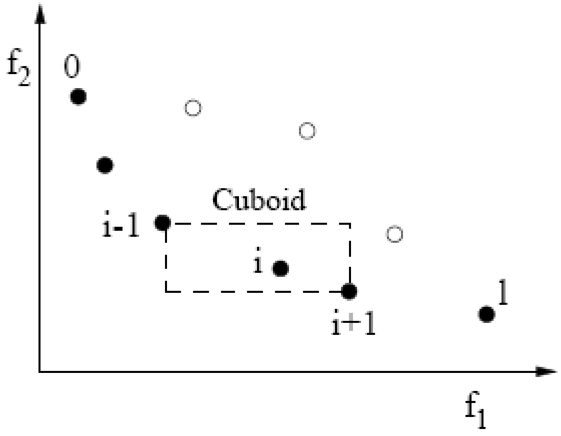


Figure B.3: Crowding distance estimation.

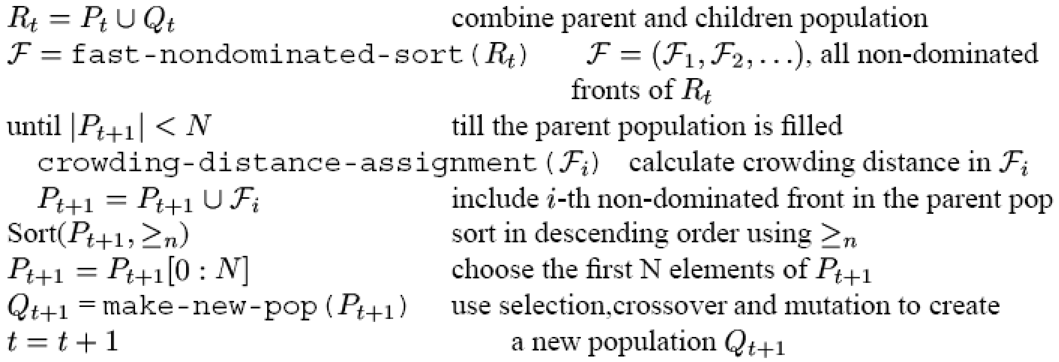
The crowding distance is then used to help spread out the Pareto front by introducing the crowding comparison operator ($\geq_n$), which works as shown below:

Given the non dominated rank (i_{rank}) and the local crowding distance, ($i_{distance}$), $\geq_n$ is defined as:

$$i \geq_n j \quad \text{if } (i_{rank} < j_{rank}) \text{ or } ((i_{rank} = j_{rank}) \text{ and } (i_{distance} < j_{distance}))$$

B.2.3 The Main NSGA-II loop

The main NSGA-II loop is given in the pseudo code in Figure B.4



```
$R_t = P_t \cup Q_t$                                  combine parent and children population
$\mathcal{F}$ = fast-nondominated-sort($R_t$)          $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$, all non-dominated
                                                          fronts of $R_t$
until $|P_{t+1}| < N$                                 till the parent population is filled
    crowding-distance-assignment($\mathcal{F}_i$)      calculate crowding distance in $\mathcal{F}_i$
    $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$            include $i$-th non-dominated front in the parent pop
Sort($P_{t+1}, \geq_n$)                               sort in descending order using $\geq_n$
$P_{t+1} = P_{t+1}[0 : N]$                            choose the first N elements of $P_{t+1}$
$Q_{t+1}$ = make-new-pop($P_{t+1}$)                   use selection,crossover and mutation to create
$t = t + 1$                                               a new population $Q_{t+1}$
```

Figure B.4: Main NSGA-II loop pseudo code.

The design and use joints that can disengage with minimum labor, part damage, and material contamination are critical to ensure effective service, part reuse, and material recycling. In this dissertation, we proposed the concept of heat-reversible locator-snap systems as joints that can satisfy the aforementioned requirements. The assembly process is analogous to regular locator-snap system. During disassembly, snaps can be released by the application of localized heat at specific locations. The in-plane thermal expansion constrained by locators and temperature gradient along the wall thickness are exploited to realize the out-of-plane bulging of the enclosure wall that releases the snaps.

Appendix C: Parallel Load Balancing Source Code

Initialization of GA Code on all processors

```
int main(int argc, char *argv[])
{
    MPI_Init(&argc, &argv);
    MPI_Status stat;
    MPI_Comm_size(MPI_COMM_WORLD, &nproc);
    MPI_Comm_rank(MPI_COMM_WORLD, &myid);
    if (myid == 0){ //main processor (manager)
        double time1 = MPI_Wtime();
        //run the GA code
        double time2 = MPI_Wtime();
             //send message to other proc to terminate
        double myflag = 2e299; //>INFINITY
        for (int myi = 1; myi<nproc; myi++)
                    MPI_Send(&myflag, 1, MPI_DOUBLE, myi, myi, MPI_COMM_WORLD);
        cout<< "elapsed time : "<< time2-time1 <<endl;  }
    else{   //all other processors (Workers)
        int myflag = 0;
        const int noft = nobjfn + nconst + 1;
        const int npop_P_FS = npop+5; //pop. can be higher than npop
        const int invars = indvars + inivars;//all variables
        double ad_my_vars[invars*npop_P_FS];
        int ai_my_int_vars[inivars];
        double ad_fvals[noft*npop_P_FS];
        while (myflag!=1){
            MPI_Recv(ad_my_vars, invars*npop_P_FS, MPI_DOUBLE, 0, myid,
            MPI_COMM_WORLD, &stat);
            if (ad_my_vars[0] > 1.0e299)
                myflag = 1;
            else{
                MPI_Get_count(&stat, MPI_DOUBLE, &my_ncnt);
                my_ncnt = my_ncnt/invars;
                for (int i = 0; i<my_ncnt; i++) {
                    for (int j=0; j<inivars; j++)
                        ai_my_int_vars[j]=round(ad_my_vars[i*invars+indvars+j]);
                        GAof(&ad_fvals[i*noft], &ad_fvals[i*noft+nobjfn],
                        ai_my_int_vars,&ad_my_vars[i*invars], (void *)
                        &spass_data,&ad_fvals[i*noft+nobjfn+nconst]);
                }
                MPI_Send(ad_fvals,my_ncnt,*noft,MPI_DOUBLE,0,myid,
                MPI_COMM_WORLD);
            }
        }
    }
    return 0;
}
```

Time function to select the desired parallelization scheme

```
int timepln(double *tmeanstd)
{
    double tmean = tmeanstd[0], tstd = tmeanstd[1];
    if (tmean < 2.5e-3)
       return 2;
    else
       return 3;
}
```

Define Scheme 1, divide population evenly and manager will do work

```
proc_share = floor(double(my_count)/double(nproc));
my_ncnts = proc_share*inv;
my_ncntr = proc_share*inof_t;
for (int my_i= 1; my_i<(nproc-1); my_i++)
    {
        my_strts = my_i*proc_share*inv;
        MPI_Isend(&my_vars[my_strts], my_ncnts, MPI_DOUBLE, my_i, my_i,
        MPI_COMM_WORLD,&reqs[my_i-1]);
        my_strtr = my_i*proc_share*inof_t;
        MPI_Irecv(&my_of_vals[my_strtr], my_ncntr, MPI_DOUBLE, my_i, my_i,
        MPI_COMM_WORLD, &reqr[my_i-1]);
    }
//send rest to last proc
my_strts = (nproc-1)*proc_share*inv;
my_ncnts = my_nvars - my_strts;
MPI_Isend(&my_vars[my_strts], my_ncnts, MPI_DOUBLE, nproc-1, nproc-1,
MPI_COMM_WORLD, &reqs[nproc-2]);
my_strtr = (nproc-1)*proc_share*inof_t;
my_ncntr = my_nofvals - my_strtr;
MPI_Irecv(&my_of_vals[my_strtr], my_ncntr, MPI_DOUBLE, nproc-1, nproc-1,
MPI_COMM_WORLD, &reqr[nproc-2]);
//manager does work
for (int my_k=0; my_k<proc_share; my_k++)
{
    for (int j=0; j<iniv; j++)
        ai_my_int_vars[j]=round(my_vars[my_k*inv+indv+j]);
    gendat.pfof(&my_of_vals[inof_t*my_k],&my_of_vals[inof_t*my_k+inof],
    ai_my_int_vars,&my_vars[inv*my_k],gendat.pvpar,&my_of_vals[inof_t*my_k+inof+
    incon]);
}
//wait to all processors to finish
MPI_Waitall(nproc-1, reqr, stat);
//update population
for(int my_j=0; my_j <my_count; my_j++)
{
    for (int j=0; j<iniv; j++)
        ai_my_int_vars[j]=round(my_vars[my_j*inv+indv+j]);
    pnewpop->repmember(&my_of_vals[my_j*inof_t],&my_of_vals[my_j*inof_t+inof],
    ai_my_int_vars, &my_vars[my_j*inv]);
    mytim3[my_j] = my_of_vals[my_j*inof_t+inof+incon];
}
//find the times for finished generation to decide the next generation scheme
timefun(mytim3, my_count, tmeanstd);
```

Define Scheme 3, divide population partially, X=80%

```
double per = 0.8;       // X
proc_share = floor(double(my_count)*per/double(nproc) + 0.5);
my_ncnts = proc_share*inv;
my_ncntr = proc_share*inof_t;
for (int my_i= 1; my_i<nproc; my_i++)
{
    my_strts = (my_i-1)*proc_share*inv;
    MPI_Isend(&my_vars[my_strts], my_ncnts, MPI_DOUBLE, my_i, my_i,
    MPI_COMM_WORLD,&reqs[my_i-1]);
    my_strtr = (my_i-1)*proc_share*inof_t;
    MPI_Irecv(&my_of_vals[my_strtr], my_ncntr, MPI_DOUBLE, my_i, my_i,
    MPI_COMM_WORLD, &reqr[my_i-1]);
}
int mngstrt = proc_share*(nproc-1);
int myind, myflag=1;
while (mngstrt < my_count)
{// test to see if any proc finished
   MPI_Testany(nproc-1, reqr, &myind, &myflag, stat);
   if (myflag)
    {//a proc finished, get it's OF values. divide the remaining //chromosomes
    and send them to that proc
        proc_share = ceil(double(my_count-mngstrt)*per/double(nproc));
        my_ncnts = proc_share*inv;
        my_ncntr = proc_share*inof_t;
        my_strts = mngstrt*inv;
        MPI_Isend(&my_vars[my_strts], my_ncnts, MPI_DOUBLE, myind+1, myind+1,
        MPI_COMM_WORLD,&reqs[myind]);
        my_strtr = mngstrt*inof_t;
        MPI_Irecv(&my_of_vals[my_strtr], my_ncntr, MPI_DOUBLE, myind+1, myind+1,
        MPI_COMM_WORLD, &reqr[myind]);
        mngstrt+= proc_share;
    }
    else
    {//manager does work instead of remaining waiting
        for (int j=0; j<iniv; j++)
            ai_my_int_vars[j]=round(my_vars[inv*mngstrt+indv+j]);
        gendat.pfof(&my_of_vals[inof_t*mngstrt],
        &my_of_vals[inof_t*mngstrt+inof],ai_my_int_vars,&my_vars[inv*mngstrt],
        gendat.pvpar,&my_of_vals[inof_t*mngstrt+inof+incon]);
        mngstrt++;//update the no of finished/sent chromosomes
   }
}
//wait to all processors to finish
MPI_Waitall(nproc-1, reqr, stat);
//update population
for(int my_j=0; my_j <my_count; my_j++)
{
    for (int j=0; j<iniv; j++)
        ai_my_int_vars[j]=round(my_vars[my_j*inv+indv+j]);
    pnewpop->repmember(&my_of_vals[my_j*inof_t],&my_of_vals[my_j*inof_t+inof],
    ai_my_int_vars, &my_vars[my_j*inv]);
    mytim3[my_j] = my_of_vals[my_j*inof_t+inof+incon];
}
//find the times for finished generation to decide the next generation scheme
timefun(mytim3, my_count, tmeanstd);
```

References

[1] Zhang, H., Kuo T., Lu. H, Huang, S., 1997, "Environmentally Conscious Design and Manufacturing: A State-of-the-Art Survey," *Journal of Manufacturing Systems*, Vol. 18, No. 5, pp. 352-371.

[2] Basel Action Network, 2005, "The digital dump: exporting re-use and abuse to Africa," http://www.computertakeback.com/docUploads/TheDigitalDumpWeb.pdf

[3] European Union directive on Waste Electrical and Electronic Equipment, http://www.berr.gov.uk/files/file41319.pdf

[4] Chen, R., Navin-Chandra, D. and Prinz, F., 1993, "Product design for Recyclability: A cost benefit analysis model and its applications," IEEE international symposium on electronics and the environment, Arlington, VA, pp. 178-183.

[5] Rios, P., Stuart, J. and Grant, E., 2003, "Plastics disassembly verses bulk recycling: engineering design for end-of-life electronics resource recovery," Environmental Science and Technology, Vol. 37, no. 23, pp 5463-5470.

[6] The Aluminum Company of America, http://www.alcoa.com/rigid_packaging/en/docs/quality_rating_procedures.doc

[7] Kang, H-Y. and Schoenung, J., 2005, "Eclectronic waste recycling: a review of US infrastructure and technology options," Resources, Conservation and Recycling, Vol. 45, pp 368-400.

[8] Sodhi, R., Sonnenberg, M. and Das, S., 2004, "Evaluating the unfastening effort in design for disassembly and serviceability," *Journal of Engineering Design*, Vol. 15, No. 1, pp. 69-90.

[9] Audi world, http://www.audiworld.com

[10] Ashley, S., 1994, "Aluminum vehicle breaks new ground," *Mechanical Engineering*, Vol. 116, No. 116, pp. 50-51.

[11] Irving, B., 1998, "Interest in welded Aluminum automobiles gathers momentum worldwide," *Welding Journal*, Vol. 77, No. 6, pp. 31-35.

[12] Graedel, T. and Klee, R., 2002, “Getting serious about sustainability,” *Environmental Science and Technology*, Vol. 36, No. 4, pp. 523-529.

[13] “DIRECTIVE 2000/53/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 18 September2000 on end-of life vehicles,” *Official Journal of the European Communities*, September 2000.

[14] Ferrão, P. and Amaral, J., 2006, “Design for recycling in the automobile industry: new approaches and new tools,” *Journal of Engineering Design*, Vol. 17, No. 5, pp. 447-462.

[15] Ogushi, Y. and Kandlikar, M., 2006, “The impact of End-of-life vehicle recycling law on automobile recovery in Japan,” Proceedings - Fourth International Symposium on Environmentally Conscious Design and Inverse Manufacturing, Eco Design 2005.

[16] Boothroyd, G. and Dewhurst, P., 1983. *Design for Assembly Handbook*, University of Massachusetts, Amherst.

[17] Boothroyd, G. and Alting, L., 1992, “Design for Assembly and Disassembly,” Annals of the CIRP, Vol. 41, No. 2, pp 625-636.

[18] Ishii, K., Eubanks, C., and DiMarco, P, 1994, “Design for Product Retirement and Material Life-cycle,” *Materials and Design*, Vol. 15, pp. 225-233.

[19] Ishii, K. and Lee, B., 1996, “Reverse Fishbone Diagram: A Tool in Aid of Design for Product Retirement,” Proceedings of *the ASME Design Technical Conference*, 96-DETC/DFM-1272.

[20] Lee, K. and Gadh, R., 1996, “Computer Aided Design for Disassembly: A Destructive Approach,” Proceedings of *the IEEE International Symposium on Electronics and Environment*, pp. 173-178.

[21] Harjula, T., Rapoza, B., Knight, W. and Boothroyd, G., 1996, “Design for Disassembly and the Environment,” *Annals of the CIRP*, Vol. 45, No. 1 pp. 109-114.

[22] Zhang, H., Kuo, T., Lu, H. and Huang S., 1997, “Environmentally Conscious Design and Manufacturing: A state-of-the art survey,” Journal of Manufacturing Systems, Vol. 18, No. 5, pp. 352-371.

[23] Viswanathan, S. and Allada, V., 1999, “Value-Based Product Structure Evaluation for Disassembly,” Proceedings of *the EcoDesign ’99: First International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, pp. 778-783.

[24] Koch, P., "Snap fit Design," The Pennsylvania State University, Eire, PA. http://engr.bd.psu.edu/pkoch/plasticdesign/snap_design.htm

[25] Jovane, F., Alting, L., Armillotta, A., Eversheim, W., Feldmann, K. and Seliger, G., 1993, "A key issue in product life cycle: disassembly," *Annals of the CIRP*, Vol. 42, pp. 651-658.

[26] Keoleian, G. and Menerey, D., 1994, "Sustainable development by design: review of life cycle design and other approaches," *Journal of the Air & Waste Management Association*, Vol. 44, pp. 645-668.

[27] Gungor, A. and Gupta, S., 1999, "Issues in environmentally conscious manufacturing and product recovery: a survey," *Computers and Industrial Engineering*, Vol. 36, pp. 811-853.

[28] Shetty, D., Rawolle, K. and Campana, C., 2000, "A New Methodology for Ease-of-Disassembly in Product Design," *Recent Advances in Design for Manufacture (DFM)*, Vol. 109, pp. 39-50.

[29] Desai, A. and Mital, A., 2003, "Review of literature on disassembly algorithms and design for disassembly guidelines for product design," *International Journal of Industrial Engineering*, Vol. 10, No. 3, pp. 244-255.

[30] Kuo, T., 2000, "Disassembly sequence and cost analysis for electromechanical products," *Robotics and computer integrated manufacturing*, Vol. 16, pp. 43-54.

[31] Chow, W., 1978, *Cost Reduction in Product Design*, Van Nostrand Reinhold, NY.

[32] Boothroyd, G., Dewhurst, P. and Knight, W., 1994, *Product Design for Manufacture and Assembly*, Dekker, NY.

[33] Staller, N. and Petrie, S., 2006, "The True Cost of Snaps," Assembly Magazine, http://www.assemblymag.com/CDA/Archives/3eb44059616c9010VgnVCM100000f932a8c0____

[34] Deb, K., 2001, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley and Sons, West Sussex, England.

[35] Coello, C., Veldhuizen, D. and Lmont, G., 2002, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers.

[36] Dewhurst, P., 1993, "Product Design for Manufacture: Design for Disassembly," Industrial Engineering, Vol. 25, No. 9, pp. 26-28.

[37] Lynch, T., 1995, "Design's Next Step: Serviceability," Design News, Vol. 6, No. 26, pp. 79-82.

[38] Seaver, W., 1994, “Design Consideration for Remanufacturability, Recyclability and Reusability of User Interface Modules,” IEEE International Symposium on Electronics and the Environment, pp. 241-245.

[39] Kirby, R. and Wadehra, I., 1993, “Designing Business Machines for Disassembly and Recycling,” IEEE International Symposium on Electronics and the Environment, pp. 32-36.

[40] Tavakoli, M., Mariappan, J. and Huang, J., 2003, “Design for Assembly verses Design for Disassembly - a comparison of guidelines,” Proceedings of the 2003 *ASME International Mechanical Engineering Congress & Exposition*, November 16-21, Washington, D.C., IMECE2003-43951.

[41] Masanet, E., Auer, R., Tsuda, D., Barillot, T. and Baynes, A., 2002, “An assessment and prioritization of "design for recycling" guidelines for plastic components,” IEEE International Symposium on Electronics and the Environment, pp. 5-10.

[42] Reap, J. and Bras, B., 2002, “Design for Disassembly and the Value of Robotic Semi-Destructive Disassembly,” Proceedings of *the ASME 2002 Design Engineering Technical Conferences*, Montreal, Canada, September 29-October 2.

[43] Knight, W. and Boothroyd, G., 1997, “Analysis of Products for End-Of-Life Management,” Professional Program Proceedings, Electronics Industries Forum of New England, May 6-8, pp. 9-28.

[44] Das, S., Mani, V., Caudill, R. and Limaye, K., 2002, “Strategies and Economics in the Disassembly of Personal Computers - A Case Study,” IEEE International Symposium on Electronics and the Environment, May 6-9, San Francisco, CA, pp. 257-262

[45] Johnson, M. and Wang, M., 1994, “Planning Product Disassembly for Material Recovery Opportunities,” *International Journal of Product Research*, Vol. 33, No. 11, pp. 3119-3142

[46] Viswanathan, S. and Allada, V., 2001, “Configuration analysis to support product redesign for end-of-life disassembly,” *International Journal of Production Research*, Vol. 39, No. 8, pp. 1733-1753

[47] McGlothlin, S. and Kroll, E., 1995, “Systematic estimation of disassembly difficulties: Application to computer monitors,” IEEE International Symposium on Electronics and the Environment, May 1-3, Orlando, Fl, pp. 83-88.

[48] E. Kroll, B. Beardsley, and A. Parulian, 1996, “A Methodology to Evaluate Ease of Disassembly for Product Recycling,” *IIE Transactions*, Vol. 28, No. 10, pp. 837-845.

[49] Das, S., Yedlarajiah, P. and Narendra, R., 2000, "An Approach for Estimating the End-Of-Life Product Disassembly Effort and Cost," *International Journal of Production Research,* Vol. 38, No. 3, pp. 657-673.

[50] Hulla, A., Jalali, K., Hamza, K., Skerlos, S., Saitou, K., 2003, "Multi-criteria Decision Making for Optimization of Product Disassembly under Multiple Situations," *Environmental Science and Technology, Special issue on Principles of Green Engineering,* Vol. 37 No. 23, pp. 5303-5313.

[51] Desai, A. and Mital, A., 2003, "Evaluation of disassemblability to enable design for disassembly in mass production," *International Journal of Industrial Ergonomics*, Vol. 32, No. 4, pp. 265-281.

[52] Desai, A. and Mital, A., 2005, "Incorporating work factors in design for disassembly in product design," *Journal of Manufacturing Technology Management*, Vol. 16, No. 7, pp. 712-32.

[53] O'Shea, B., Kaebernick, H., Grewal, S., Perlewitz, H., Müller, K. and Seliger, G., 1999, "Method for Automatic Tool Selection for Disassembly Planning," *Assembly Automation,* Vol. 19, No. 1, pp. 47-54.

[54] Masui K., Mizuhara, K. Ishii, K. and Rose, C., 1999, "Development of Products Embedded Disassembly Process Based on End-of-Life Strategies," Proceedings of the EcoDesign '99: *First International Symposium on Environmentally Conscious Design and Inverse Manufacturing,* pp. 570-575.

[55] Takeuchi, S. and Saitou, K., 2005, "Design for Product Embedded Disassembly Sequence," Proceedings of the 6th IEEE International Symposium on Assembly and Task Planning, Montréal, Canada, July 19-21.

[56] Takeuchi, S. and Saitou, K., 2005, "Design for Multiple Product-Embedded Disassembly Pathways" Proceedings of the IEEE Conference on Automation Science and Engineering, Edmonton, Canada, August 1-2.

[57] Takeuchi, S. and Saitou, K., 2005, "Design for Product-Embedded Disassembly," Proceedings of the 2005 ASME Design Engineering Technical Conferences, Long Beach, California, September 24-28, DETC2005-85260.

[58] Takeuchi, S. and Saitou, K., "Design for Product-Embedded Disassembly with Maximum Profit," Proceedings of the EcoDesign 2005: 4th International Symposium on Environmentally Conscious Design and Inverse Manufacturing, Tokyo, Japan, December 12–14.

[59] Takeuchi, S. and Saitou, K., 2006, "Design for Optimal End-of-Life Scenario via Product-embedded Disassembly," LCE2006: The 13th CIRP International Conference on Life Cycle Engineering, May 31-June 2, Leuven, Belgium, pp. 423-428.

[60] Takeuchi, S. and Saitou, K., 2006, "Design for Optimal End-Of-Life Scenario via Product Embedded Disassembly," Proceedings of the 2006 ASME Design Engineering Technical Conferences, September 10-13, Philadelphia, Pennsylvania, DETC2006-99475.

[61] Chiodo, J., Billett, E. and Harrison D., 1999, "Preliminary Investigation of Active Disassembly Using Shape Memory Polymers," Proceedings of the Eco-Design '99: First International Symposium on Environmentally Conscious Design and Inverse Manufacturing, pp. 590-596.

[62] Chiodo, J., Billett, E. and Harrison, D., 1999, "Active Disassembly Using Shape Memory Polymers for the Mobile Industry," IEEE International Symposium on Electronics and the Environment, Danvers, Massachusetts, May 11-13, pp. 151-156.

[63] Chiodo, J., McLaren, J., Billett, E. and Harrison, D., 2000, "Isolating LCD's at End-of-Life using Active Disassembly Technology: A Feasibility Study," The proceedings of the IEEE International Symposium on Electronics and the Environment, May 8-10, San Francisco, CA, pp. 318-323.

[64] Chiodo, J., Harrison, D. and Billett, E., 2002, "An initial investigation into active disassembly using shape memory polymers," *Proceedings of the IMECHE Part B Journal of Engineering Manufacture*, Vol.215, pp.733-741.

[65] Chiodo, J., Jones, N., Billett, E. and Harrison, D., 2002, "Shape memory alloy actuators for active disassembly using 'smart' materials of consumer electronic products," *Materials and Design*, Vol. 23, pp. 471-478.

[66] Arnaiz S., Bodenhoefer K., Constantin H., Hussein H., Irasarri L., Schnecke D., et al., 2002, "Active disassembly using smart materials (ADSM), a status report of the ongoing EU project," Proceedings of Care Innovation, Vienna; 2002.

[67] Jones, N., Harrison, D., Hussein, H., Billett, E. and Chiodo, J., 2004, "Electrically self-powered active disassembly," Proceedings of the Institution of Mechanical Engineers, Part B (Journal of Engineering Manufacture), Vol. 218, No. B7, 2004, pp 689-697.

[68] Koyu, S., Hideo, O., Masanobu, T. and Takeo, Y., 2003, "Study of auto-disassembly system using shape memory materials," 3rd International Symposium on Environmentally Conscious Design and Inverse Manufacturing - EcoDesign'03, pp 504-509.

[69] Sharp Technical Report, 2004, "Social and Environmental Activities, Developing-easy technologies," http://sharp-world.com/corporate/eco/technology/parts.html

[70] Li, Y., Saitou, K., Kikuchi N., Skerlos, S., and Papalambros, P., 2001, "Design of Heat-Activated Reversible Integral Attachments for Product-Embedded

Disassembly," Proceedings of the EcoDesign 2001: 2nd International Symposium on Environmentally Conscious Design and Inverse Manufacturing, Tokyo, Japan, December 12–15, p. 360–365.

[71] Bendsøe, M. and Kikuchi, N., 1988, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, No. 2, pp. 197-224.

[72] Howell, L., 2001, *Compliant Mechanisms*, John Wiley & Sons, Inc.

[73] Li, Y., Saitou, K., Kikuchi N., 2002, "Design of Heat-Activated Compliant Mechanisms for Product-Embedded Disassembly," Proceedings of the Fifth World Congress on Computational Mechanics, Vienna, Austria, July 7-12.

[74] Li, Y., Saitou, K., and Kikuchi, N., 2003, "Design of Heat-Activated Reversible Integral Attachments for Product-Embedded Disassembly," *International Journal of CAD/CAM*, Vol. 3, pp. 26-40.

[75] Willems, B., Dewulf, W. and Duflou, J., 2005, "Design for active disassembly (DfAD): an outline for future research," Proceedings of the 2005 IEEE International Symposium on Electronics and the Environment, New Orleans, LA, May 16-19, pp. 129-134.

[76] Suri, G.. and Luscher, A., 2000, "Structural Abstraction in Snap-fit Analysis," *ASME Journal of Mechanical Design*, Vol. 122, pp. 295-402.

[77] Nichols, D. and Luscher, A., 1999, "Generation of Design Data through Numerical Modeling of a Post and Dome Feature," Proceedings of the 1999 ASME Design Engineering Technical Conferences, Las Vegas, Nevada, September 12-15, DETC1999/DAC-8596.

[78] Turnbull, V., 1984, "Design Considerations for Cantilever Snap-Fit Latches in Thermoplastics," Proceedings of the Winter Annual Meeting of ASME, 84-WA/Mats-28, pp. 1-8.

[79] Wang, L., Gabriele, G. and Luscher, A., 1995, "Failure Analysis of a Bayonet-Finger Snap-Fit," Proceedings of the ANTEC '95, pp. 3799-3803.

[80] Larsen G. and Larson, R., 1994, "Parametric Finite-Element Analysis of U-Shaped Snap-Fits," Proceedings of the ANTEC '94, pp. 3081-3084.

[81] Genc, S., Messler, R., Bonenberger, P. and Gabriele, G., 1997, "Enumeration of Possible Design Options for Integral Attachment Using a Hierarchical Classification Scheme," *ASME Journal of Mechanical Design*, Vol. 119, pp. 178-184.

[82] Genc, S. Messler Jr., R. and Gabriele, G., 1998, "A Systematic Approach to Integral Snap-Fit Attachment Design," *Research in Engineering Design*, Vol. 10, pp. 84-93.

[83] Genc, S. Messler Jr., R. and Gabriele, G., 1998, "A Hierarchical Classification Scheme to Define and Order the Design Space for Integral Snap-Fit Assembly," *Research in Engineering Design*, Vol. 10, pp. 94-106.

[84] Bonenberger, Paul R., 2000, *The First Snap- fit Handbook, Creating Attachments for Plastic Parts*, Hanser Gardner Publication, Inc., Cincinnati.

[85] Luscher, A., Suri G. and Bodmann, D., 1998, "Enumeration of Snap-Fit Assembly Motions," Proceedings of ANTEC '98, pp. 2677-2681.

[86] Messler, R., Genc, S. and Gabriele, G., 1997, "Integral attachment using snap-fit features: a key to assembly automation. Part 1 – introduction to integral attachment using snap-fit features," *Assembly Automation*, Vol. 17, No. 2, pp. 143-155.

[87] Messler, R., Genc, S. and Gabriele, G., 1997, "Integral attachment using snap-fit features: a key to assembly automation. Part 2 – bringing order to integral attachment: attachment-level design," *Assembly Automation*, Vol. 17, No. 2, pp. 156-165.

[88] Messler, R., Genc, S. and Gabriele, G., 1997, "Integral attachment using snap-fit features: a key to assembly automation. Part 3 – an attachment-level design methodology," *Assembly Automation*, Vol. 17, No. 3, pp. 239-248.

[89] Messler, R., Genc, S. and Gabriele, G., 1997, "Integral attachment using snap-fit features: a key to assembly automation. Part 4 – selection of locking features," *Assembly Automation*, Vol. 17, No. 4, pp. 315-328.

[90] Genc, S., Messler, R. and Gabriele, G., 1998, "Integral attachment using snap-fit features: a key to assembly automation. Part 5 – A procedure to constrain parts fully and generate alternative attachment concepts," *Assembly Automation*, Vol. 18, No. 1, pp. 68-74.

[91] Genc, S., Messler, R. and Gabriele, G., 1998, "Integral attachment using snap-fit features: A key to assembly automation. Part 6 – evaluating alternatives for design optimization," *Assembly Automation*, Vol. 18, No. 2, pp. 153-165.

[92] Genc, S., Messler, R. and Gabriele, G., 1998, "Integral attachment using snap-fit features: a key to assembly automation. Part 7 – Testing the conceptual design methodology with a case study," *Assembly Automation*, Vol. 18, No. 3, pp 223-236.

[93] Ball, R. S., 1900, *A Treatise on the Theory of Screws*, 1900, Cambridge University Press, Cambridge, UK.

[94] Waldron, K. J., 1966, “The Constraint Analysis of Mechanisms,” *Journal of Mechanisms*, 1966, Vol. 1, No. 2, 101–114.

[95] Ohwovoriole, M. and Roth, B., 1981, “An extension of screw theory,” *ASME Journal of mechanical Design*, Vol. 103, No. 4, pp. 725-735.

[96] Sugimoto, K. and Duffy, J., 1982, “Application of linear algebra to screw Systems,” *Mechanism and Machine Theory*, Vol. 17, No.1, pp. 73-83.

[97] Konkar, R., and Cutkosky, M., 1995, “Incremental Kinematic Analysis of Mechanisms,” *ASME Journal of Mechanical Design*, Vol. 117, pp. 589–596.

[98] Adams, J. D., and Whitney, D. E., 1999, “Application of Screw Theory to Constraint Analysis of Assemblies of Rigid Parts,” Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning, Porto, Portugal, July 2-24, pp.69–74.

[99] Adams, J. D., and Whitney, D. E., 1999, “Application of Screw Theory to Motion Analysis of Assemblies of Rigid Parts,” Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning, Porto, Portugal, July 21-24, pp.75–80.

[100] Lee, B. and Saitou, K., 2006, “Three-Dimensional Assembly Synthesis for Robust Dimensional Integrity based on Screw Theory,” *ASME Journal of Mechanical Design*, Vol. 128, pp. 243-251.

[101] Holland, J., 1975, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

[102] Branke, J., Schmeck, H., Deb, K. and Reddy, S., 2004, “Parallelizing Multi-Objective Evolutionary Algorithms: Cone Separation,” *KanGAL Report No. 2004017*.

[103] Hiroyasu, T., Miki, M. and Watanbe, S., 2000, “The New Model of Parallel Genetic Algorithm in Multi-Objective Optimization Problems - Divided Range Multi-Objective Genetic Algorithm -,” Proceedings of Congress on Evolutionary Computation, Vol. 1, pp. 333-340.

[104] Quagliarella, D. and Vicini., A., 1998, “Sub-population Policies for a Parallel Multiobjective Genetic Algorithm with Applications to Wing Design,” *IEEE International Conference On Systems, Man, And Cybernetics*, San Diego, California, Vol. 4, pp. 3142-3147.

[105] Stanley, T. and Mudge, T., 1995, “A Parallel Genetic Algorithm for Multiobjective Microprocessor Design,” Proceedings of the Sixth International Conference on Genetic Algorithms.

[106] Okamoto, N. and Zhao, Q., 2002, “Synchronous Parallel GA with Nearly Zero sequential Computations,” Proceedings of the First International Symposium on Cyber Worlds (CW.02), IEEE.

[107] Alba, E. and Troya, J., 2001, “Analyzing synchronous and asynchronous parallel distributed genetic algorithms,” *Future Generation Computer Systems*, Vol. 17, pp. 451–465.

[108] Adeli, H. and Kumar, S., 1995, “Distributed genetic algorithm for structural optimization,” *Journal of Aerospace engineering*, Vol. 8, No. 3, pp. 156-163.

[109] Marco, N. and Lanteri, S., 2000, “A two-level parallelization strategy for Genetic Algorithms applied to optimum shape design,” *Parallel Computing*, Vol. 26, pp. 377-397.

[110] Hong, C., Kim, W. and Kim, Y., 2004, “Distributed Channel Routing Using Genetic Algorithm,” PDCAT 2004, LNCS 3320, pp. 234–237, 2004.

[111] Aguirre, H., Tanaka, K., Sugimura, T. and Oshita, S., 2000, “Improved Distributed Genetic Algorithm with Cooperative-Competitive Genetic Operators,” Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Vol. 5, pp. 3816-3822.

[112] Alba, E. and Troya, J., 2000, “Influence of the Migration Policy in Parallel Distributed GAs with Structured and Panmictic Populations,” *Applied Intelligence* Vol. 12, pp. 163-181.

[113] Cantù-Paz, E. and Goldberg, D., 2000, “Efficient parallel genetic algorithms: theory and practice,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 186 pp. 221-238.

[114] S. C. Liu, S. J. Hu, and T. C. Woo, “Tolerance analysis for sheet metal assemblies,” *ASME Journal of Mechanical Design*, vol. 118, pp. 62-67, 1996.

[115] S. C. Liu and S. J. Hu, “Variation simulation for deformable sheet metal assemblies using finite element methods,” *ASME Journal of Manufacturing Science and Engineering*, Vol. 119, pp. 368-374, 1997.

[116] Y. J. Long and S. J. Hu, “A unified model for variation simulation of sheet metal assemblies”, in *Geometric Design Tolerancing: Theories, Standards and Applications*, H. A. ElMaraghy, Ed. Chapman & Hall, 1998, pp.208-219.

[117] K. G. Merkely, K. W. Chase, and E. Perry, “An introduction to tolerance analysis of flexible assemblies,” Proceedings of the 1996 MSC World Users Conferences.

[118] M. Chang and D. C. Gossard, “Modeling the assembly of compliant, non-ideal parts,” *Computer-Aided Design*, Vol. 29, No. 10, pp. 701-708, 1997.

[119] Lee, B., Shalaby, M., Collins, R., Crisan, V., Walls, S., Robinson, D. and Saitou, K., 2007, “Variation Analysis of Three Dimensional non-rigid Assemblies,” IEEE International Symposium on Assembly and Manufacturing, July 22-25, Ann Arbor, MI.

[120] Pereira, F., Machado, P., Costa, E., and Cardoso, A., 1999, “Graph Based Crossover – A Case Study with the Busy Beaver Problem,” Proceedings of the 1999 Genetic and Evolutionary Computation Conference.

[121] Syswerda G., 1990, “Uniform crossover in genetic algorithms.” Proceedings of 3rd International Conference on Genetic Algorithms, pp. 2–9.

[122] Herrera, F. and Lozano, M., 1996, “Heuristic crossovers for real-coded genetic algorithms based on fuzzy connectives,” Parallel Problem Solving from Nature - PPSN IV. International Conference on Evolutionary Computation. 5th Conference on Parallel Problem Solving from Nature, September 22-26, Berlin, Germany.

[123] Michaelwicz, Z., 1996, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd Edition, Springer-Verlag, Berlin, Heidelberg.

[124] Das, S., 2000, “Life-Cycle Impacts of Aluminum Body-in-White Automotive Material,” *The journal of the Minerals, Metals & Materials Society*, Vol. 52, pp. 41-44.

[125] “Design for aluminum recycling,” 1993, *Automotive Engineering*, Vol. 101, pp. 65-68.

[126] Audi Space Frame (ASF ®) aluminum body structure. http://www.audi.com

[127] Ungureanu, C., Das, S. and Jawahir, I., 2007, “Life-cycle cost analysis: aluminum versus steel in passenger cars,” Aluminum alloys for transportation, packaging, aerospace and other applications, February 25 - March 1, Orlando, Fl., pp. 11-24.

[128] Martchek, K., Fisher, E., Wasson, A., 1996, “A Response to “The Environmental Impact of Steel and Aluminum Body-in-Whites,” *The journal of the Minerals, Metals & Materials Society*, Vol. 48, pp. 40-41.

[129] Kasravi, K. “vehicle body design,” TEC 452, Central Michigan University.

www.kasravi.com/cmu/tec452/BodyEngineering/VibrationNoise.htm

[130] Goldberg, D., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.

[131] Babu, B., Parande, A. and Basha, C., 2007, “Electrical and electronic waste: A global environmental problem,” *Waste Management and Research*, Vol. 25, No. 4, pp. 307-318.

[132] Cantù-Paz, E., 2000, *Efficient and Accurate Parallel Genetic Algorithms*, Kluwer Academic Publishers.

[133] Doerfler, D. and Brightwell, R., 2006, “Measuring MPI send and receive overhead and application availability in high performance network interfaces,” Recent Advances in Parallel Virtual Machine and Message Passing Interface, 13th European PVM/MPI User's Group Meeting. Lecture Notes in Computer Science, Vol. 4192, pp.331-338.

[134] Garner, D., 2006, “Recycling technology: unlocking the riddle of LCD re-use,” University of York Communications Office - University of York - Heslington – YORK.

[135] Cui, J. and Forssberg, E., 2003, “Mechanical recycling of waste electric and electronic equipment: a review,” *Journal of Hazardous Materials* Vol. 99 No. 3, pp. 243–263.

[136] Zhang, S., Forssberg E., van Houwelingen, J., Rem, P. and Wei, L., 2000, “End of life electric and electronic equipment management towards the 21st century,” *Waste Management and Research*, Vol. 18 No. 1 pp. 73-85.

[137] *The California Electronics Recycling Act*, 2003, http://www.ciwmb.ca.gov/electronics/act2003/

[138] “Electronic Hazardous Waste (E-Waste),” *California Department of Toxic Substances Control (DTSC)*, Monday, November 26, 2007. http://www.dtsc.ca.gov/HazardousWaste/EWaste/

[139] Hunt, K., 1978, *Kinematic Geometry of Mechanisms*, Oxford University, Press, New York.

[140] Roth, B., 1984, “Screws, Motors, and Wrenches that can not be Bought in a Hardware Store,” Robotics Research, The first International Symposium, MIT Press, Cambridge, MA, pp. 679–735.