

Product configuration optimization for disassembly planning: A differential approach

Shivakumar Viswanathan^a, Venkat Allada^{b,*}

^aComputer Science Department, Brandeis University, USA

^bSustainable Design Laboratory, 205, Engineering Management Department, University of Missouri-Rolla, Rolla, MO 65409, USA

Received 2 March 2004; accepted 2 December 2004

Available online 5 March 2005

Abstract

Product configuration immensely influences the suitability of a product for end-of-life (EOL) disassembly. The product configuration is the relative spatial and logical arrangement of the different parts/sub-assemblies of the product with respect to each other. The complexity involved in studying the influence of configuration design on EOL disassembly has limited the scope of the current design for disassembly (DfD) approaches to guideline-based prescriptive methods and index-based evaluation techniques. The application of these approaches has primarily been limited to specific case studies of product redesign. Many of the current methods do not provide the necessary rigor that will lead to the creation of a theoretical base for addressing product configuration issues which is indispensable during product redesign. Though fraught with obstacles, studying the effects of product configuration on DfD will be useful to develop automated configuration optimization methods for EOL disassembly. To this end, a model to study the combinatorial configuration design optimization problem from a disassembly perspective is described in this study. The different structural principles of the design space derived in this study provide insights into the possibilities and the natural shortcomings of automated optimization of a product by relating the effects of design constraints and disassembly requirements on product redesign. A hierarchical evolutionary programming based algorithm is also developed to test the design solutions generated by the proposed model.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Disassembly; Product configuration; Optimization; Evolutionary programming

1. Introduction

Increasing concern regarding the environmental effects associated with a product's life cycle has propelled the end-of-life (EOL) disassembly to prominence. The EOL disassembly is a process that could be effectively used to transform products, at the end of their useful life, from the fully assembled state to the part and sub-assembly states required for various reuse and recycling processes [1,2]. However, significant investments of labor and resources associated

with EOL disassembly are major obstacles for its effective integration as viable green product strategy. Design for disassembly (DfD) is a powerful approach for improving the product's EOL 'disassemblability' by appropriate design of the product itself rather than improvement restricted to optimizing the disassembly processes and tasks for a given product. Due to the extent of its influence and the difficulties involved in its implementation, DfD has received significant research attention. Interested readers are referred to any of the several reviews of the research literature in DfD and the related domains of design for recycling and remanufacture [3–8]. This paper deals with a significant issue affecting DfD—the optimization of the product configuration for EOL disassembly.

* Corresponding author. Tel.: +1 573 341 4573;
fax: +1 573 341 6567.

E-mail address: allada@umr.edu (V. Allada).

Nomenclature

ξ	finite set of assemblies (products) with each member, g , having identical disassembly constraints	S_{ovp}	optimal value disassembly sequence in Ω_Q that considers the optimal trade-off between the value extracted and cost of disassembly
τ_g	blocking topology of each member g of ξ	S_{bvp}	best value disassembly sequence in Ω_Q having the lowest petition magnitude
E_g	suitability of a blocking topology τ_g for EOL disassembly expressed as a positive real number	S_{wvp}	disassembly sequence in Ω having the worst value precedence
S	Partial ordered set or poset $= (Q, <)$, where $x < y$ implies that part y is removed prior to part x ($x, y \in$ assembly Q)	ξ	assembly configuration space of products to be disassembled
Ω	disassembly sequence space of assembly Q containing m parts represented as $\{S_1, S_2, S_3, \dots, S_k\}$ where $k = m!$, where all such sequences (feasible and infeasible) are of length m	δ	distance measure defined in the sequence space Ω
Ω_Q	disassembly sequence subspace which is the subset of Ω containing all feasible sequences possible on Q	$d(g_1, g_2)$	distance measure between any two members g_1 and g_2 of ξ
VP	distribution of value associated with a disassembly sequence	V_{max}	Maximum part value of a given assembly
S_{dvp}	disassembly sequence (desired value precedence) where the highest valued parts can be extracted early in the disassembly	V_{min}	Minimum part value of a given assembly
p	petition function	c_j	characteristic coefficient associated with each part of a given assembly
P_S	real numbered petition value that is obtained by the petition function from the VP of a given disassembly sequence S	C	characteristic vector containing a particular set of characteristic coefficients
		r_i	removability value
		K	predefined number of assembly samples that are searched by the disassembiosis strategy
		R	equivalence relation defined on ξ
		k	distance from g that is acceptable as a differential modification
		L_{ij}	state of a cell in the difference array (DA)

1.1. Motivation

The product configuration is the relative spatial and logical arrangement of the different parts/sub-assemblies of the product with respect to each other. The qualifier *relative* in the definition distinguishes the configuration from the actual physical realization or geometry of the product. The relational nature of the configuration lends itself conveniently to graph-based representations. As a result it has become common practice to refer to the topology of these representing graphs as the topology of the assembly [9]. The EOL disassembly relevant topology will be referred to as the *blocking topology* henceforth to distinguish it from the other design concerns that are affected by the product topology (such as product modularity). The terms *assembly* and *product* will be used interchangeably in this study.

The blocking topology controls the possible disassembly sequences for a given assembly structure [10–12]. In an earlier paper [13], we proposed a formalism to describe the effect of the product configuration on the achievement of the objectives of EOL disassembly. In the context of the formal model, the economic objectives motivating DfD were used to identify three general redesign strategies, namely

(a) reduce the cost of each disassembly operation, (b) increase the value of the product segments, and (c) rearrange the value and cost elements in the disassembly sequence. The last strategy is most affected by the blocking topology of the product. The influence of the blocking topology has been widely recognized and the different approaches used to address the issue of optimizing the blocking topology for EOL disassembly can be grouped into two categories. The first category consists of approaches that prescribe to the designer blocking topologies that have known desirable disassembly characteristics. Examples are the design guidelines to increase the parallelism in the assembly [14,15] reduction of the sequence dependence of the assembly to improve the product's disassembly characteristics [16]. The second category consists of approaches that focus more on the evaluation of the topology using different indices. Some examples include, Jovane et al. [17] who proposed a set of indices using fuzzy logic constructs to identify the disassembly operations to be improved by redesign depending on their associated cost, revenue and the blocking topology. Feldmann and Meedt [18] proposed indices to analyze the effect of the product structure. The first index, the *extraction factor*, identifies the components that have high potential

profit but require longer disassembly times. The other index, called the *separation potential*, identifies the fasteners that require longer disassembly times and block parts that have high revenues. Zussman et al. [19] proposed a disassembly evaluation to support product recycling. Though the methods belonging to these two categories address definite aspects of the problem, they provide little in terms of concrete design advice especially during redesign when the question of *how* the poor topological characteristics of a given assembly are to be improved is of great importance.

A number of obstacles need to be overcome to develop methods to identify the specific design changes required to improve the configuration for disassembly. One significant obstacle arises due to the fact that the blocking topology is an abstraction derived from the geometry of the product assembly. So a change in the topology can only be achieved by appropriate modifications to the product geometry. Therefore, the identification of the geometric modifications to be made for a desired change in the topology is a problem that needs to be addressed. Closely associated with this issue are the problems of dealing with complicated assembly geometries, assemblies with large numbers of parts, and the handling of complex design constraints [20]. The presence of these daunting obstacles and the benefits associated with overcoming them to whatever extent possible is the basic motivation behind the present work. With this motivation, a model for configuration optimization through geometric modification is proposed in this paper.

This model builds on the configuration-value (CV) model [13], described briefly in the next section, and the Cellular Automata based approach to assembly design proposed by Minami et al. [20]. In the latter work, a Cellular Automata representation was used to generate sequences for assembly planning. The representation was developed as a possible way to bypass typical computational difficulties in design optimization by providing a simplified version of the actual geometry of assemblies that allows relatively simpler and quicker geometric reasoning and simulation. Further, the ‘atomic’ representation is suited for the generation of multiple geometries (and is of particular import for the ideas presented in the present work).

The layout of the paper is as follows: The problem is formulated in detail in Section 2. The development of the model using this formulation is presented in Section 3. The computational test-bed developed to test the claims of the model is described in Section 4. Examples of some experiments conducted are presented in Section 5. Section 6 sets out to summarize and discuss the findings and Section 7 contains our concluding remarks.

2. Problem formulation

2.1. Rationale

Let ξ be a finite set of assemblies with each member having identical disassembly constraints. Let each member

g of ξ have a unique blocking topology associated with it given by τ_g , where every τ_g defined over ξ belongs to a set τ . This can be expressed as a function, $T: \xi \rightarrow \tau$ where $\tau_g = T(g)$. Let the suitability of a blocking topology τ_g for EOL disassembly be expressible as a positive real number E_g defined by the function, $e: \tau \rightarrow \mathbb{R}^+$ such that $E_g = e(\tau_g)$. Also, the suitability of a blocking topology for EOL disassembly depends inversely on E_g . The most suitable assembly in ξ has $E_g = 0$. The task of determining the geometric modifications to improve the configuration or to reduce E_g could be approached in two different ways.

Since, the suitability of an assembly for EOL disassembly is determined by E_g , a single function can be obtained by composing e and T such that $(e \circ T) = p: \xi \rightarrow \mathbb{R}$. Thus, $E_g = e(T(g)) = p(g)$. The first approach to improve configuration is by obtaining the inverse of p . If the inverse function, $p^{-1}: \mathbb{R}^+ \rightarrow \xi$, is known then by specifying the desired E_g , the corresponding g' could be obtained. Therefore, the modifications required to improve g would be those that transform g to g' . A problem with this method is that T has a many-to-one mapping function. So even if e^{-1} is obtained, a valid T^{-1} would be difficult to define as there would be more than one geometry corresponding to the same τ'_g . The question would then arise as to which of these possible geometries is to be chosen as the redesigned g' . This is not a trivial question during redesign as the design freedom available to make changes to the geometry is low and any arbitrary geometry g' cannot be chosen. However, this inverse approach is very useful during the early stages of the design process where such freedom is available. Matter of fact, it is the basic rationale of the prescriptive methods mentioned in Section 1. These prescriptive methods identify members of τ that have $e(\tau_g)$ equal to or are close to zero, which is equivalent to obtaining the inverse of e^{-1} for some desirable E_g . But most prescriptive methods go no further. The fact that T is a many-to-one mapping function can be advantageously used since the designer has the freedom to identify a convenient geometry that realizes the prescribed topologies and satisfies design requirements.

The problem of identifying the modifications to be made is most critical during redesign where the freedom to make changes to the design is very limited. The second approach to configuration optimization is based on this constraint. The basic idea is to explore the space of geometric modifications that are possible during the redesign to find the improved geometry g' . This takes the form of an iterative process of making small or differential modifications to the assembly to be improved, g , to obtain g' and subsequently evaluating $p(g')$ till a satisfactory value of $E_{g'}$ is obtained, if at all possible.

This idea of differential modification leads to a number of questions such as: How can this improvement by differential changes be efficiently achieved? Can any particular geometry be improved by differential changes? Can all geometries be equally improved by differential changes? What

determines the ‘improvability’ of an assembly by differential changes? In order to focus on answering these questions, we use simple block world assemblies for conducting our investigations. The block-world assemblies and real-world assemblies are indistinguishable in the topological domain the implications of the proposed model are not greatly affected by this simplifying choice. Further, for the sake of simplicity, ξ is chosen to be a finite space of assemblies rather than the actual infinite space. The function p is described in Section 2.2. Section 2.3 describes the assembly space. In Section 2.4, a specific reformulation of the problem in terms of p and ξ is presented.

2.2. The function, p

The configuration-value (CV) model was proposed to provide a formal framework to describe the configuration redesign problem for EOL disassembly [13]. In this model the sequences permitted by the given assembly’s blocking topology define a characteristic subspace in a larger sequence space. This general sequence space is the set of all the disassembly sequences possible for every blocking topology possible on the same set of components/sub-assemblies. Each of the members of the sequence space is mapped to a real number line allowing the definition of a metric on the sequence space. The relationship between the metrics on the subspace and the entire sequence space are then used to identify a target for redesign initiatives. The function, p , is derived from the relation between the metrics on the subspace and sequence space. Before describing p , a brief overview of the CV model concepts is provided below.

2.2.1. CV model: basics

The CV model is restricted to the analysis of non-destructive disassembly where complete disassembly is not a necessary constraint. Also, only one component/sub-assembly (also referred to as part in this paper) is removed at every stage of the disassembly.

If Q is an assembly having m parts that need to be disassembled then a disassembly sequence on Q is expressed as a partially ordered set or poset, $\mathbf{S} = (Q, <)$, where $x < y$ implies that part y is removed prior to part x ($x, y \in Q$). The sequence space, Ω , is defined as the set of all such sequences of length m , feasible or not, on the m parts of Q . Hence, $\Omega = \{\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \dots, \mathbf{S}_r\}$ where $r = m!$. The sequence subspace, Ω_Q , is the subset of Ω containing all the feasible sequences possible on Q .

Due to the economic objectives of EOL disassembly each part has a particular EOL value, which will be referred from this point on simply as value (refer to Viswanathan and Allada [13] for a more detailed discussion on the economic arguments). The distribution of value associated with a sequence is called its *value precedence* (VP). The disassembly sequences where the highest valued parts can be extracted early in the disassembly are very desirable. So, the ideal

sequence is one where the parts are removed in the descending order of their value. The VP of this sequence is therefore called the *desired value precedence* (DVP). This sequence, \mathbf{S}_{dvp} , belongs to Ω but may not belong to Ω_Q . Without loss of generality it is assumed that no two parts have the same value. Therefore, no two sequences in the space have identical VP_s and the \mathbf{S}_{dvp} is unique.

The distance between any two sequences in Ω , $\delta(\mathbf{S}_i, \mathbf{S}_j)$ or $|\mathbf{S}_i - \mathbf{S}_j|$, is equal to the number of ordered pairs in \mathbf{S}_i that are not in \mathbf{S}_j . Therefore, if $\mathbf{S}_i = a < b < c < d$ and $\mathbf{S}_j = a < c < b < d$ then $\delta(\mathbf{S}_i, \mathbf{S}_j) = 1$ as they differ by one ordered pair $b < c$. Since, \mathbf{S}_{dvp} is the unique ideal sequence, any sequence \mathbf{S}_x where $\delta(\mathbf{S}_{\text{dvp}}, \mathbf{S}_x) > 0$ is non-ideal. Hence, the farther a sequence is from \mathbf{S}_{dvp} , the lower is its suitability for value extraction during EOL disassembly.

The extent of departure of a sequence from \mathbf{S}_{dvp} is better quantified using the concept of the petition function. The petition function maps the VP of a sequence \mathbf{S} to a single real-valued number $\mathbf{P}_\mathbf{S}$ as given by Eq. (1). By this Eq. (1), $\mathbf{P}_{\text{dvp}} = 0$. So for any \mathbf{S}_x and \mathbf{S}_y in Ω , if $\mathbf{P}_x > \mathbf{P}_y$ then it implies that \mathbf{S}_x is ‘non-ideal’ to a greater extent than \mathbf{S}_y .

$$\mathbf{P}_\mathbf{S} = \sum_{i=1}^m \sum_{j=1}^m P_{ij}, \quad (1)$$

where

$$P_{ab} = \begin{cases} T_1 e^{T_2} & \text{if } a < b (a, b \in Q) \text{ and } V_b < V_a, \\ 0 & \text{Otherwise,} \end{cases}$$

where,

$$T_1 = (V_a - V_b) / (V_{\text{max}} - V_{\text{min}}),$$

$$T_2 = (V_a - V_{\text{min}}) / (V_{\text{max}} - V_{\text{min}}),$$

$$V_{\text{max}} = \text{maximum value of a part in } \mathbf{S},$$

$$V_{\text{min}} = \text{minimum value of a part in } \mathbf{S},$$

$$V_{\text{max}} \neq V_{\text{min}}.$$

Using the above definitions, Ω can be conceptualized as an undirected graph where an edge exists between any two sequences \mathbf{S}_i and \mathbf{S}_j if $\delta(\mathbf{S}_i, \mathbf{S}_j) = 1$ ($\mathbf{S}_i, \mathbf{S}_j \in \Omega$). Each vertex of this graph, i.e., every sequence, has $(m - 1)$ neighbors where m is the number of parts in the assembly, Q . This graph is connected implying that there exists at least one path between any two sequences, \mathbf{S}_i and \mathbf{S}_j belonging to Ω . It is also seen that the Ω_Q for a valid Q is connected though every connected subset of Ω need not necessarily correspond to a real assembly.

2.2.2. Targets for configuration redesign

The DVP, though ideal from the value perspective, does not consider the disassembly costs. On the other hand, the *optimal value precedence* (OVP) or \mathbf{S}_{ovp} is the sequence in Ω_Q that considers the optimal trade-off between the value extracted and cost of disassembly. \mathbf{S}_{ovp} may not be equal to \mathbf{S}_{dvp} . Therefore, the objective for configuration redesign is to make $\mathbf{S}_{\text{ovp}} = \mathbf{S}_{\text{dvp}}$ or to at least reduce $\delta(\mathbf{S}_{\text{dvp}}, \mathbf{S}_{\text{ovp}})$.

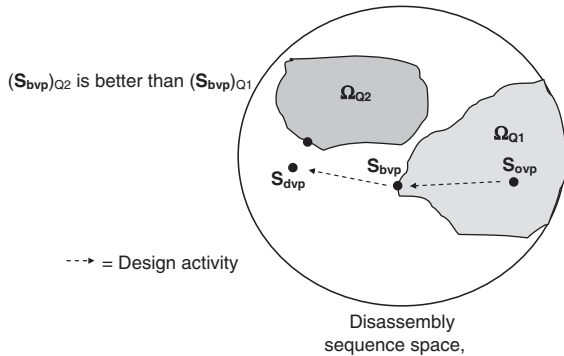


Fig. 1. Schematic of configuration-value centered DfD.

S_{ovp} is not necessarily the sequence having the lowest petition magnitude in Ω_Q . The sequence in Ω_Q having the lowest petition magnitude is called the *best value precedence (BVP)* or S_{bvp} . So, if S_{dvp} belongs to Ω_Q then $S_{bvp} = S_{dvp}$.

Therefore, the three sequences, S_{dvp} , S_{bvp} and S_{ovp} , can be used to characterize a given configuration. S_{dvp} is in effect the global reference point while S_{bvp} defines a local reference point for S_{ovp} . By this interpretation, the objective to make $S_{ovp} = S_{dvp}$ could be approached in two stages: (1) reducing the distance between S_{bvp} and S_{dvp} and, (2) reducing the distance between the S_{ovp} and S_{bvp} . This idea can be expressed using Fig. 1. Since this paper is primarily concerned with configuration related issues rather than cost, only the first objective of making the $S_{bvp} = S_{dvp}$ is investigated. Therefore, P_{bvp} can be used as a simple measure of the suitability of an assembly for value extraction during EOL disassembly, i.e., $E_g = p(g) = P_{bvp}$ of g .

2.3. Assembly space, ξ

The ideal characteristics of the assembly space ξ are listed below:

- Any member of the space can be transformed deterministically to any other member by the successive application of a defined set of differential (small) geometrical modifications.
- The set of differential geometrical modifications provides a natural quantitative measure of the difference in geometry between any two members of the space.
- It has a finite number of members such that any two members are comparable both in geometry and mapping using the function p . The mapping requirement mandates that the members of ξ be compatible with the assumptions used in the CV model as it provides the function p .

In this paper, ξ is similar to the Cellular Automata assembly representation in [20]. It is defined as the set of all m -part assemblies with identical disassembly constraints

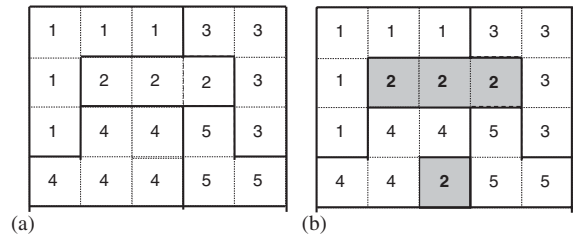


Fig. 2. Examples of valid and invalid members of ξ .

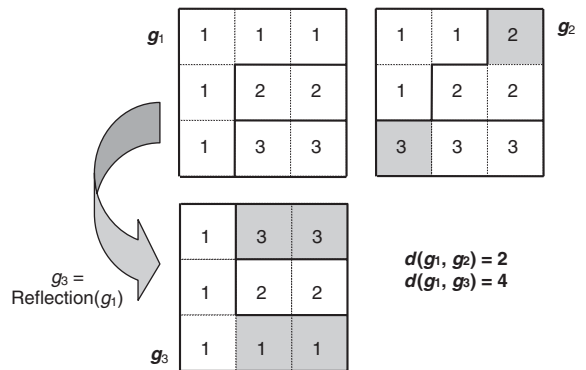


Fig. 3. Distance between members of ξ .

defined on a finite $M \times N$ grid such that:

1. Each part is the union of an integral, non-zero number of connected cells of the grid. Any two cells are connected if they share an edge.
2. The assembly fully covers all the cells of the grid.
3. The assembly can be fully disassembled such that:
 - a. Only one part is removed at each step.
 - b. The parts are removed along straight line paths in the $X+$, $X-$, $Y+$ or $Y-$ directions.
 - c. The disassembly is monotonic i.e. all disassembled parts are do not occupy any temporary intermediate positions.
 - d. The product is not repositioned during the entire disassembly process.
4. Each part has an EOL value associated with it that is independent of the size or shape of the part.

Examples of valid and invalid members of ξ are shown in Figs. 2a and b, respectively.

The distance measure, $d(g_1, g_2)$, between any two members g_1 and g_2 of ξ is defined as the number of cells in g_1 that do not belong to the same part as the corresponding cells of g_2 . For example, the distance $d(g_1, g_2)$ in Fig. 3 is equal to 2. It can also be seen that $d(g_1, g_2) = d(g_2, g_1)$. The

distance is measured using a fixed reference so $d(g_1, g_2) = 0$ if and only if $g_1 = g_2$ with respect to the fixed reference. So, if g_1 and g_2 are equivalent, i.e., if one can be converted to the other by rigid body transformations, it does not follow that $d(g_1, g_2) = 0$. For example, in Fig. 3, the distance between g_1 and g_3 (g_3 obtained by reflection of g_1 about the $Y+$ direction) is not equal to zero.

Similar to the distance measure δ defined in the sequence space Ω , the measure, d , provides ξ with a structure that can be conceived as an undirected graph where the members of ξ are the vertices and an edge exists between any two vertices, g_1 and g_2 , if $d(g_1, g_2) = 1$. The distance between any two members of ξ is effectively the magnitude of the change required to transform one member into the other.

Apart from the four assumptions mentioned previously in this section, the following assumptions are made for the proposed model:

1. This model does not consider functional constraints. Without considering functional constraints, the differential approach may result in absurd assemblies if a function critical feature is removed to facilitate disassembly. The reason for omitting functional constraints was not by oversight but to focus on the fundamental principles that determine the configuration effects of a product independent of the application context (interpretation). The incorporation of functional constraints into a differential representation is an issue that would need to be addressed while optimizing actual products.
2. In this study, we assume that no two parts of a given assembly have the same economic values so that we can illustrate our method using a simplified example problem. However, relaxing this assumption can be accommodated in our model by arbitrarily break the tie if two parts have equal values.
3. The model considers only differential (i.e., small) geometrical changes to the assembly to improve its disassembly characteristics. Hence, the proposed model performs a local search on the landscape to optimize the assembly. The focus of this model is not on making major “overhaul” to the product configuration to improve its disassembly characteristics.
4. The model is designed to take into account the complete disassembly scenario. The choice of how much to disassemble (selective disassembly) is based on the cost–benefit analysis that is highly case specific. This model presents the complete disassembly scenario so that the designers have the freedom to choose the cut-off point for selective disassembly. Moreover, one can use the complete disassembly scenario as the ideal goal for benchmarking purposes.
5. The value of the parts is invariant to changes in part geometry. Though may be not true in all cases, this simplification was made with the reasoning that since the approach is differential in nature small changes in the geometry may not result in large changes in the part value.

2.4. Problem reformulation

Using the characteristics of ξ and the function p , the problem can be reformulated as a combinatorial optimization problem that is described below.

Consider an undirected graph that represents the structure of ξ . Using the function p , a height can be associated with each member of ξ proportional to the magnitude of its associated \mathbf{P}_{bvp} . This can be visualized as a landscape in which assemblies having low \mathbf{P}_{bvp} 's form the ‘valleys’ and assemblies with high \mathbf{P}_{bvp} 's form the ‘hills.’ The lowest point in the landscape represents assemblies having $\mathbf{P}_{bvp} = \mathbf{P}_{dvp} = 0$.

Hence, the improvement of an assembly g by differential modification implies the identification of an assembly h in ξ such that h is in a valley with respect to g (i.e., $h \mathbf{P}_{bvp} < g \mathbf{P}_{bvp}$) and h lies in the neighborhood of g (i.e., $d(g, h) \leq k$, where k is the distance from g that is acceptable as a differential modification). Therefore, a local configuration optimization is essentially the process of performing a local search on the landscape, around the assembly to be optimized.

With this combinatorial formulation of the problem the next step is to characterize the landscape. Understanding the nature of this landscape, for example, the ‘ruggedness’ of the landscape, the distribution of the local optima, the probability of a particular local optimum lying in the neighborhood of any given point on the landscape, etc., is critical to develop mechanisms to search the landscape effectively [21]. This characterization is addressed in the next section.

3. Characterization of the optimization landscape

3.1. General characteristics

An important feature of an optimization landscape is its ruggedness. The ruggedness of an optimization landscape is a measure of the correlation between adjacent points in the landscape [21]. When there is a high correlation in the landscape with adjacent points having a similar height, then the landscape is ‘smooth.’ As the correlation decreases the landscape becomes increasingly rugged and the height of a point provides little information about the height of any of its neighboring points. The smoothness of the landscape is dependent on the extent of the interrelation between the parameters that define the space. When the parameters are not related to the effect of each parameter on the height of the point is additive, i.e., each parameter has an effect on the height largely independent of the other parameter values. As a result the landscape is smooth. But as the effect of the parameters become increasingly interrelated, the landscape tends to get increasingly rugged.

In the present problem, the states of each cell are the parameters defining an assembly as a point in ξ . The effect of each cell of an assembly on the \mathbf{P}_{bvp} is highly dependent on the states of the remaining cells of the assembly.

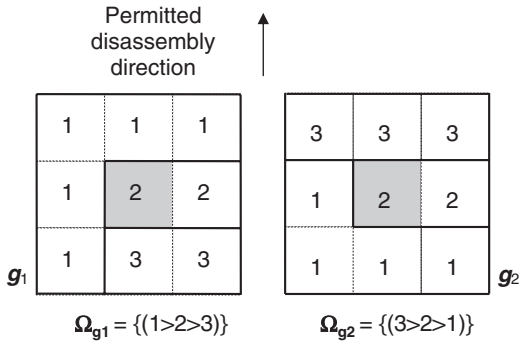


Fig. 4. Example of interrelated nature of cell state effect on \mathbf{P}_{bvp} .

In Fig. 4, though the highlighted cell has the same state in both cases the associated effect is very different. This is because the states of the other cells in the assembly are different in both cases showing the significant dependence of the effect of a cell on the states of the other cells. Due to this high interrelation between cell states, it can be expected that the landscape of ξ shows a high degree of ruggedness (and the characterization is performed with this assumption).

Now, p is a many-to-one mapping function. Hence, more than one member of ξ can have the same \mathbf{S}_{bvp} . This suggests the utility of identifying equivalence classes based on the heights of the different members on the landscape. Let \mathbf{R} be an equivalence relation defined on ξ such that $\mathbf{R} \subseteq \xi \times \xi$ and an ordered pair (x, y) belongs to \mathbf{R} if and only if $x\mathbf{S}_{bvp} = y\mathbf{S}_{bvp}$ ($x, y \in \xi$). Therefore,

- (a) if $x \in \xi$ then $(x, x) \in \mathbf{R}$;
- (b) if (x, y) is in \mathbf{R} then (y, x) is in \mathbf{R} ; and
- (c) if (x, y) and (y, z) belong to \mathbf{R} then (x, z) belongs to \mathbf{R} .

This relation can be used to partition ξ into a collection of subsets denoted by ξ/\mathbf{R} . A set E belongs to ξ/\mathbf{R} if and only if there is a member x of ξ such that $E = \{y|y \in \xi \text{ and } (x, y) \in \mathbf{R}\}$. Each such set E is called an equivalence class. An equivalence class E where all the members have $\mathbf{S}_{bvp} = {}^k\mathbf{S}$ (where ${}^k\mathbf{S}$ is a sequence in Ω having a unique index k) is denoted as $[{}^k\mathbf{S}]$. So for a given ξ there are $m!$ equivalence classes corresponding to each of the sequences in Ω .

Characterization of the landscape is greatly simplified using these equivalence classes. The relationship between equivalence classes is discussed in Sections 3.2–3.4 and the internal structure of each class is discussed in Section 3.5.

3.2. Isomorphism property

An assembly g is said to be *isomorphic* to h if g can be transformed to an assembly g' such that $d(g', h) = 0$, only by interchanging the part identities of g and without making any

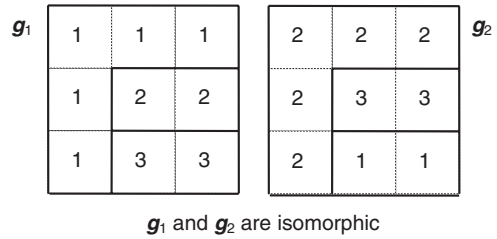


Fig. 5. Isomorphism.

modification to the geometry of g , or performing any rigid body transformations to it. For example, g_1 is isomorphic to g_2 as shown in Fig. 5.

As g is a valid assembly the validity of its geometry does not change by merely interchanging the part identities. Therefore,

Lemma 1. *If an assembly g belongs to ξ then all assemblies that are isomorphic to g also belong to ξ .*

From this simple observation it can be further inferred that

Lemma 2. *If there exists an assembly g in ξ that permits exactly one sequence i.e. $|\Omega_g| = 1$, then every \mathbf{S} in Ω is \mathbf{S}_{bvp} for some assembly in ξ i.e., $[\mathbf{S}]$ is non-empty $\forall \mathbf{S} \in \Omega$.*

Refer to Appendix A for the proof of Lemmas 1 and 2.

3.3. Relative size of equivalence classes

From the CV model it is seen that every valid assembly g has a sequence set Ω_g that is connected (refer to Section 2.2.1) but every connected set in Ω does not necessarily correspond to a valid assembly. However, a function to determine whether a given set of connected sequences corresponds to a valid assembly is still not known. Using Lemmas 1 and 2, we can prove the following:

Lemma 3. *$[\mathbf{S}_{wvp}]$ is the smallest equivalence class, where \mathbf{S}_{wvp} is the sequence in Ω having the worst value precedence, i.e., the largest \mathbf{P} .*

Refer to Appendix B for the proof of Lemma 3. With a similar approach it can be easily shown $[\mathbf{S}_{dvp}]$ is the largest equivalence class, where \mathbf{S}_{dvp} is the sequence in Ω having the lowest \mathbf{P} . It can be safely concluded that in general the size of $[\mathbf{S}]$ tends to get larger as $\mathbf{P}_{\mathbf{S}}$ gets smaller.

If the landscape is rugged it can be surmised that there exists a possibility that an assembly belonging to a particular equivalence class could be transformed with k changes to an assembly belonging to any other equivalence class. So if each of the equivalence classes were of equal size then for a given assembly in ξ with ${}^a\mathbf{P}_{bvp} > 0$, the probability of

finding an assembly having $\mathbf{P}_{bvp} < {}^a\mathbf{P}_{bvp}$ in its vicinity would decrease uniformly as ${}^a\mathbf{P}_{bvp}$ decreases. But from Lemma 3 it can be seen that the sizes of the equivalence classes are not equal and get larger as \mathbf{P}_{bvp} decreases. Hence, the decrease in the probability of finding an improved assembly with a lower \mathbf{P}_{bvp} in the vicinity decreases non-linearly such that the probability of finding an improvement for the assembly is always higher than if the equivalence classes were of the same size. Furthermore, the rate of decrease of the probability of finding an improvement is greater for assemblies having \mathbf{P}_{bvp} 's close to zero as compared to those having \mathbf{P}_{bvp} 's close to \mathbf{P}_{wvp} . It can be definitely expected that these results would have very important implications on the effectiveness of a local search algorithm on the landscape.

3.4. Scalability

An important aspect of a differential approach to configuration optimization is the effect of an increase in the number of parts m on the improbability of assemblies in a particular equivalence class. Consider the case where the number of parts increases without a change in the maximum and minimum values of the parts and a change in the grid size. For this case, an estimate of the change in the average minimum difference between the \mathbf{P}_{bvp} 's of each equivalence class due to an increase in the number of parts is derived below.

By definition a petition is given by

$$p_{ab} = \frac{V_a - V_b}{V_{\max} - V_{\min}} e^{(V_a - V_{\min}) / (V_{\max} - V_{\min})}, \quad (2)$$

where, V_{\max} is the maximum part value and V_{\min} the minimum part value.

This can be approximated to

$$p_{ab} \approx (V_a - V_b) / (V_{\max} - V_{\min}), \quad (3)$$

The average minimum difference in values between any two parts is

$$\approx V_{\text{diff}} / (m - 1), \quad (4)$$

where $V_{\text{diff}} = V_{\max} - V_{\min}$.

From Eqs. (3) and (4), the size of the smallest petition can be estimated to be

$$p_{\min} = (V_{\text{diff}} / (m - 1)) / V_{\text{diff}} = 1 / (m - 1). \quad (5)$$

Hence, the size of all other petitions can be expressed in terms of p_{\min} as

$$\approx b p_{\min}, \quad (6)$$

where b is an integer such that $1 \leq b \leq (m - 1)$.

The CV model assumes that every sequence can be expressed as a poset chain, i.e., where every part is comparable to every other part in the sequence. As \mathbf{S}_{wvp} is a chain and has the maximum number of petitions it implies that

the parts are removed in the ascending order of value. So, the magnitude of \mathbf{P}_{wvp} can be estimated using

$$\mathbf{P}_{wvp} \approx (m - 1)p_{\min} + 2(m - 2)p_{\min} + \dots + k(m - k)p_{\min} + \dots + (m - 1)p_{\min} \quad (7)$$

$$= p_{\min} \sum_{i=1}^{m-1} i(m - i). \quad (8)$$

Eq. (8) can be expressed as follows:

$$p_{\min} \left[m \sum_{i=1}^{m-1} i - \sum_{i=1}^{m-1} i^2 \right]. \quad (9)$$

Using standard expressions for series summations, Eq. (9) can be expressed as follows:

$$p_{\min} \left[\frac{m^2(m - 1)}{2} - \frac{m(m - 1)(2m - 1)}{6} \right]. \quad (10)$$

By substituting the expression for p_{\min} from Eq. (5) in Eq. (10), the \mathbf{P}_{wvp} can be expressed as follows:

$$\mathbf{P}_{wvp} = \frac{m(m + 1)}{6}, \quad (11)$$

All \mathbf{P}_{bvp} 's in ξ lie in the interval $[0, \mathbf{P}_{wvp}]$. The average minimum difference between \mathbf{P}_{bvp} 's is approximated as follows:

$$D_m = \frac{m(m + 1)}{6(m - 1)!}. \quad (12)$$

Suppose the number of parts is increased by one to $m + 1$, then the average minimum difference between the \mathbf{P}_{bvp} 's is obtained by substituting $(m + 1)$ for m in Eq. (12),

$$D_{m+1} = \frac{(m + 1)((m + 1) + 1)}{6(m + 1 - 1)!} = \frac{(m + 1)(m + 2)}{6m!}. \quad (13)$$

Therefore, the extent of change in this average minimum difference between \mathbf{P}_{bvp} 's is given by

$$\Delta D = (D_m - D_{m+1}) / (D_m), \quad (14)$$

$$\Delta D = \frac{m^2 - m - 2}{m^2}, \quad (15)$$

The variation of ΔD versus m is plotted in Fig. 6. From Fig. 6, it can be seen that the effect of adding a part on D_m is initially low but increases rapidly as m increases. For large m , the effect of adding a new part is very large but as m increases tending asymptotically to a point where every new addition results in an almost 100% reduction in D_m .

The implication of this result is that given an assembly belonging to a particular equivalence class $[\mathbf{S}]$ the number of equivalence classes having better \mathbf{P}_{bvp} 's increases greatly as the number of parts increases for a given grid size. But the difference in magnitude between these better \mathbf{P}_{bvp} 's also decreases. As a result, the probability that a small geometric change Δg would result in a large improvement in the \mathbf{P}_{bvp} of an assembly g decreases rapidly as m increases.

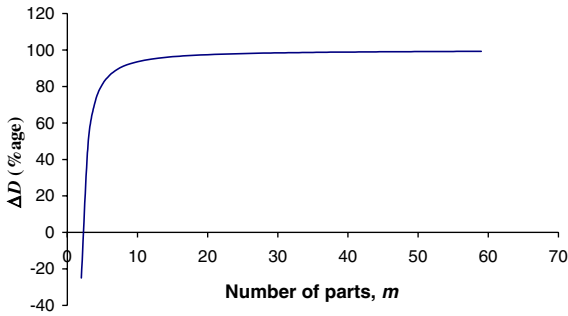


Fig. 6. Variation of ΔD with m .

3.5. Structure of an equivalence class

The members of an equivalence class almost never occur in isolation. At present the evidence to show that all the members of an equivalence class are connected has remained elusive but it can be confidently stated that each equivalence class consists of a small number of connected components where each component has more than one assembly. So, each of these connected components would form ‘plateaus’ on the landscape. Each of these plateaus has an edge and a central flat region so assemblies lying close to the periphery require fewer changes to be transformed to a different equivalence class as compared to assemblies lying far from

the periphery. The existence of these plateaus does not contradict the assumption of ruggedness of the landscape. This is because these plateaus are located in a high-dimensional space so the shape of a plateau depends on the equivalence classes with which it is compared, any of which may lie adjacent to a given plateau. An assembly may lie on the periphery with respect to one equivalence class and in the interior with respect to another equivalence class. While the blocking topology decides which equivalence class an assembly belongs to, the location of an assembly on a plateau is determined by the geometry of the assembly. In order to provide an intuitive idea of the geometric effect on the specific location of an assembly on its plateau, consider the example shown in Fig. 7.

A general inference, as also visible in the example above, is that the chances of finding a valley in the immediate neighborhood of any assembly increases as the regularity of the parts in an assembly increases. Regular refers loosely to the extent of convexity of the part (a rectangular part is considered to be highly regular). This suggests that the improbability of an assembly depends on its regularity as the extent of modification required to improve highly regular assemblies would not really qualify as differential changes. This leads to the question of how the improbability of an assembly can be estimated a priori from its regularity features. This is an important issue that requires to be investigated but is not addressed here and is left as a general observation awaiting further research.

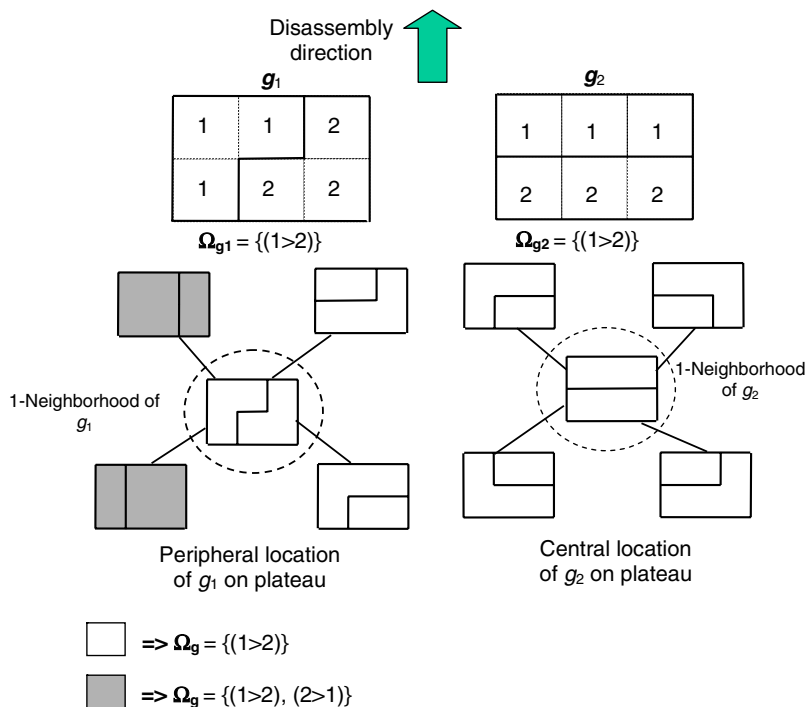


Fig. 7. Effect of geometry on location of assembly on plateau.

To conclude, the different properties of the landscape discussed in this section reveal the complexity that would need to be addressed in developing an optimization procedure to navigate this landscape. In the next section, the local search mechanism that was developed to investigate the landscape and which led to the serendipitous uncovering of some of the properties discussed above is presented.

4. Local search mechanism

4.1. Disassembiosis: rationale

Due to the very large number of members in ξ and the NP-hard nature of identifying the \mathbf{P}_{bvp} of an assembly, determining whether an improved assembly h exists in the k -neighborhood of a given assembly g is a non-trivial problem. So answering this question of whether such an assembly exists is inseparably linked to the procedural aspect of how such an assembly h is to be identified. Since a brute force method for the determination of h is impractical, a cumulative random search mechanism was used. Cumulative random search techniques like genetic algorithms, evolutionary programming, and simulated annealing have been effectively used to search large spaces [22,23]. In this study, a hierarchical evolutionary programming mechanism modified for the specific needs of the disassembly application is used to search the landscape. This hierarchical mechanism is called *disassembiosis*, derived from *disassembly* and *symbiosis*; the reason for the name will become apparent shortly.

The search strategy implemented by the disassembiosis method is as described below. The process begins at the specific assembly of interest, g_0 . The search is performed by successively sampling a predefined number (or population) of assemblies, K , in each iteration. Two different sampling patterns can be identified—radial and lateral. Radial sampling refers to the strategy to successively sample assemblies at progressively greater distances from assemblies in the previous generation (and hence from g_0). In the lateral sampling procedure, the assemblies that are sampled are not at increasing distances from g_0 but lie within regions defined by assemblies in the previous generation. This is to more thoroughly sample areas already traversed by radial sampling. The fraction of the K points to be sampled using the radial and lateral methods are defined a priori.

In the first generation, the set G_1 of K assemblies is randomly chosen from all the assemblies lying at a unit distance from g_0 . Since the focus is on identifying improvements, the choice of assemblies for the second generation G_2 is not random. G_2 is obtained from the assemblies G_1 based on the magnitude of their \mathbf{P}_{bvp} 's with respect to the \mathbf{P}_{bvp} of g_1 and each other with a preferential selection of assemblies that are improvements over g_1 . The search process is not stopped when the first improved assembly is obtained but when no further improved assembly is detected over a number of generations. In the jargon of evolution program-

ming, the K points are the individuals in each generation. The radial sampling is achieved using a mutation operator and lateral sampling using a crossover operator.

4.2. Implementation

The evolution was performed with an elitist, fitness proportionate selection with replacement. The probability of the i th point in a particular generation being selected during a trial for a crossover or a mutation operation is given by

Selection probability of i th individual

$$= \frac{1 - {}^i\mathbf{P}_{bvp}/\mathbf{F}}{n - 1}, \quad (16)$$

where $\mathbf{F} = \sum_{j=1}^n {}^j\mathbf{P}_{bvp}$ and n is the generator population size.

4.2.1. Mutation

In the present implementation, radial sampling by mutation advances only by unit distance in every generation, i.e., only one cell is transformed. Transformation refers to the change in state of a cell from belonging to a part x to some other part y . The cell to be transformed is obtained by first randomly choosing a part (where each part has an equal probability to be chosen). For the selected part, a cell belonging to that part is randomly selected that satisfy predefined cell neighborhood patterns. The change in state of this cell is dependent on the states of its neighboring cells.

4.2.2. Crossover

To perform the crossover operation, two parent assemblies from the previous generation are selected as described above. The offspring assemblies are obtained by an exchange of 'cells' between the parents. Due to the differences in the geometry of the parents, it is critical that the crossover results in offspring that are valid members of ξ , i.e., satisfy the conditions mentioned in Section 2.3. The following procedure is used to increase the chances of obtaining a successful crossover is described below.

A difference array (DA) of the same size as the parent assemblies is first defined. The state of a cell, L_{ij} , of the DA is obtained by taking the difference of the corresponding cells of the parent assemblies and setting the non-zero differences to 1, i.e.,

$$DA_{L_{ij}} = \text{parent1}_{L_{ij}} - \text{parent2}_{L_{ij}}. \quad (17)$$

A binary array is obtained, where the 1's represent the points of geometric incompatibility between the parents. Two parent assemblies and their corresponding DA are shown in Fig. 8a where the darkened cells have $L_{ij} = 1$ and the white cells have $L_{ij} = 0$.

For a successful crossover to take place, the crossover line should be such that it does not intersect adjacent cells

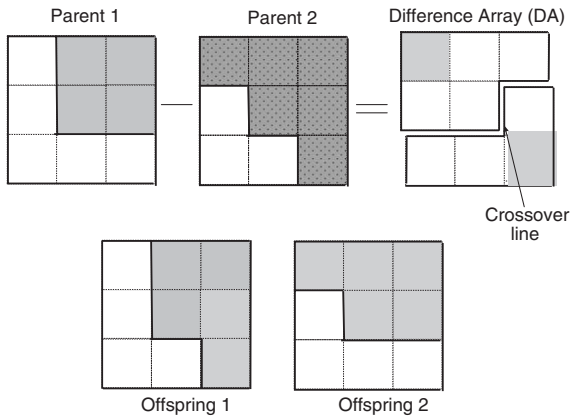


Fig. 8. Working of the crossover operation.

having 1-states. The procedure used to obtain the crossover line is as follows:

- Obtain the DA = (Parent)₁ – (Parent)₂.
- Determine the smallest rectangle encompassing all cells having 1-states.
- Identify the larger side, *l*, of this rectangle.
- The central partition line dividing *l* is used to divide the DA into two fragments such that it does not intersect two adjacent 1-state cells (do not share an edge that is common to adjacent 1-state cells). In order to satisfy this condition, the crossover line needs not be a single straight line but can consist of multiple connected segments as shown in Fig. 8a. Furthermore at no point should the crossover line intersect itself.

The parents exchange the corresponding cells defined by the crossover line to generate a pair of off-springs as shown in Fig. 8b. The crossover in the generators occurs without mutation. This procedure does not ensure the validity of the blocking relations, and in some situations, it may not be possible to obtain a valid dissection line. In such cases, the parents are mutated and transmitted to the next generation.

4.3. Sequence determination: implementation

Up to this point while discussing the evolution of the assemblies in ξ it was implicitly assumed that $p(g)(g \in \xi)$ is known and available. However, if this is not the case, then S_{bvp} has to be specifically determined for each assembly sampled. This can be done using a real-valued genetic algorithm. As this evolution identifies the P_{bvp} of the assemblies, its efficiency directly affects the selection probability of an assembly during the ξ search process. This mutual dependence of the two levels of evolution is the reason for the root *symbiosis* in disassembiosis.

The assembly sequence problem is a complex combinatorial optimization problem that has been addressed by many researchers [11,22–31].

The disassembly sequencing algorithm used in this study is based on the rationale that a sequence is obtained as a result of a series of decisions. At each stage in the disassembly a decision is made as to which part is to be removed of the parts that can be removed. So controlling the decisions in effect controls the sequence generated. The decision function is obtained as follows: Each of the parts is associated with a characteristic coefficient (c_j) where $0 \leq c_i \leq 1$. Using these coefficients a decision variable called the removability, r_i , is evaluated for each of the parts that can be removed at every stage of the disassembly. At a given stage in the disassembly, the removability value is obtained as shown in Eq. (18). On obtaining these removability values, the decision criterion used is to simply remove the part having the highest removability value associated with it. On removing this part, the same process is repeated at the next stage of the disassembly process and so on.

$$r_i = \begin{cases} V_i - \sum_{j=1}^n c_j V_j & (\forall j \neq i \text{ and } i \text{ is removable}), \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

As can be seen, these decisions are controlled by the characteristic coefficients of each part. Therefore by varying the coefficients different sequences are obtained. Using this principle the characteristic vector, **C**, containing a particular set of characteristic coefficients is used as the evolvable chromosome with each coefficient acting as a modifiable gene. A modifiable gene is a concept in which a gene can assume different values during evaluation. So given a particular assembly, a population of characteristic vectors is evolved with the selection pressure to determine the sequence having the lowest **P** or P_{bvp} .

The evolution is performed by the use of crossover and mutation operators, with an elitist, proportionate fitness selection with replacement. The probability of the *i*th individual being selected for crossover during a selection trial is given by Eq. (19). A single crossover point is chosen randomly for the crossover operation

Selection probability of *i*th individual

$$= \frac{1 - P_i/F}{n - 1}, \quad (19)$$

where $F = \sum_{j=1}^n P_j$ and *n* is the population size,

The site for mutation is chosen randomly, where all genes have an equal probability of being chosen. The selected coefficient, c_i , is mutated by adding to it a random number chosen over the interval $[-1, 1]$. The coefficient is then set to 1, if this sum exceeds 1 and to zero if the sum is less than zero.

The efficiency of the evolution was improved using an explicit genetic memory concept. Explicit memory-based approaches are capable of storing promising solutions from previous generations and using them later into the search population.

Consider a population of assemblies $G_1 = \{g_1, g_2, g_3, \dots, g_K\}$ where the function $p(g_i)$ is to be evaluated for each member of G_1 . On performing these evaluations, using the algorithm described above, a sequence S_{best} equal to or close to the S_{bvp} is obtained for each member of G_1 . Now, some members of G_1 are subsequently selected and appear in the next generation of assemblies G_2 after mutation and crossover. Let g_i be such a member of G_1 that is selected and appears in G_2 after modification as g'_i . Each member of G_2 again needs to be evaluated including g'_i . Rather than starting the evaluation of g'_i afresh, the evolutionary effort used in identifying S_{best} for g_i in the previous generation can be used to increase the efficiency of the evolution to find S_{best} for g'_i . This is done by using the characteristic vector associated with the S_{best} of g_i directly as a member of the initial population used to find the S_{best} for g'_i .

5. Experiments and results

The following three cases—assemblies *A*, *B* and *C*—(shown in Figs. 9a–c) are used to demonstrate the possibilities of optimization using differential modification. The parameters used to define ξ , p and the disassembiosis mechanism are given in Table 1.

The results of the evolution in all three cases produced interesting results. Samples of improved variants of assemblies *A* and *C* are shown in Figs. 10a and b, respectively. The plot of P_{bvp} of the best individual in each generation of a particular run of the three cases is shown in Fig. 11. From Fig. 11, it can be seen that assemblies *A* and *C* show distinct improvements in their topologies, while assembly *B* remains completely unaffected over the entire evolution. The assembly *B* was specifically chosen to display the effect of the location of an assembly on a plateau due to its regularity. It can be inferred here from Fig. 11 that assembly *B* is located far from the periphery of the plateau with respect to all the equivalence classes having lower P values. However, it will subsequently be revealed that the implications of regularity are deeper than what is presented here.

The other factor central to improvement by the present model is the extent of change in the starting assembly. The scatter plot of P_{bvp} of the best individual in each generation versus its distance from the starting assembly over all the runs is shown for each of the three cases in Figs. 12a–c. Since disassembiosis is a random local search mechanism, the smallest distance associated with a particular P on the plots is not necessarily the minimum distance at which such an assembly occurs from the starting assembly. Therefore, at this point the thesis of the paper that a differential approach can be used to improve the blocking topology of an assembly for EOL disassembly stands verified. Though not sufficient as an absolute proof, these results also conform to the different predictions made about the landscape as discussed below.

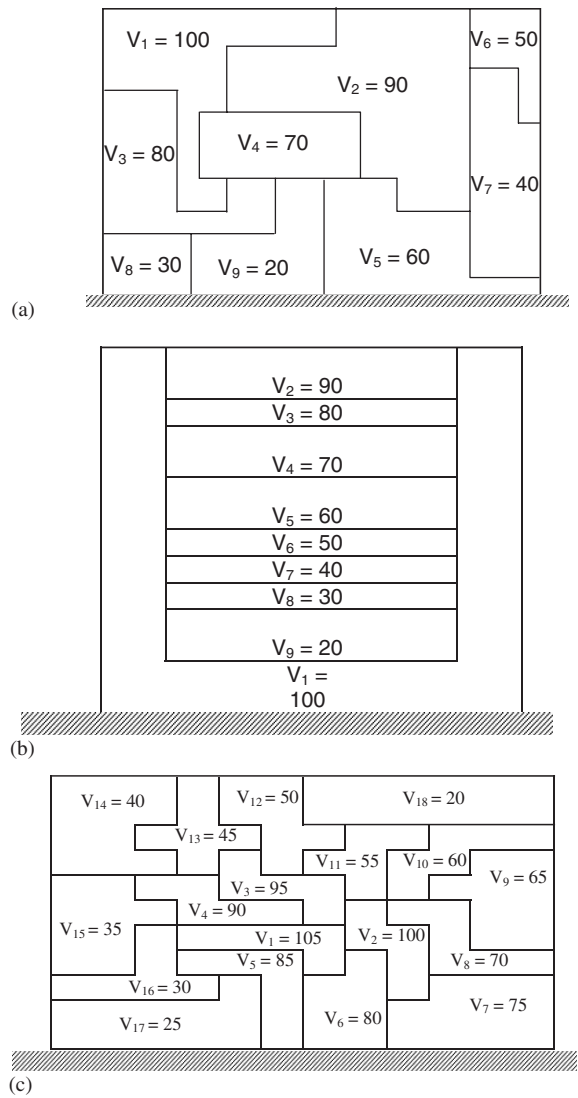


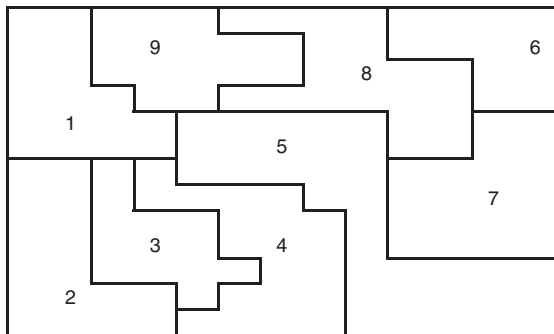
Fig. 9. Example assemblies: (a) assembly *A*, (b) assembly *B*, and (c) assembly *C*.

5.1. Scalability

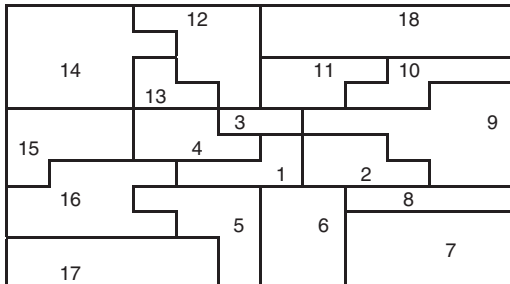
A significant feature from Figs. 12a and c is that the improvements in P with distance closely matches the predictions made that as the number of parts increases the probability that a differential change in the geometry of an assembly would result in a large improvement in P_{bvp} decreases rapidly. Comparing the plots of assembly *C* (having 18 parts) and that of assembly *A* (having 9 parts) the differences in extent of improvements at comparable distances is very visible, with the improvements being smaller in assembly *C*. This trend is also visible in the difference between the curves corresponding to assemblies *C* and *A* in Fig. 10 where the fitness improvement of assembly *C* is much

Table 1
Disassembiosis parameters

	Assembly A	Assembly B	Assembly C
Grid size	13 × 13	13 × 13	12 × 12
No. of parts, m	9	9	18
Max., min part values	100, 20	100, 20	105, 20
ξ evolution			
Population size	35	35	45
No. of generations	200	200	200
Sequencer evolution			
Population size	25	25	40
No. of generations	30	30	50



(a)



(b)

Fig. 10. Improved variants obtained after evolution using disassembiosis: (a) assembly A and (b) assembly C.

flatter than that of assembly A that shows considerably larger changes in \mathbf{P} .

5.2. Plateaus

A noticeable feature in Figs. 12a and c is the clustering of assemblies having the same \mathbf{P}_{bvp} 's at contiguous distances from the starting assembly. This in no way implies that if two assemblies x and y with the same \mathbf{P}_{bvp} have $d(x, g) = l$ and $d(y, g) = l + 1$ then $d(x, y) = 1$. However, the clustering observation definitely provides some credence to the assertion that members of the same equivalence class form connected plateaus in the fitness landscape.

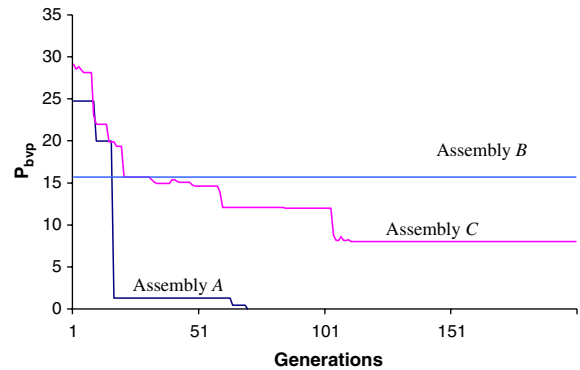


Fig. 11. Variation in \mathbf{P}_{bvp} of the best individual in each generation.

5.3. Relative size effects

In Figs. 12a and c, the number of assemblies explored at each \mathbf{P} show increasing trend as \mathbf{P} decreases. Putting the relative size effect and the observation of the plateauiness of each equivalence class suggests that the plateaus in an equivalence class increase in size as \mathbf{P} decreases. These plots do indicate that as \mathbf{P}_{bvp} of an assembly decreases the probability of finding another assembly with a lower \mathbf{P}_{bvp} within a neighborhood of k units decreases. However, it is insufficient to say anything about the non-uniform decrease in the probability.

5.4. Evolutionary dynamics

The evolutionary dynamics of the disassembiotic mechanism reveal some interesting properties of the landscape of ξ . Consider Figs. 13a and b. They show the variation of the population fitness of the evolving assemblies A and C. It can be seen that the average \mathbf{P}_{bvp} of the evolving population of assembly A converges very noticeably as individuals with lower \mathbf{P}_{bvp} are born into the population and come to dominate it. However, in contrast the evolving population of assembly C does not show this marked convergence and the trend towards convergence is much slower when compared to assembly A. A starting conjecture to explain this difference could be by attributing it reduced selection pressure due to the effect of increasing the number of parts. This is because with the reduction in the difference in the \mathbf{P}_{bvp} 's between assemblies of different equivalence classes the difference in the selection probability also reduces. As a result, the effect of an improvement does not propagate quickly through the population.

Now, consider Figs. 14 and 15 showing the variation in the average distance of the evolving population from the starting design and the variation of the average distance of the population from the best individual from the previous generation for assemblies A and C, respectively. The noticeable correlation between the two curves in Fig. 15 and the

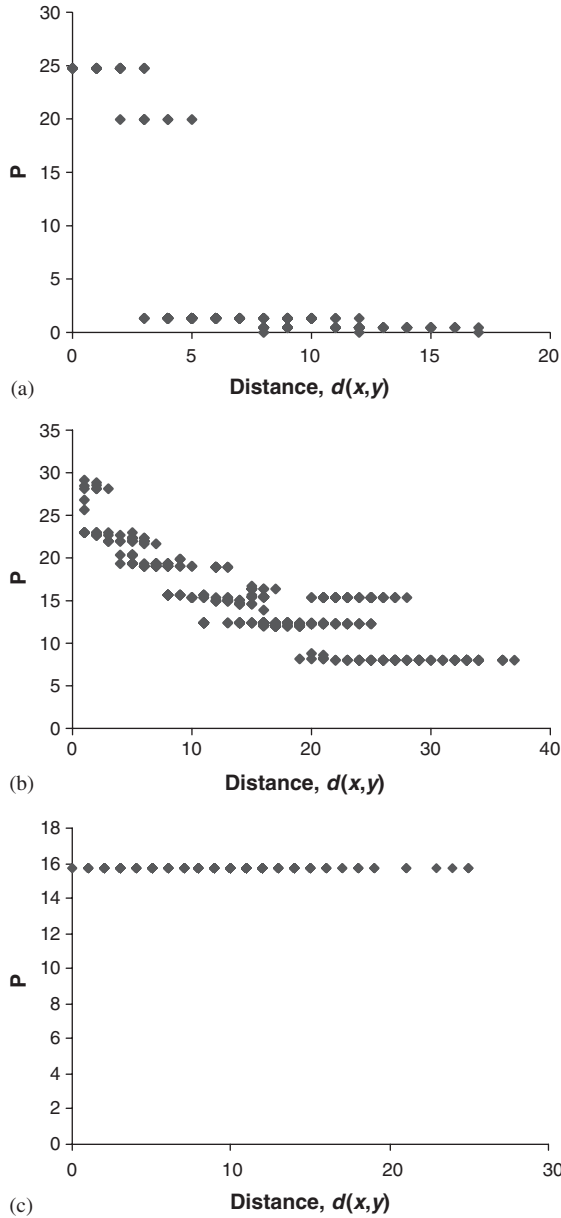


Fig. 12. Plot of P_{best} of each generation versus distance of best individual from starting assembly: (a) assembly A, (b) assembly B, and (c) assembly C.

lack of it in Fig. 14 show that the explanation of difference in selection pressure was too simplistic to explain the difference in dynamics between assemblies A and C as the reduction in the average distance of the population from the best individual show the emergence of a preferential search direction in the direction of areas having a lower height. This is evidence of a distinct selection pressure exerted by the individual(s) having low P_{bvp} 's on the population as in Fig. 15.

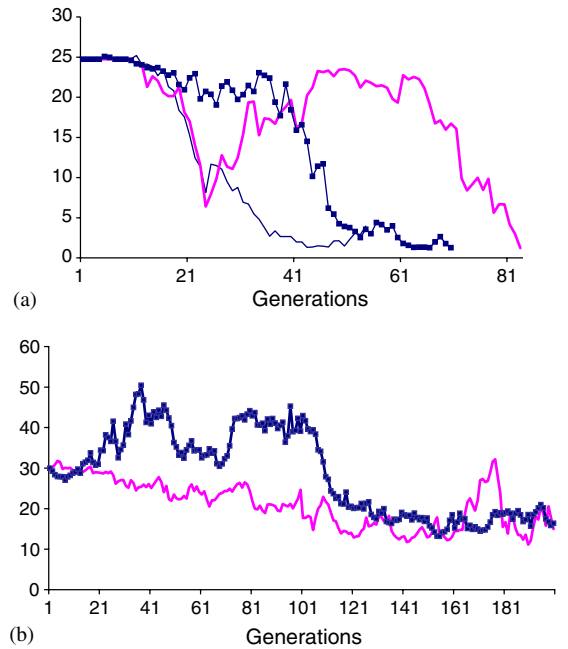


Fig. 13. Variation of average P_{bvp} of the population over different generations: — magenta — dark blue — light blue indicate different simulation runs: (a) assembly A and (b) assembly C.

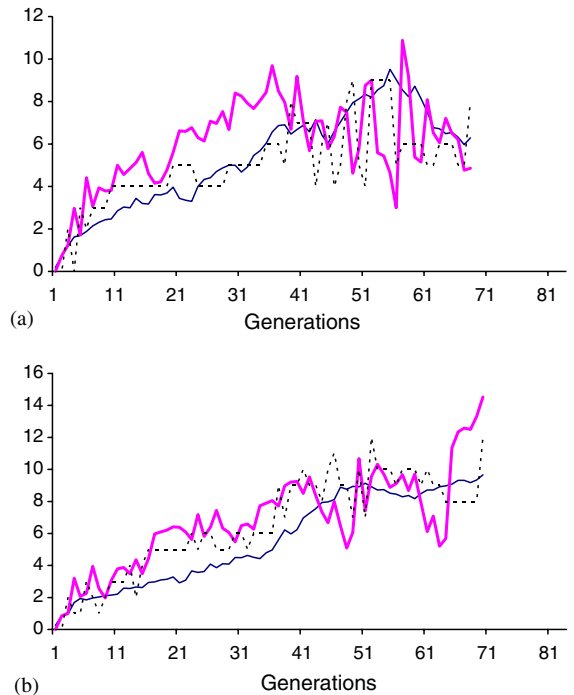


Fig. 14. Variation of distance in evolutionary dynamics of assembly A: — magenta — average distance of population from best individual in previous generation; — dark blue — average distance of population from starting assembly; — light blue — average distance of best individual from starting assembly. (a) Run 1 and (b) run 2.

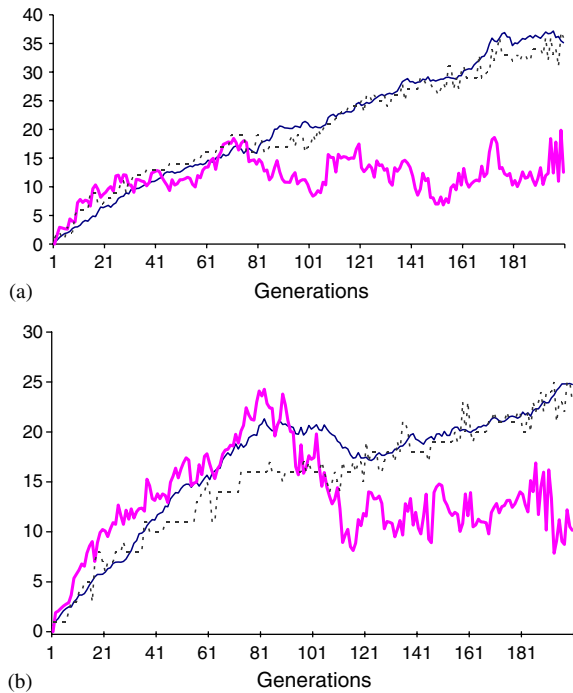


Fig. 15. Variation of distance in evolutionary dynamics of assembly C: —, average distance of population from best individual in previous generation; —, average distance of population from starting assembly; - - - -, average distance of best individual from starting assembly. (a) Run 1 and (b) run 2.

An auxiliary effect is that of the structure of the plateaus in assemblies *A* and *C*. The large size of the plateau in assembly *A* adds to the significant selection pressure due to the large differences in \mathbf{P}_{bvp} . As a result the population increasingly comes to contain a large number of individuals from this plateau. Due to the size of the plateau, the evolution tends to drift randomly and increasingly distant individuals on the plateau are selected in each successive generation and the emergence of a specific direction does not occur. In contrast, the plateaus in assembly *C* are smaller (so the distance between the individuals on the same plateau is not large) resulting in the emergence of a specific direction. However, due to the low selection pressure this direction emerges only after a number of generations have passed. Furthermore, due to the small size of the plateau when individuals are selected for reproduction and undergo mutation or crossover, the possibility that the resultant offspring belongs to another equivalence class is also higher. As the difference in \mathbf{P}_{bvp} is not large, the offspring only have a marginally higher \mathbf{P}_{bvp} than \mathbf{P}_{bvp} of the fittest individual in the population thus preventing the fitness of the population from converging quickly.

The contribution of the structure of the equivalence class in driving the evolutionary dynamics can be dramatically

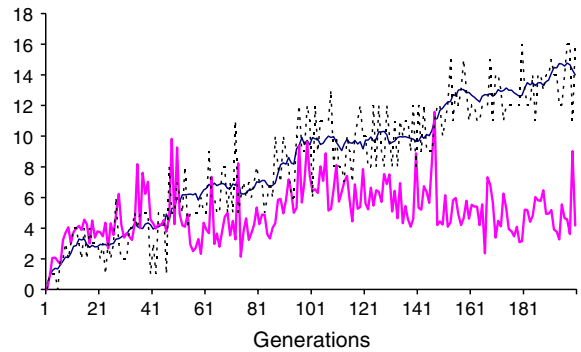


Fig. 16. Variation of distance in evolutionary dynamics of assembly *B*: —, average distance of population from best individual in previous generation; —, average distance of population from starting assembly; - - - -, average distance of best individual from starting assembly.

shown with assembly *B*. As mentioned earlier, the evolution of *B* does not result in any improvements so the question of selection pressure due to difference in \mathbf{P}_{bvp} does not arise. Yet, there is a noticeable emergence of a preferential search direction as can be seen in Fig. 16. The reason for this is easily visible. Parts 3, 6, 7, 8 and a portion of part 1 of the assembly *B* in Fig. 9b are similar to a line on the plane in that, with the exception of the ends, the removal of a point (cell) from anywhere in the line results in a violation of the connectedness requirement for the line. This greatly reduces the number of neighbors of assembly *B*. Further each of these neighbors lie on the same plateau. The relatively small size of the plateau and the absence of adjacent equivalence classes results in the emergence of a preferential direction, even in the absence of different \mathbf{P}_{bvp} 's. This example shows the complex implications that arise due to regularity. This dependence of the evolutionary dynamics on the structure of the assembly stresses the importance of methods specifically designed to search ξ with its unique characteristics rather than general domain independent techniques.

6. Discussion

A concern in the development of this model using a simple block world is its applicability to real-world assemblies. The central idea of the model rests on the existence of a differential representation of assembly geometry and the characteristics of the resultant optimization landscape and is not tied to this specific cellular representation. The simplifications are really in the CV model and the classes of assemblies that can be analyzed with this differential approach are dependent on the CV model. The CV model by itself is independent of the geometry of the assemblies as it is largely based on topological characteristics.

In the context of the possibilities of automated configuration optimization the findings of the model are telling. Despite their obvious utility in a redesign context, a

differential approach to automated optimization is inherently limited in serving as a panacea for configuration improvement. The findings that support this limitation are listed below:

- As the number of parts increases, the probability that a differential change in the geometry of an assembly would result in a large improvement in \mathbf{P}_{bvp} decreases rapidly.
- The stubborn intransigence of regular assemblies to improvement by differential change.

These results confirm the oft-mentioned guideline about the importance of making large-scale design changes especially those related to the configuration early in the design process.

The results also raise the question about the typical ways of approaching DfD. For example, a typical guideline in DfD is to reduce the number of parts. However, in many cases, an increase in product complexity results in an increase in the number of parts. So rather than neglecting this possibility, the proposed model could be used to find the meaning of modularity in a bottom-up way by investigating the implications of successively adding parts with arbitrary values to the assembly. It would be of great utility to see if there exist certain portions of the landscape that are less rugged than others. This could imply the existence of a class of assemblies that are modular not merely by construction but in behavior, in the sense that adding parts does not affect their disassembly characteristics significantly.

Another possible avenue that could be investigated using this differential approach is the robustness of the assembly to deformation. Since EOL disassembly is performed, as the name suggests, on products at the end of their useful life the state of the product after abuse by the user affects the product's disassemblability significantly. In terms of the differential model, the robustness of an assembly would be the extent of deformation that the geometry of the assembly can take while still remaining in the same equivalence class.

Another design trend that could benefit from this differential approach is that of mass customization and design for variety. With the knowledge of the structure of the equivalence classes, multiple geometries having the required desirable disassembly characteristics could be obtained.

In summation, it can be seen that there remain a large number of issues to be investigated to develop competent configuration optimization techniques for EOL disassembly. This model provides little in terms of an improved representation or an efficient algorithm but it is envisioned that the principles unearthed regarding the complexity of the geography of the optimization landscape would greatly help in the development of these computational aspects.

From the standpoint of extending the model, a specific area for future research is a rigorous investigation of the effect of regularity and associated geometric reasoning problems in order to estimate the improbability of assemblies

prior to optimization. Other issues include usage of techniques from the growing area of complexity theory [21] where sophisticated methods to study such landscapes are increasingly being developed.

7. Conclusions

In this paper a novel model to relate the geometry of an assembly and its effects on EOL disassembly as a result of its configuration is presented. Though leaving a number of questions unanswered, it is our belief that this model presents a significant effort in exploring the Pandora's box of automated configuration optimization methods for EOL disassembly. The model reveals some facets of the inherent complexity of such an undertaking but presents an optimistic avenue for the development of tailor-made algorithms to perform automated configuration optimization using the specific characteristics of the EOL disassembly domain.

Acknowledgements

This work has been supported in part by the US National Science Foundation Grants #BES 9727136 and DMI 9900226.

Appendix A. Proof for Lemmas 1 and 2

Let g be an assembly in ξ such that Ω_g has only one member say \mathbf{S}_g . From Lemma 1 it follows that all assemblies that are isomorphic to g also belong to ξ . Since there are m different parts, there are $m!$ ways of uniquely interchanging the part identities. Hence there are $m!$ distinct assemblies in ξ that are isomorphic to g , including g itself. As g permits only one sequence each of the assemblies that are isomorphic to g only have one sequence associated with them. Since these assemblies permit only sequence, this sequence is also their \mathbf{S}_{bvp} . Further, being distinct assemblies, no two of these isomorphic assemblies permit identical sequences. Ω has only $m!$ sequences so it follows that $[\mathbf{S}] \neq \{\emptyset\}$, for every \mathbf{S} in Ω .

The reason for basing this lemma on the existence of an assembly g having $|\Omega_g| = 1$ is because such an assembly cannot belong to ξ when no external disassembly constraints are defined i.e. when the assemblies are free-floating with no disassembly direction being blocked by an external obstacle.

Appendix B. Proof for Lemma 3

No two parts have the same value so there exist no sequence in Ω having $\mathbf{P} \geq \mathbf{P}_{\text{wvp}}$. Therefore, any assembly g having $\mathbf{S}_{\text{bvp}} = \mathbf{S}_{\text{wvp}}$ necessarily has only one sequence i.e.

$\Omega_g = \{S_{wvp}\}$. If this were not true and Ω_g were equal to say $\{S_{wvp}, S_a\}$, then S_a would be the S_{bvp} because P_a would be less than P_{wvp} , which contradicts the fact that the S_{bvp} of g is S_{wvp} .

So from Lemma 1, the number of members in $[S_{wvp}]$ is equal to the number of non-isomorphic assemblies in ξ that permit exactly one sequence (say L_1), i.e., $|[S_{wvp}]| = L_1$. From Lemma 2, each member of $[S_{wvp}]$ would have an isomorphic assembly belonging to every $[S]$, where $S \in \Omega$. Therefore, every equivalence class has at least L_1 members.

Consider the second worst sequence, S_{wvp-1} , which is necessarily a neighbor of S_{wvp} , i.e., $\delta(S_{wvp-1}, S_{wvp}) = 1$. As S_{wvp-1} and S_{wvp} are neighbors, a member x of $[S_{wvp-1}]$ can have $\Omega_x = \{S_{wvp-1}, S_{wvp}\}$ as the S_{bvp} of x would still be equal to S_{wvp-1} . Let L_2 be the number of assemblies in $[S_{wvp-1}]$ having $\Omega_x = \{S_{wvp-1}, S_{wvp}\}$.

Now, the equivalence class $[S_{wvp-1}]$ already contains L_1 assemblies. Therefore $|[S_{wvp-1}]| = L_1 + L_2$. The equality in this expression is because no member of $[S_{wvp-1}]$ can allow any other sequence than S_{wvp-1} and S_{wvp} .

Extending this reasoning, every S with the exception of S_{wvp} has at least one neighbor S' such that $P_S < P_{S'}$. Therefore, every $[S]$ except $[S_{wvp}]$ has at least one member x having $\Omega_x = \{S, S'\}$. From Lemma 1, it follows that there are at least L_2 such assemblies in every $[S]$ ($S \neq S_{wvp}$), for every S' associated with S . Hence, $|[S]| \geq L_1 + L_2 > L_1$ for every $S \neq S_{wvp}$. Therefore, $[S_{wvp}]$ is the equivalence class with the least number of members.

References

- [1] Navinchandra D. ReStar: a design tool for environmental recovery analysis. Proceedings of ICED 1993; 780–7.
- [2] Alting L, Legarth JB. Life cycle engineering and design. Annals of the CIRP 1995;44:569–79.
- [3] Boothroyd G, Alting L. Design for assembly and disassembly. Annals of the CIRP 1992;41:625–36.
- [4] Wittenberg G. Life after death for consumer products: design for disassembling. Assembly Automation 1992;12:21–5.
- [5] Jovane F, Alting L, Armillotta A, Eversheim W, Feldmann K, Seliger G. A key issue in product life cycle: disassembly. Annals of the CIRP 1993;42:651–8.
- [6] Keoleian GA, Menerey D. Sustainable development by design: review of Life Cycle Design and other approaches. Air and Waste 1994;44:645–68.
- [7] Zhang HC, Kuo TC, Lu H, Huang SH. Environmentally conscious design and manufacturing: a state-of-the-art survey. Journal of Manufacturing Systems 1997;16:352–71.
- [8] Gungor A, Gupta SM. Issues in environmentally conscious manufacturing and product recovery: a survey. Computers and Industrial Engineering 1999;36:811–53.
- [9] Homem De Mello LS. Sequence planning for robotic assembly of tetrahedral truss structures. IEEE Transactions on Systems, Man and Cybernetics 1995;25:304–12.
- [10] Homem De Mello LS, Sanderson AC. AND/OR graph representation of assembly plans. IEEE Transactions on Robotics and Automation 1991;6:188–991.
- [11] Baldwin DF, Abell TE, Lui MM, De Fazio TL, Whitney DE. An integrated computer aid for generating and evaluating assembly sequences for mechanical products. IEEE Transactions on Robotics and Automation 1991;7:78–94.
- [12] Wilson RH, Latombe J. Geometric reasoning about mechanical assembly. Artificial Intelligence 1994;71:371–96.
- [13] Viswanathan S, Allada V. Product configuration analysis to support redesign for end-of-life disassembly. International Journal Production Research 2001;39:1733–53.
- [14] Simon M. Design for dismantling. Professional Engineering 1991; 20-2.
- [15] Simon M, Dowie T. Quantitative assessment of design recyclability. Report DDR/TR8, Manchester Metropolitan University, UK; 1993; <http://sun1.mpcce.stu.mmu.ac.uk/pages/projects/dfe/pubs/dfe8/report8.htm>.
- [16] Ishii K, Lee B. Reverse fishbone diagram: a tool in aid of design for product retirement, Proceedings ASME design engineering technical conferences and computers in engineering conference, 96-DETC/DFM-1272; 1996.
- [17] Jovane F, Semeraro Q, Armillotta A. Computer-aided disassembly planning as a support to product redesign. In: Krause F-L, Seliger G, editors. Life cycle networks. London, UK: Chapman and Hall; 1997. p. 388–99.
- [18] Feldmann K, Meedt O. Determination and evaluation of the optimal end of life strategy for products based on simulation of disassembly and recycling. In: Krause F-L, Seliger G, editors. Life cycle networks. London, UK: Chapman & Hall; 1997. p. 249–50.
- [19] Zussman E, Kriwet A, Seliger G. Disassembly-oriented assessment methodology to support design for recycling. Annals of the CIRP 1994;43:9–14.
- [20] Minami S, Pahng KF, Jakiela MJ, Srivastava A. A cellular automata representation for assembly simulation and sequence generation. In: Proceedings IEEE international symposium on assembly and task planning, 95TB8123. 1995. p. 56–65.
- [21] Kauffman SA. The origins of order: self-organization and selection in evolution. New York: Oxford University Press; 1993.
- [22] Sebaaly MF, Fujimoto H. A genetic planner for assembly automation. In: Proceedings of the IEEE conference on evolutionary computation. 1996. p. 401–6.
- [23] Gen M, Cheng RW. Genetic algorithms and engineering design. New York: Wiley; 1997.
- [24] De Fazio TL, Whitney DE. Simplified generation of all mechanical assembly sequences. Journal of Robotics and Automation 1987;6:640–58.
- [25] Heemskerck CJM, Van Luttervelt CA. The use of heuristic in assembly sequence planning. Annals of the CIRP 1989;38: 37–40.
- [26] Delchambre A. Computer-aided assembly planning. London, UK: Chapman & Hall; 1992.
- [27] Arai E, Iwata K. CAD system with product assembly/disassembly planning function. Robotics and Computer Aided Integrated Manufacturing 1993;10:41–8.
- [28] Chen CLP, Wichman CA. A systematic approach for design and planning of mechanical assemblies. AI-EDAM 1993;7: 19–36.

- [29] Cho DY, Cho HS. Inference on robotic assembly precedence constraints using a part contact level graph. *Robotica* 1993;11:173–83.
- [30] Laperriere L, Elmaraghy H. GAPP: a generative assembly process planner. *Journal of Manufacturing Systems* 1996;15:282–93.
- [31] Gottipolu RB, Ghosh K. Representation and selection of assembly sequences in computer-aided assembly process planning. *International Journal of Production Research* 1997;35:3447–65.

Further reading

- [32] Harjula T, Rapoza B, Knight WA, Boothroyd G. Design for disassembly and the environment. *Annals of the CIRP* 1996;45:109–14.