Complexity Measurement Of Cyber Physical Systems

by

Gurpreet Singh

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved December 2011 by the
Graduate Supervisory Committee:

Jami Shah, Chair
George Runger
Joseph Davidson

ARIZONA STATE UNIVERSITY

May 2012

ABSTRACT

Modern automotive and aerospace products are large cyber-physical system involving both software and hardware, composed of mechanical, electrical and electronic components. The increasing complexity of such systems is a major concern as it impacts development time and effort, as well as, initial and operational costs. Towards the goal of measuring complexity, the first step is to determine factors that contribute to it and metrics to qualify it. These complexity components can be further use to (a) estimate the cost of cyber-physical system, (b) develop methods that can reduce the cost of cyber-physical system and (c) make decision such as selecting one design from a set of possible solutions or variants.

To determine the contributions to complexity we conducted survey at an aerospace company. We found out three types of contribution to the complexity of the system: Artifact complexity, Design process complexity and Manufacturing complexity. In all three domains, we found three types of metrics: size complexity, numeric complexity (degree of coupling) and technological complexity (solvability).We propose a formal representation for all three domains as graphs, but with different interpretations of entity (node) and relation (link) corresponding to the above three aspects. Complexities of these components are measured using algorithms defined in graph theory.

Two experiments were conducted to check the meaningfulness and feasibility of complexity metrics. First experiment was mechanical transmission

and the scope of this experiment was component level. All the design stages, from concept to manufacturing, were considered in this experiment. The second experiment was conducted on hybrid powertrains. The scope of this experiment was assembly level and only artifact complexity is considered because of the limited resources.

Finally the calibration of these complexity measures was conducted at an aerospace company but the results cannot be included in this thesis.

Dedicated to my father Sardool Singh, mother Harmeet Kaur and brother Dharamjeet Singh whose love and support made me what I am and let me pursue my dreams. Special mention to my roommates Sunil Soni, Bhushan Parikh, Siva Kareti Reddy and friends back home Nitin, Vipul, Casar, Ajay, Dabas, Depu, Sunil and Tanmoy on whom I can always count for support.

ACKNOWLEDGEMENTS

My most sincere appreciation and gratitude go to my committee chair and advisor, Dr. Jami Shah who has been a very patient and inspiring mentor and guide in my professional life. Without his contribution and support this work would not have been possible. I would like to give a special mention to Dr Joseph K. Davidson and Dr George Runger, who were always willing to spend time with me, answer my questions and helped me all along.

I would also like to thank students in the Design Automation Laboratory – Srinath Balaji, Prashant Mohan, Mahmood dinar, Zihan Zhang, Xiang Khe, Peng Zhao, Manikandan Mohan, Ying Chen, Yadong shen, Samir Savalia – for their assistance in research and help in writing technical reports, codes for the complexity software. They were great colleagues to have and were always ready to share their thoughts, think about and help me with my ideas.

A special thanks to Ronald Howard for his knowledge, software and research data on hybrid power-trains.

I wish to acknowledge the financial support for this work provided by DARPA, META II program.

TABLE OF CONTENTS

Page

CHAPTER

APPENDIX Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Background

Design is an iterative process of developing a solution to a given set of specifications that not only performs desired functions but also meets all the required objectives. In the past several decades new designs, technologies and equipment are being developed and providing a foundation for new research projects. The need of higher performance and lower maintenance in specification sets has leaded us to build more and more complex and complicated systems. As systems have become more complex they have become not only more expensive to develop, but also more difficult to predict accurate development costs. This situation leads to the over-budget and delay of the research projects. According to the US Government Accountability Office (GAO) a major research project in last decade, on average, costs 40% more than the initial estimated cost [1]. The study also shows that the average delay in delivering the final product was 22 months.

1.2 Problem statement

Since current aerospace and defense projects are getting more and more complex and over budget, metric needs to be developed that can be used for measuring complexity in large cyber-physical projects. The purpose of the complexity measures is to assist in choosing a design from several alternatives, identify sources of complexity and attempt to reduce or manage it.

Analysis on complex projects shows that the largest component of growth in the cost of design projects has come from increased complexity [1]. The ability of avoiding unnecessary complexity and to control necessary complexity becomes increasingly important in the product development and process management fields. Managing complexity in an intelligent and efficient way becomes a key factor to success. According to the GAO, the research projects that had excellent information ahead of research projects finish with spending 30 % fewer resources.

Keeping complexity under control can become a competitive edge. It will not only eliminate the need of expensive testing but also get rid of iteration of design process. Complexity can also help in setting up the curriculum for the courses by assigning the less complex problem earlier in the course and more complex problems in the end.

The aim of this research is to come up with a process that can measure complexity of cyber-physical systems. A large number of methods have been proposed for complexity measures in various domains but none of them gives true quantitative value of complexity [8]. Consequently there is a need of a procedure or technique that can provide exact quantitative value of complexity. The complexity metrics must have the following characteristics:

1) Holistic: The measure should be simple, easily interpreted, and can be applied to all type of products.
2) Correlate with reality: It is very important that the quantitative value of complexity metrics should be consistent with expert opinion and correlate

with reality. So if an expert considers a product more complex than other, then complexity measures must demonstrate the same results.

3) Consistent: The complexity measures should also be consistent at different levels of abstraction. For example, if the complexity metrics are giving results more to one design at component level then the results should not conflict at a higher level e.g. assembly level of abstraction.
4) Sensitive: Complexity metrics should be sensitive enough to show a difference in evaluations when there is a small change in design.
5) Repeatable: If two experts compute the complexity of the same product, then the complexity metrics results should be same.
6) Scalable: Complexity metrics should be able to estimate complexity at all stages of product development process.

In addition to these properties complexity metrics should have the following mathematical properties:

1. Monotonicity: According to this property, a complexity measure should go only increase or only decrease when specifications, components, variables are added and subtracted to a system respectively.
2. Transitivity: If the complexity result of project A is greater than B and project B is greater than C then complexity result of project A must be greater than project C.
3. Non-dimensionality: Complexity metrics should not have any dimension otherwise it would not be able to give a single value of complexity for

cyber-physical system i.e. systems consist of mechanical, electrical and electronics components.

4. Continuity: The response curve of complexity metrics should be a single unbroken curve and it should not have sudden jumps.
5. Invariant to affine transformation: According to this property complexity metrics results should not be affected by translation (addition of a number) and scaling (multiplication with a number).

1.3 Organization of the Dissertation

This dissertation is organized in 6 chapters. Chapter 2 includes literature on all the previous research performed related to the complexity measurement in all the domains.

Chapter 3 has the definitions and procedures for computing the unweighted complexity metrics. The details of first experiment, Mechanical Transmission, are also included in chapter 3. Design and implementation Matlab code for computing unweighted complexity metrics is given in chapter 3.

Chapter 4 contains the information regarding the second experiment “Hybrid powertrains” and definition of weighted complexity measures. The detail of the Matlab code and C++ code for weighted complexity measures is also included in fourth chapter.

In Chapter 5, methods of representation the graphs of multi-domain artifact are discussed. It also includes the process of calculating the interaction complexity of multi domain product. The information regarding the Matlab code and C++ code for functional graph representation and computing the interaction complexity are also included in fourth chapter.

The conclusion of the research and a projection of future work are discussed in Chapter 6. Appendixes contain the results tables, Matlab code of measuring complexity and graphs generated for 2 experiments.

CHAPTER 2

LITERATURE REVIEW

There are several opinions on complexity by researchers from different fields. Complexity can be defined as:

- Complexity is a measure of uncertainty in achieving the functional requirements [9]. In another words, complexity is a function of the relationship between design range and system range. For instance complexity is zero if the system range is completely inside the design range. Complexity is infinite is the system range is outside the design range.
- Complexity should include how the parts are assembled into the whole structure [25]. The complexity of the system is not same as the sum of the complexity of the parts of the system. This coupling between the parts leads to the view that complexity is not a simple additive property from the components to the assembly, but rather there are emergent properties that are only found collectively in the assembly.
- Design complexity is a function of information required pertaining to design [3]. Information can be described in terms of operations needed to satisfy the goals of a problem be that manufacture or design. As more information is required, the complexity increases. Therefore the best design is the one in which the probability of successfully achieving the required information is maximum.

2.1 Literature on complexity

A precise definition or quantitative value of complexity can be useful in minimizing the efforts and development time of a design process. A lot of research has been completed and some are still going on in various fields to find out the key measures of complexity. Some of the fields are information science, computer science, system science and engineering design & manufacturing. These complexity metrics and be further classified into the following categories:

2.1.1 Design Complexity

Dan Braha and Oded Maimon proposed that the information content required for designing a system can be determined by the size of vocabulary and length of design [3]. The size of vocabulary (η) for a design problem is given as

$$\eta = \rho + N$$

Where ρ is the total number of unique operators and N is the total number of unique operands. The length of the design problem (L) can be determined by summing up all the operators (N_1) and operands (N_2)

$$L = N_1 + N_2$$

The Information content (H) of a design problem is given by

$$H = L \cdot \log_2 \eta = (N_1 + N_2) \cdot \log_2(\rho + N)$$

The design process of any system develops at a stage by stage basis. The information content at each stage defines the design form at a particular level of abstraction. The abstraction level (A) at a desired stage can be determined from the information content at the initial stage (H*) and at the desired stage (H) as

$$A = \frac{H^*}{H}$$

Design effort (E) provides the measure of the mental ability required to solve a design problem. It is determined by dividing information content (H) to the abstraction level (A).

$$E = \frac{1}{A} \cdot H$$

The total time (T) required to complete the design can be calculated from the design effort by the formula

$$T = \left(\frac{1}{S \cdot A} \right) \cdot H = \frac{H^2}{H^* \cdot S}$$

Where S is the rate at which a designer makes the decision process 1 bit of information. Empirical studies show that the range of S for the normal human is between 5 and 20.

Shah and Summers proposed three measures of complexity i.e. size, coupling, and solvability [8]. The size measures of the problem, product and process by the formulae

$$Cx_{size\_prob} = \{(M^0 + C^0) \times \ln |idv + ddv + dr + mg|\}$$

$$Cx_{size\_product} = \{(M^0 + C^0) \times \ln |idv + ddv + dr|\}$$

$$Cx_{size\_process} = \{(M^0 + C^0 + P_{op}) \times \ln|idv + ddv + dr + a_{op} + e_{op} + r_{op} + s_{op}|\}$$

Where M^0 is number of primitive modules available in a specific representation, C^0 is number of relationship available between all available modules, idv is number of independent design variable, ddv is number of dependent design variables, dr is number of design relations, mg is number of measures of goodness, P_{op} is number of unique process type, a_{op} is number of analysis operators, e_{op} is number of evaluation operators, r_{op} is number of representation mapping operators, s_{op} is number of synthesis operators.

The coupling between the variables at multiple levels can be represented by using graphs based format. This method describes the problems and products by nodes that represent the task or physical component. The nodes are connected to each other with the relations (links) according to the dependencies between them. The relations are removed one by one until graph is fully decomposed. The total count of relations gives the quantitative value of coupling between the variables.

Solvability expresses how much effort is required to solve a problem. It also specifies the amount of uncertainty, unpredictability and lack of knowledge in untangling a problem. Following equations are used for measuring the solvability of a product, problem and artifact:

$$\mathrm{Cx}_{\mathrm{solvability}} = \sum(\mathrm{k1} \cdot \mathrm{a}_{\mathrm{op}} + \mathrm{k2} \cdot \mathrm{s}_{\mathrm{op}} + \mathrm{k3} \cdot \mathrm{e}_{\mathrm{op}} + \mathrm{k4} \cdot \mathrm{r}_{\mathrm{op}})$$

$$\mathrm{Cx}_{\mathrm{DOF\_prob}} = \sum \mathrm{DOF(idv)} + \sum \mathrm{DOF(ddv)} + \sum \mathrm{DOF(mg)} - \sum \mathrm{DOF(dr)}$$

$$\mathrm{Cx}_{\mathrm{DOF\_art}} = \sum \mathrm{DOF(idv)} + \sum \mathrm{DOF(ddv)} - \sum \mathrm{DOF(dr)}$$

Where k1, k2, k3, k4 are coefficient factors that account for different reasoning complexities of reasoning process, the domain knowledge that is available and requires for each operator, and the design skills for executing that operator. DOF represent the degree of freedom, a_{op} is number of analysis operators, e_{op} is number of evaluation operators, r_{op} is number of representation mapping operators, s_{op} is number of synthesis operators, , idv is number of independent design variable, ddv is number of dependent design variables, dr is number of design relations, mg is number of measures of goodness.

Haik and Yang classified the complexity into the following components:

a) Complexity due to variability: Variation in Design parameter makes the manufacturing challenging [21]. If the manufacturing process follows normal distribution function with mean (μ) and variance (σ^2) then complexity due to variance is given by

$$h(f) = \ln\sqrt{2\cdot\pi\cdot e\cdot\sigma^2}$$

b) Complexity due to vulnerability: It is consists of three parts which are mapping, sensitivity and dimension. The mapping part refers to the topological structure of the design matrix [A]. Design matrix relates the functional requirements {FR} of a design problem to the design parameters {DP} by

$$\{FR\}_{m\times 1} = [A]_{m\times p}\{DP\}_{p\times 1}$$

The sensitivity part refers to the magnitude and sign of the design matrix coefficients. The dimension part mentions the size of the design problem. Combination of mapping, sensitivity and dimension gives complexity due to vulnerability (C_V) which can be expressed as

$$C_V = \sum_{i=1}^{m} \ln[A_{ii}]$$

Where *m* is the number of functional requirements.

c) The complexity due to correlation between two DPs that are distributed as normalized bi-variate normal is given by

$$h(\phi(DP_l, DP_k) = \ln(2 \cdot e \cdot \pi\sqrt{1-\rho^2})$$

Where ρ is the correlation coefficient

Bashir and Thomas proposed that first functional decomposition of a design a problem should be performed that gives all the elementary functions of the design problem [6]. Then a functional tree is prepared from the elementary functions and process complexity (PC) is can be computed by

$$PC = \sum_{j=1}^{l} f_j \times j$$

Where *f j* is the number of function at level j and l is the number of levels.

Li Chen and Simon Li used incidence matrix to compute the coupling between the design components and design attribute [7]. Rows of the incidence matrix consist of all the attributes and column with all components. Value 1 in the matrix cells represents the dependency of the attributes and components and value 0 represents non-dependency as shown in the figure 2.1. The coupling strength between two components is given by

$$r_{col\_ij} = \frac{\sum_{k=1}^{m} \min(m_{ki}, m_{kj})}{\sum_{k=1}^{m} \max(m_{ki}, m_{kj})} \qquad i, j \in [1, n]$$

Where r_{col_ij} is the measure of coupling between ith component (column) and jth component (column). The numerator corresponds to the number of 1-1 matches and denominator corresponds to number of 1-1, 1-0 and 0-0 matches.

Attributes	**Components**						
	I	**II**	**III**	**IV**	**V**	**VI**	**VII**
A	0	1	0	0	1	1	1
b	0	0	1	1	0	0	1
c	1	0	0	0	0	1	1
d	1	1	1	0	1	0	0
e	1	1	0	0	0	1	0

Figure 2-1 Example of incidence matrix

Similarly the coupling strength between two attributes is given by

$$r_{row\_ij} = \frac{\sum_{k=1}^{n} \min(m_{ki}, m_{kj})}{\sum_{k=1}^{n} \max(m_{ki}, m_{kj})} \qquad i, j \in [1, m]$$

2.1.2 Information content and Entropy measures: Entropy measures are based upon the amount of uncertainty in achieving all the functional requirements. At the initial development stage of any product, there is a lot of uncertainty due to the lack of knowledge about the product's behavior. However at the final development stages the amount of uncertainty decreases and the probability of successfully achieving the desired results increases.

According to the Nam P Suh, information content (I), or entropy, is inversely proportional to the probability (p_i) of successfully achieving all the function requirements [9].

$$I = \sum_{i=1}^{n} \log(1/p_i) = -\sum_{i=1}^{n} \log(p_i)$$

Where, log is either the logarithm in base 2 (with the unit of bits) or the natural logarithm (with the unit of nats).

Shannon defines entropy in terms of possible events whose probabilities of occurrence are p_1, p_2… p_n [10]. The uncertainty of the outcome is given by the relation

$$H = \sum_{i=1}^{n} p_i \cdot \log(1/p_i) = -\sum_{i=1}^{n} p_i \cdot \log(p_i)$$

Where, H is the entropy of the set of probabilities p_1, p_2… p_n and must satisfy the following three assumptions:

1) H should be continuous in the pi.
2) If all p_i are equal, p_i = 1/n, then H should be a monotonic increasing function of n.
3) If a choice be broken down into two successive choices, the original H should be the weighted sum of the individual values of H.

Limitations of entropy measures: According to Shah and Runger, entropy measures do not possess proper mathematical properties and do not provide common sense measures of complexity because these are measure of designs goodness of fit [11]. The problem with Suhs entropy measure is that the complexity was defined as a measure of uncertainty in achieving the specified functional requirements. Thus a task is simple if all the functional requirements can be satisfied and vice versa. The ability of satisfying functional requirements is not same as the complexity of a design problem. It does not estimate the functional or physical structure of the artifact, i.e. how the design achieves the functional requirements. The limitation of Shannon's entropy measure is that it is a measure of data dispersion. The variation of the h-value depends upon σ and (b-a). The variation of h value for three type of Probability density function (PDF) is shown in figures 2.1 (a), 2.2(b) and 2.2(c).

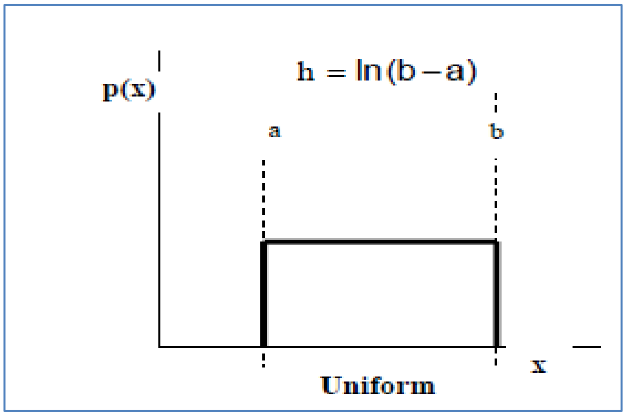


Figure 2-2(a) Differential entropy values h for uniform PDF [11]

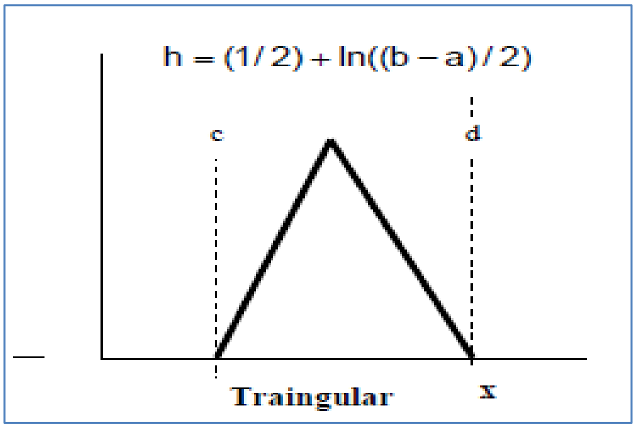


Figure 2-2(b) Differential entropy values h for triangular PDF [11]

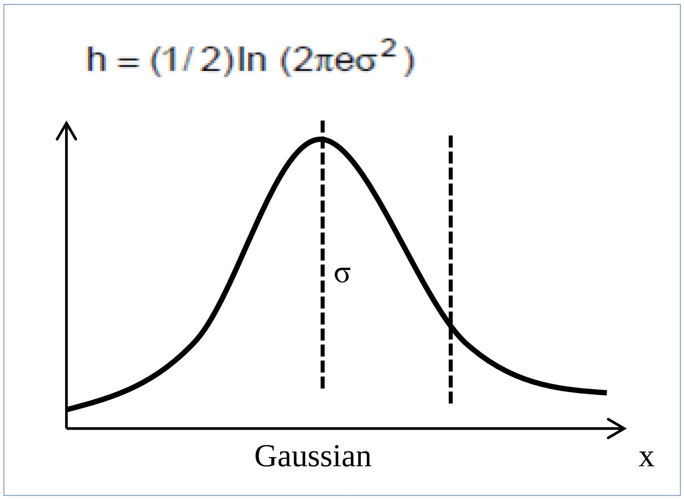


Figure 2-2(c) Differential entropy values h for Gaussian PDF [11]

In the triangular and uniform distribution, entropy depends only on (b-a) and it does not depends on the location of the distribution. Moreover, h for a Gaussian distribution is solely dependent on the variance and clearly has nothing to do with the mean.

A measure of dispersion or variance alone cannot even give us an indication of the goodness of fit of functional requirements. Another problem with this measure is that h has the dimension of log pi. Therefore, entropies from different pi's of interest cannot be added.

Meyer and Curley proposed a method to assess knowledge complexity [12]. It is assessed from a set of variables as given below:

i. Breadth of decision making domain
ii. Depth of decision making domain
iii. Rete of change of decision making domain
iv. Breadth of information inputs
v. Required information inputs
vi. Comprehensiveness of decision output

After assigning the scores to each variable, a weight factor is calculated for each variable. The score and weight for each variable are used to find out the Knowledge complexity (C_K) as

$$C_K = \left(\sum_{i=1}^{n} Score_i \times weight_i \right) / norm$$

Where n is the number of variables.

2.1.3 Complexity in computer science and information science

In computer science, the complexity of a program or software is estimated by total number of classes, number of objects, number of for loops and while loops, number of individual member and helper functions [4]. Every so often the total number of lines written for the program or software is also considered as a size measure.

Briand et al. (1996) has identified a framework for coupling complexity in software applications [5]. These coupling can be computed from:

1. Class linkage: it is the amount a class interacts with the other classes via interface and friend function.
2. Indirect coupling: When class A is connected to class B and class B to class C. Therefore class A is indirectly coupled to class C.
3. Inheritance: it is the property by which a base class is used as a foundation for the derived class.
4. Aggregation: It is the property by which an object contains the instance of other classes.

The field of mathematics plays vital role in computer science [40]. One of the important areas in mathematics is graph theory. It is used in modeling transport networks, activity networks and theory of games. Several algorithms have been developed to solve problems that are modeled in the form of graphs. Some algorithms are as follows:

1. Shortest path algorithm in a network
2. Algorithms for searching an element in a data structure e.g. DFS, BFS etc.
3. Algorithms to find the connectedness in a graph. Connectedness tells about densely a network is connected. It is expressed as the ratio of the number of connection to the maximum number of connection between the elements of a graph.
4. Algorithms to find the clustering present in the neighborhood of the element. The clustering of an element is determined number of connection between the neighbors elements to the maximum number of connection can exist between the neighbor elements.
5. Algorithms to find the cycles in a graph
6. Algorithms to find adjacency matrices (a matrix that shows the connectivity of the elements in the graph).

2.1.4 Complexity in manufacturing

Frizelle G measures manufacturing complexity by the progress of the part through manufacturing process and measures the obstacles that increase the lead time [22]. The formal definition of Kolmogorov-Saini entropy $h^{KS}(\mu)$ of process μ is given by

$$h^{KS}(\mu) = \lim_{n \to \infty} \frac{1}{n} H^{n}(\mu)$$

Where $H_n(\mu)$ is the joint entropy of the successive states forming a trajectory of the process μ from time 1 to n. It is computed as

$$H^n(\mu) = -\sum \mu(s_1^n)\log \mu(s_1^n)$$

$\mu(s_1^n)$ is the probability that a string $s_1^n = (s_{i1},.....s_{in})$ was observed in process μ between time 1 and n.

Calinescu et al. (1998) proposed a model for measuring manufacturing complexity on three assumptions [23]. First each subsystem is assumed to be an immigration-emigration process. Secondly, the more complex a process becomes, less reliably it will perform. Finally, the most complex are likely to be the bottlenecks. Based on these assumptions two fundamental types of manufacturing complexity are identified: Structural (static) complexity and dynamic (operational) complexity.

Structural (static) complexity arises from the effect of the product structure on the resources that will produce it and it is given by

$$H_{Static}(S) = -\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}\log_2 p_{ij}$$

Where M represents number of resources, N represents the number of possible states at resources, and p_{ij} is the probability of resource j being in state i.

Dynamic (operational) complexity determines the operational behavior from direct observations of the process, in terms of how queues behave. Dynamic complexity can be generated by internal sources (how well the facility is controlled) and by external sources (the effect of the customers and market). It is given by

$$H_{Dynamic}(S) = -P\log_2 P - (1-P)\cdot\log_2(1-P)$$
$$-(1-P)\left(\sum_{i=1}^{M^q}\sum_{j=1}^{N^q} p_{ij}^q \log_2 p_{ij}^q + \sum_{i=1}^{M^m}\sum_{j=1}^{N^m} p_{ij}^m \log_2 p_{ij}^m + \sum_{i=1}^{M^b}\sum_{j=1}^{N^b} p_{ij}^b \log_2 p_{ij}^b\right)$$

Where P represents the probability of the system being under control, p^q is the probability of having queues of varying length greater than 1, p^m is the probability of having length 0 or 1, p^b is the probability of having non-programmable states, M represents the number of resources and Nj represents the number of states at resource j.

Deshmukh presented another entropic measure for machined parts that is an aggregation over total number of operations associated with a part mix the number of parts to be concurrently produce in the manufacturing system during a particular schedule and total number of machines associated with a given part set [24].

$$Hp = -C\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{R}\sum_{l=1}^{N}\pi_{ijkl}\log\pi_{ijkl}$$

where C is a positive constant corresponding to the unit of measure, M represents the total number of operations associated with a part mix, and R represents total number of machines associated with the given part set. π_{ijkl} represent the processing time for operation i on machine k for part P_l.

Most of the literature on the manufacturing complexity suggests that complexity depends on the product features and manufacturing resources (particular process plan). Boothroyd et al. derived complexity metrics for assembly of manual, semi-automatic and automatic processes based on part count and assembly operations [25]. It was also proposed that in the assembly time can be minimize by reducing part count. However, this opinion does not account for the fact that fewer parts may become more complex to manufacture which results less assembly time but more manufacturing time [26]. The net result of this method could lead to more overall time and cost of manufacturing the product.

Many researchers proposed guidelines for Design for manufacturing (DFM). Banker suggested that the manufacturing complexity have a monotonic relationship with production cost which is function of parts attributes such as geometry features, internal and external undercuts, dimension tolerances, parting plane configuration, surface finish and texture [27]. Sors et al. computed the manufacturing complexity of plastic parts from the outer and inner surface complexities. A number between 0 and 5 is assigned to every feature (surface) and the aggregate of all the feature gives the value of manufacturing complexity [28].

Various studies measures implicit and explicit manufacturing complexity from part count, manufacturing time, produce-ability index, weight and lead time. Shah and Wright documented some measures that are employed in various DFM studies, system and tools [29].

Qualitative scores based on the Good practice rule: Specific rules are defined for domains which depend on manufacturing processes. These rules are then used to compare design specifications and process capability in terms of part count, geometric shape, tolerances, material and accuracy. A quantitative value is obtained by assigning scores to rules and weighing factors are used to compute complexity [30].

Direct cost estimate as a measure of measure of manufacturing complexity: Many handbooks [31, 32] and soft wares [33, 34] are available to make the cost estimate based on cost of material, feature size and shape, tolerances and surface finish. These methods only provide crude estimates because it does not account the process plan to be used for manufacturing.

Time based manufacturing rating: Elmaraghy proposed a metric termed design efficiency based on assembly time (t) and minimum part count (N_{MIN}) [35].

$$\eta = \frac{3 * N_{MIN}}{t}$$

Where multiplier of 3 is arbitrary.

1) Producibility assessment worksheet: The navy has defined a producibility matric based on 8 individual ratings (0-1) for key factor relevant to

module types [36]. For example, for mechanical modules the factors are: technical performance, design, cost, schedule, assembly process used, testing method and material/component availability. The risk value for each option is pre-specified. A score of 0.9 is assigned for completely automated assembly and 0.1 for completely manual assembly. Each design alternative is evaluated subjectively for the creation of entire range of PAWs and the scores are simply added.

2) Design tolerance to process capability ratio: Process capability index is given by

$$C_{pk} = \frac{(U - L)}{6\sigma}$$

Where U and L are upper and lower dimensional limits specified by design and σ is the standard deviation for the manufacturing process to be used. Process capability gives an indication of manufacturing difficulty and expected rejection rate. Its value should be 1 and values greater than 1 indicate poor manufacturability. The product of all acceptance rates for key parameters is used as DFM metric

$$\text{Overall DFM index} = \prod_{i=1}^{n} C_{pk}^{i}$$

3) DFM based on Taguchi loss function: Taguchi defines the quality as signal to noise ratio. A preferred design has lower sensitivity to uncontrolled variations. For tolerance design Taguchi used a parabolic function to model the loss of product quality with a design parameter drifts away from its target value. The loss Li. for parameter i is given by

$$L_i = (C_i /(U_i - L_i / 2)^2)(b^2 + \sigma^2)$$

Where Ci is the cost of failure, Ui and Li are the upper and lower limits of the parameter, b is the bias and σ is the standard deviation.

2.2 Evaluation of the complexity metrics: Table-2.1 shows what is measured by the proposed complexity metrics by the researches.

Table 2.1 Comparison of complexity metrics

Reference	**What is evaluated**	**What is measured**	**Metric type**	**Basis**
Braha and Maimon [3]	PB, PD	S	R	I
Shah and Summers [8]	PB, PD, PR	S, C, V	R	C, D
Haik and Yang [21]	PB, PD	S, V	R	I
Chen and Li [7]	PD	C	R	D
Nam P Suh [9]	PB, PD	V	R	I
Bashir & Thomas [6]	PB, PD	C	A	D
Meyer and Curley [12]	PB	S, V	A	I, C
Briand [5]	PD	C	A	C
Frizelle [22]	PR	S	A	D
Calinescu [23]	PR	C, V	R	I
Deshmukh [24]	PR	S, V	A	I
Boothroyd [25]	PR	S	A	D
Banker [27]	PD	S	A	D
Sors [28]	PD	S, V	A	D

Note:

What is evaluated—design process (PR), design problem (PB), and design product (PD)

Basis—computational/algorithmic analysis (C), information based (I), and traditional design (D)

What is measured—size (S), coupling (C), and solvability (V)

Metric—absolute measure (A) and relative measure (R)

From the table 2.1 it can be clearly seen that no proposed metrics, except Shah and Summers, give the estimate of complexity at all product development stages. Also complexity metrics only suggested by Shah and Summers include size, coupling and solvability at all product development stages. Therefore the complexity metrics are proposed by Shah and Summers are explored, subjected to experiments and modified in the next chapters so that they can provide the good estimate of complexity.

CHAPTER 3

UNWEIGHTED COMPLEXITY METRICS

3.1 Engineering Design

Engineering design is a systematic iterative procedure that involves three main steps: (a) Problem definition, (b) Solving design problem, (c) analyzing the solution against problem definition. Problem definition is a set of statements that contains all the function requirements, design objective and constraints. Design solution (or solutions) is then generated for the given problem definition by using the design knowledge and common knowledge. Design solution then analyzed and if it successfully achieves all the requirements of problem definition, it is sent to manufacture otherwise it is designed all over again. The design process includes the steps that are followed to find satisfactory solutions to the stated problem. Therefore it encompasses design solutions and analysis. Figure 3-1 shows the relationship between problem definition, design solution, analysis, design knowledge, common knowledge, design process and manufacturing.

From the figure 3-1 it can be inferred that in order to determine complexity of a design, the complexity of the each element of the tuple {Problem definition, Design process, Manufacturing} must be computed separately and then added together to get the overall complexity. Now the question is how to calculate the complexity of the individual element of the tuple.

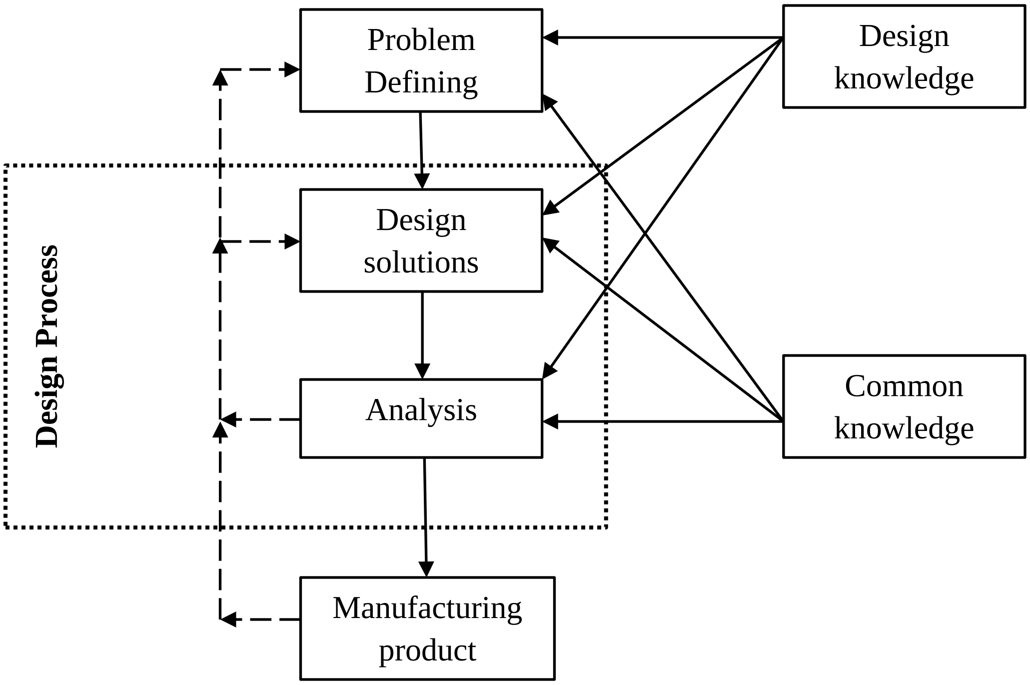


Figure 3-1 Flow chart of engineering design (modified from Ref. [8])

The complexity of the problem definition depends upon the number of functional requirements, constraints, variables etc. involve in a design. It also depends upon how these functional requirements, constraints, variables etc. are connected to each other. The problem is considered more complex if it is under-constraint i.e. number of variables is more than the number of equations.

The complexity of the design process depends upon the number of design processes, constraints, variables etc. It also depends upon how these design process steps are sequenced, how many iterations are there in the design process. The design process is considered more complex if there is no process that can solve the functional requirements.

Similarly, the complexity of the Manufacturing process depends upon the number of features a product has, number of manufacturing operations required and the sequence of manufacturing processes. The design process is considered more complex if it cannot be manufactured by standard manufacturing processes.

Based on these facts a common pattern can be find that can give the complexity of the tuple i.e. problem definition, design process, manufacturing process. This pattern consists of the following three characteristics:

a) **Size:** it can be predicted by simply identifying the total number of entities in each element of the tuple. These entities could be total number (#) of variables, # of parts, # of relations, # of logic decision, #of manufacturing processes.

b) **Coupling:** It is well recognized as one of the fundamental qualitative measure of complexity. It indicates how different entities (modules or parts) interact with each other. The more interaction between the parts results in the more coupling and therefore more complexity. While the size can be measure by just counting the elements, the coupling cannot be measure by using the same method. The coupling measure requires the representation of the tuple in a graph based format. Two methods of representing the coupling between the elements of the tuple are bipartite graph and line graph. These methods of representation are discussed in more detail in the section 3.2.

c) **Solvability:** It indicates how much it is difficult to solve the problem. A problem is difficult to solve when variables are not fully constrain. A well-defined problem is the one in which the number of unknown is equal to the number of equations. The complexity increases when variables are unconstrained i.e. the number of unknown is greater than the number of equations. These measures should be used with other complexity measures to assess complexity because they alone do not provide the full insight into the complexity.

3.2 Data representation scheme: Size can be predicted by simply identifying the total number of entities but coupling between the elements requires a data representation scheme that:

1. Can be stored in a compact format.
2. Can be easily formed and interpreted.
3. Can facilitate the computation of all complexity types i.e. size, coupling and solvability.
4. Can represent the data of all development stages i.e. problem definition, design process and manufacturing process.

In mathematics graphs are used to model relations between objects from a certain collection. The study of graphs is known as graph theory and is one of the key areas in mathematics. There are several types of graphs defined in mathematics, but two of the most common graphs used in mathematics for defining relationships between the objects are:

3.2.1 Bipartite graphs.

Bipartite graphs are used in many mathematical problems and matching problems in which a graph is divided into two sets. The relation between the elements of two sets is represented with a link. A link in bipartite can be established only between the elements of different sets. Design problem can be represent by using bipartite graph with variables/components of a system as first set, and type of relation between these variables/components as second set. An adjacency matrix of bipartite graph is a method of representing graphs in the form of matrix. It shows which elements are connected to other elements. The adjacency matrix of bipartite graph that has two sets, U and V, has the form

$$A = \begin{pmatrix} O & B \\ B^{\mathrm{T}} & O \end{pmatrix}$$

Where B is mxn matrix and O is a null matrix. m and n are the number of elements in set U and set V respectively. Figure 3-2 shows an example of bipartite chart.

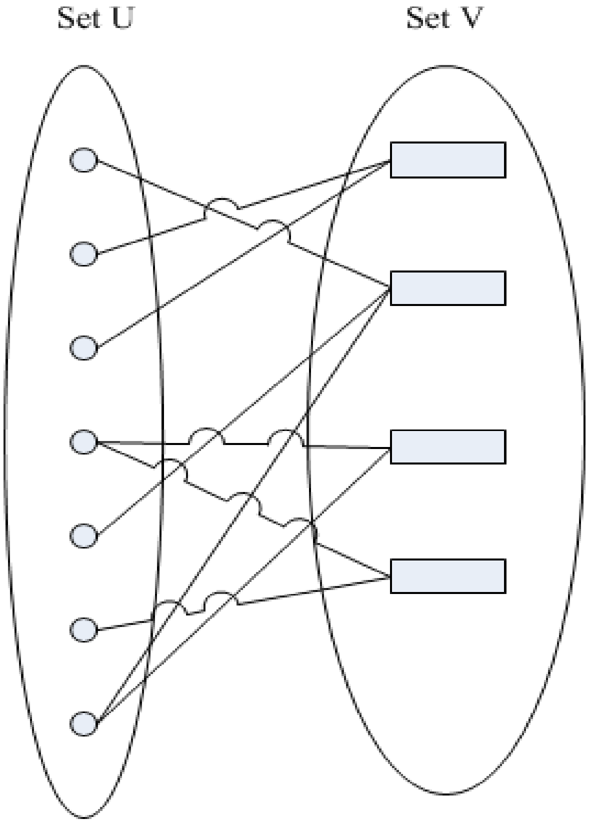


Figure 3- 2 Example of bipartite chart

Some important properties of bipartite graph:

- A bipartite graph can not contain any odd-length cycles. Therefore, a bipartite graph cannot contain a clique (a graph in which every two vertices in the subset are connected by an edge) of size 3 or higher odd sizes.
- If the two set of the bipartite graph has different elements then adjacency matrix of the bipartite graph is not a square matrix.
- All trees (graph that has no cycles) can be represent as bipartite graph [38].
- A graph is bipartite if and only if it is has two sets.

3.2.2 Line graph

A line graph is a point set consisting of finite number of points, or vertices, and a finite number of arc or edges which do not intersect, connecting pair of these points [38]. The adjacency matrix of a linear graph is a square matrix with number of rows and columns equal to the number of vertices in the graph. Figure 3-3 shows an example of line graph.

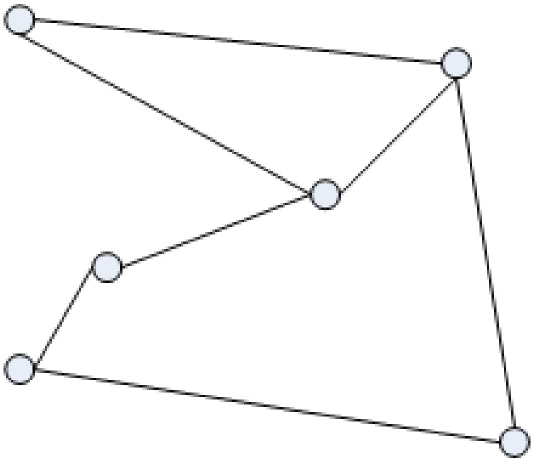


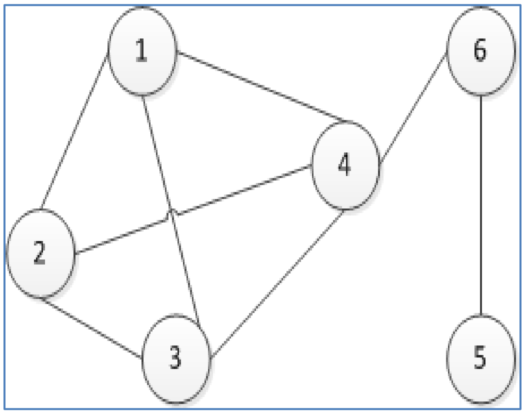
Figure 3- 3 Example of line graph

Some important properties of linear graph:

- It is possible to generalize line (network) graphs to directed graphs or undirected graph.
- The line graph can contain both odd and even number of cycles.
- Trees can also be represented using line graphs.
- Line graph can represent relationship between any numbers of sets.

Since line graph are all-purpose graphs i.e. it can be generalized to directed graphs or undirected graph and can represent any number of cycles and sets, it was selected over bipartite graph for the representation of the tuple.

3.2.3 Unweighted Adjacency matrix: it is a mathematical scheme of representing the graphs. Two nodes are called adjacent when they are connected to each other by a link. The adjacency relation (a_{ij}) is specified by value 1 if two nodes are connected and value 0 if the nodes are not connected. The matrix containing all adjacency relations in a graph is called adjacency matrix [14]. If n is number of nodes then size of the adjacency matrix is (n x n). Undirected graphs have adjacency matrixes that are symmetrical with respect to their main diagonal. On the other hand the adjacency matrixes of directed graphs are not symmetric. The example of an adjacency matrix for a random undirected graph is shown in figure 3.15. This scheme of representing the graph is easy to interpret, very compact and can be used for all development stages. This method satisfies all the requirements of data representation and therefore used to represent graphs for all development stages.



0	1	1	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	0	1
0	0	0	0	0	1
0	0	0	1	1	0

Figure 3- 4 Adjacency matrix for a random graph

Three types of graphs are required to find out the complexity of a system. The name of the graphs for the problem definition, Design process, and manufacturing process were given as artifact functional structure (Af), Design process structure (Di) and Manufacturing process structure (Mi) respectively.

3.3 Artifact Function Structure Complexity A_f: All the relationships (formulas) and variables are identified and line graph are created for the four alternatives. In the function structure (simple network graph), design variables are represented as nodes. If a design variable is related to another variable by a functional relationship, the corresponding nodes are connected by a link. A function graph does not show the direction of dependence and all links are assumed to have equal weight. Functional graph for all the alternatives are included in Appendix-A.

3.3.1. Artifact Functional structure measures

- The size of the design problem can be measured by the number of design variables (size of graph = **N**).
- Measures of coupling of the design problem can be measure by:

a) **Total number of links L** (order of graph) in the graph.

b) **Average node degree (a_{avg})**: Node degree (a_i) is number of nearest-neighbors of a node. The average node degree is calculated as a ratio of sum of all node degree to the total number of nodes.

$$a_{AVG} = \frac{\sum_{i=1}^{N} a_i}{N}$$

Where (a_i) = Node Degree (the number of the links on node 'i')

c) **Maximum node degree:** it is maximum number of links connected to a node in a graph.

d) **Connectedness (graph density):** it is a normalized measure of connectivity (range 0 to 1) and can be calculated as:

$$C = \frac{2L}{N(N-1)}$$

e) **Information content of the node degree:** it is a modified form of Shannon's Information content in which probability is replaced by node degree of nodes. It is calculated as

$$I_{VD} = \sum_{i=1}^{V} (a_i) \log_2 (a_i)$$

f) Normalized Information content of the node degree: It is defined as the ratio of Information content of the node degree to the Information content of the complete graph (maximum node degree).

$$NI_{VD} = \frac{I_{VD}}{(I_{VD})_{\max}}$$

where $(I_{VD})_{\max} = N.(N-1)\log(N-1)$

g) Average Clustering coefficient (CAVG): it is a measure of interaction among the set of nodes in the graph.

$$C_{AVG} = (\sum_{i=1}^{N} C_i)/N$$

Where Ci is the clustering coefficient and is defined as the ratio of the number of links Li between the neighbors of the node i, and the maximum no of links Li (max) =k (k-1)/2, in a complete graph that can be formed by the nearest neighbors (k) of that node i.e. $C_i = \frac{2L_i}{k_i(k_i-1)}$.

h) Bridge link: it indicates the decomposability of a graph. A Bridge link connects two different sub graphs of a structure. On removing a bridge link, the graph will split into two sub graphs. However, not all bridge links are equally important; the ratio of the size of the resulting sub-graphs indicate the importance. The importance can be found by a normalized measure N/(N-N1) where N1 is the size of the smaller sub-graph and N is the size of the uncut graph.

i) Mobility: It is the “degree-of-freedom” of the structure. It is a measure of how much the structure is over or under-constrained. The formula for mobility calculation is $M = 3(L - N - 1) + N$

3.4 Design Process Complexity p_i

The Design Process Structure represents design tasks, their sequence (process flow) and design iterations. Although functional structures depend only on the artifact, process structures depend both on the artifact and the designer. This is because designers or design organizations will not follow the exact same process for designing the same type of artifact. Thus, process structures are not unique. While the function structure is represented as an undirected graph, the process structure is a digraph.

While creating the process structure for this experiment it was assumed that all design tasks are executed by the same person. Therefore parallel working in the process and different skill levels of the designer was not taken into account.

All design tasks were performed in a manner they preferred (textbook procedures; novice designers), recorded the sequence, timing and number of iterations needed in the sub-problem execution. The design process for each of the four alternatives was represented as a graph.

3.4.1. Design Process measures

The nodes in the process structure graph represent design or analysis subtasks and the links represent information flow from one subtask to another. Design tasks are related when one or more variables output from one are used as input to another subtask. The definitions of size, coupling and decomposability measures for process structures are the same as those for artifact function structures. But the nodes and links now have different meaning.

One additional measure i.e. development Time is added to design process complexity measures. It is the sum of time taken by all subtasks.

3.5 Manufacturing Process Complexity M_i

Manufacturing complexity should correlate to manufacturing cost and time. It depends on all operations needed to manufacture and assemble a product. Many organizational factors, scheduling considerations, manufacturing location (offshore or low cost regions), suppliers' involvement, etc. can have a major influence, but these were not taken into account in the measuring manufacturing process complexity. Only artifact intrinsic factors are considered in our analysis. Another important consideration is batch size. It was assumed that the production run is 100 parts for each alternative.

Since the knowledge and experience of designers are not same, different process planners or manufacturing shops would create different manufacturing plans. It also depends on resources available of the shops involved. So manufacturing process plans also are not unique. The manufacturing plan can also be represented also as digraphs.

3.5.1. Manufacturing Process measures

The nodes represent each manufacturing operation and links represent the sequence of operations performed on the same part or material flow. All graph measures listed in artifact functional complexity have the same implication and technique to compute manufacturing process complexity. Additionally, total manufacturing time and cost were also computed. The AM Cost Estimator was used to guide the calculation of cost of operations.

3.6 Experiment 1: Mechanical transmission

3.6.1. Objective of experiment: The complexity metrics explained in the previous section was based on intuition and speculation. In order to find that the proposed complexity metrics can be used for cyber physical system and which complexity metrics are following the expert's perception and cost, it should be test through an experiment. Therefore a synthetic problem i.e. mechanical transmission was set up to evaluate proposed complexity metrics. The reason of selecting the mechanical transmission is that it is a routine design problem and the scope of the experiment is component level. Moreover all the development stages, from concepts to detail manufacturing plans, are considered in this experiment.

3.6.2. Design specifications: The design specifications of first experiment are:

i. Design a mechanical transmission with reduction ration 20:1

ii. The input and output shaft are perpendicular to each other i.e. the angle between the input and output shaft is 90 degree.

iii. The power at the input shaft is 5 Hp at 2000 rpm. Since the reduction ratio is 20:1, the rpm of the output shaft is 100 rpm.

iv. The shafts can rotate in both directions.

v. The design should maximize the efficiency and deliver low cost, heat generation, noise and total volume.

3.6.3. Alternative designs: Four alternative designs were created that meet the requirements of reduction ratio, shaft angle and input Hp. The conceptual layouts are given below along with their advantages and disadvantages.

1. **Alternative 1- worm wheel drive (single stage drive):** it is a gear arrangement in which a worm (screw) meshes with a wheel (helical gear). The conceptual layout of worm wheel drive is shown in figure 3-4.

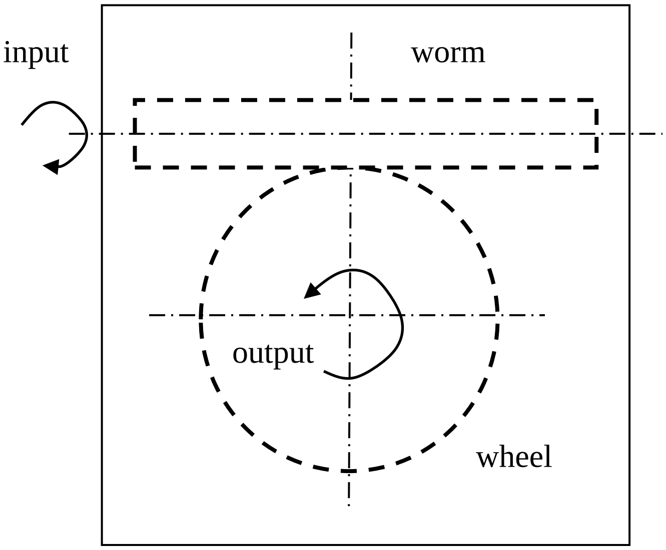


Figure 3-5 Conceptual layout of worm wheel transmission [13]

Advantages of worm wheel drive:

- It allows high gear ratio (up to 80:1) with shafts intersecting at 90 degree.
- The volume of the worm wheel drive is smallest for high gear ratio.
- It generates less noise during running because of constant meshing between worm and wheel.

Disadvantages of worm wheel drive:

- A lot of heat loss occurs due to sliding between the faces of gears. This heat loss makes this drive less efficient.
- Since the normal of contact surfaces is at an angle to the shaft axis, both radial and axial loads are present. Therefore both radial and axial loads are required.

2. **Alternative 2- Bevel pair and spur pair (two stage drives):** In this drive bevel gears change the direction shafts by 90 degree and also provide the gear ratio of 4:1. Spur pair only provide the gear ratio of 5:1. The conceptual layout of bevel pair and spur pair drive is shown in figure 3-5.

   Advantages:

   - Because both pairs have of the straight teeth, less heat is produced which results in high efficiency.
   - For spur gear only radial bearings are required. But for the bevel pair both radial and thrust bearings are required.

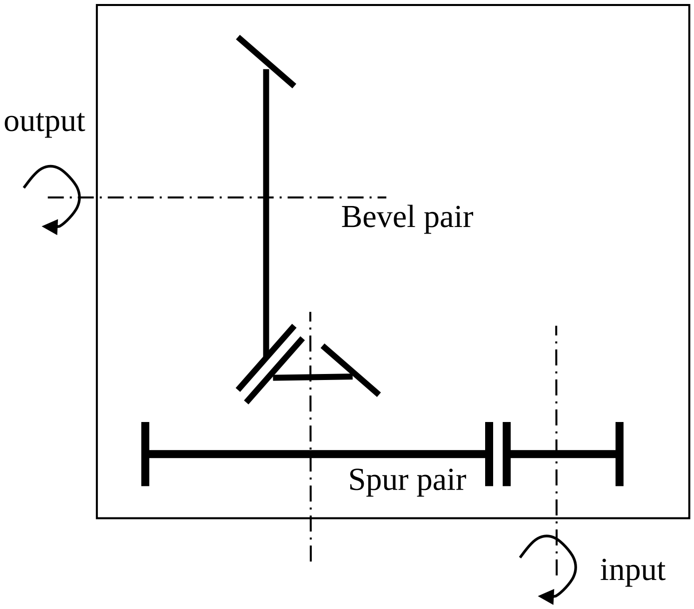


Figure 3-6 Conceptual layout of bevel pair and spur pair transmission [13]

Disadvantages:

- Because both gears are not in constant meshing this drive is very noisy at high speed.
- This drive requires more volume than alternate 1 because it had two gear pairs.

3. **Alternative 3- Face pair and spur pair (two stage drives):** In this drive face gears change the direction shafts by 90 degree and also provide the gear ratio of 4:1. Spur pair only provide the gear ratio of 5:1. The conceptual layout of face pair and spur pair drive is shown in figure 3-6.

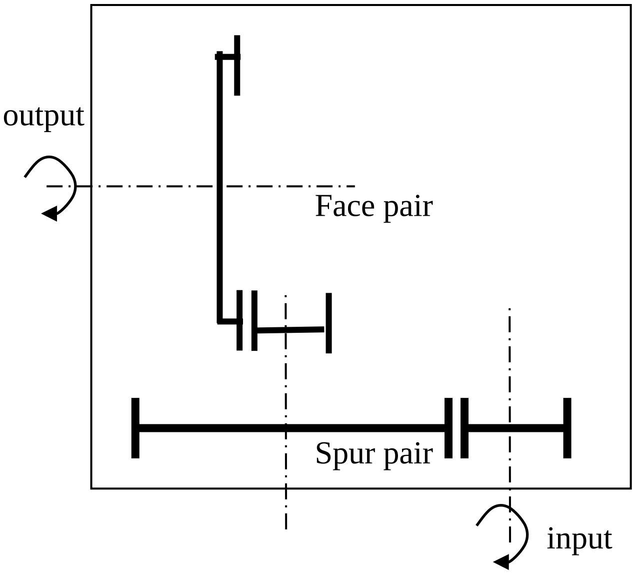


Figure 3-7 Conceptual layout of face pair and spur pair transmission [13]

Advantages:

- All the gears are either Spur or equivalent of a spur gear. It is easy to design and manufacture this transmission as compared to the bevel-Spur gear or the Hypoid-Helical gear transmission.
- Heat losses in both pairs are very small compared to worm gears which make this transmission more efficient.
- Thrust bearings are not required for spur gear drive.

Disadvantages:

- Because both gears are not in constant meshing, this transmission is very noisy compared to all other alternatives.
- This transmission is much larger than Alternatives 1 and 2.

4. **Alternative 4- Hypoid pair and helical pair (two stage drives):** In this drive Hypoid gear drive change the direction shafts by 90 degree and also

provide the gear ratio of 10:1. Helical gears pair only provide the gear ratio of 2:1. The conceptual layout of hypoid gear and helical gear drive is shown in figure 3-7.

Advantages:

- Hypoid gear is used where high reduction ratio (up to 15:1) and torque is required.
- Both hypoid and helical gear train are less noisy compare to spur and bevel gear train.
- Efficiency of this transmission is less than the bevel and spur gear but more than the worm wheel gear.

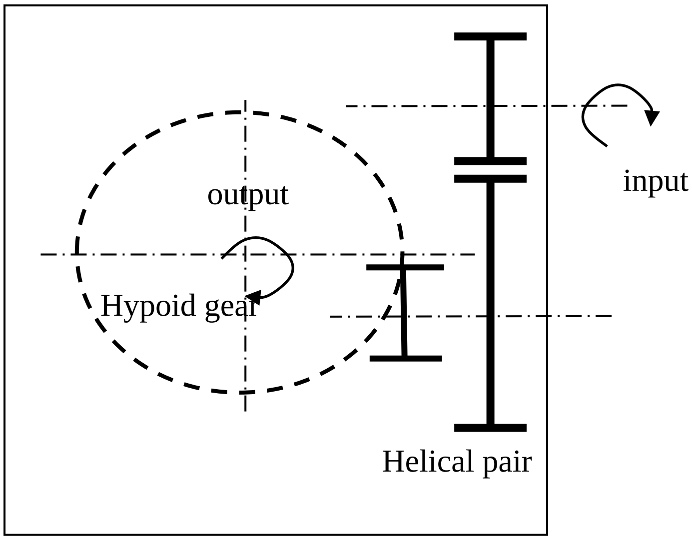


Figure 3-8 Conceptual layout of Hypoid pair and helical pair transmission [13]

Limitations:

- Both hypoid and helical gear train are difficult to manufacture because of complicated tooth profiles.
- Both hypoid and helical require radial and thrust bearings

Detail designs: All the parameter and design variables are computed using standard formulas and detailed CAD model of alternative drives along with all components such as shafts, keys etc. were prepared. The housing was not designed – it is only shown as a simple box. Bearings types were determined but COTS were not selected. The CAD models for all four designs are given below in figure 3-8, 3-9, 3-10 and 3-11.

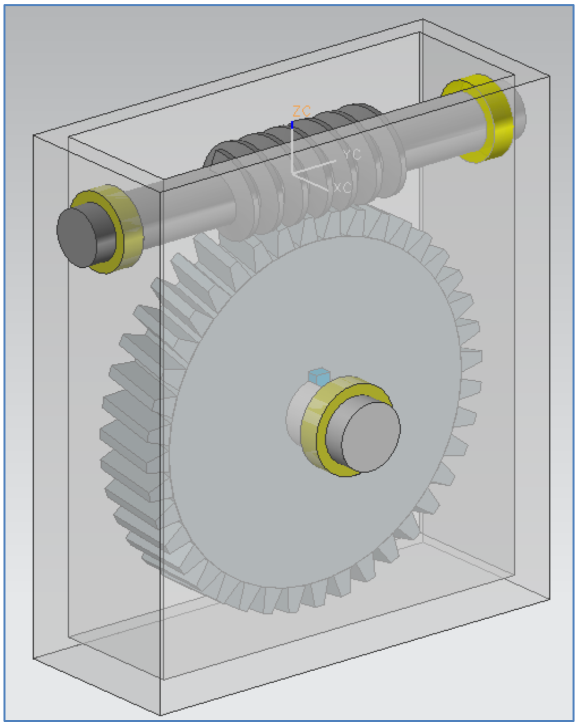


Figure 3-9 CAD model of worm wheel transmission [13]

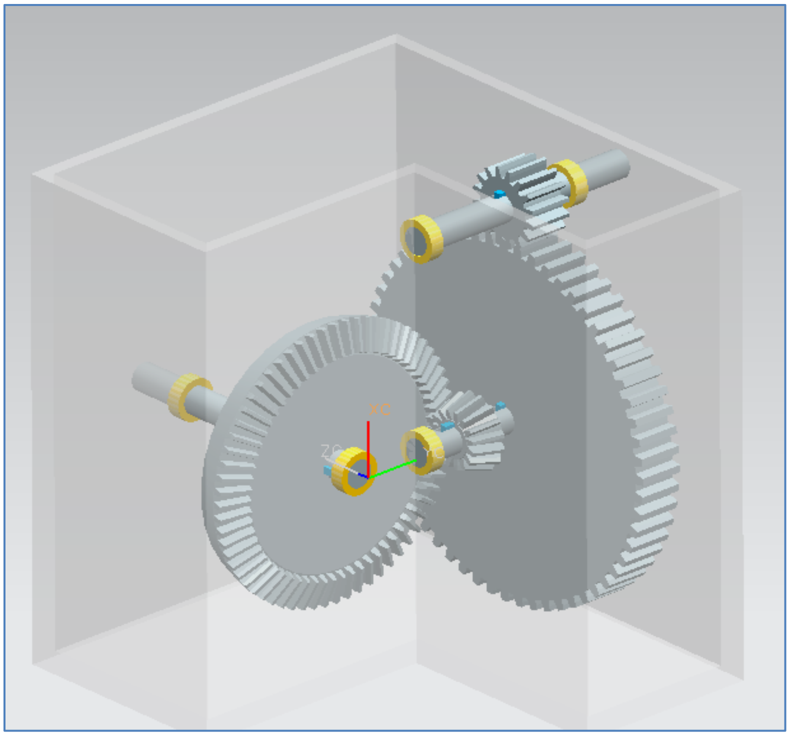


Figure 3-10 CAD model of bevel pair and spur pair transmission [13]

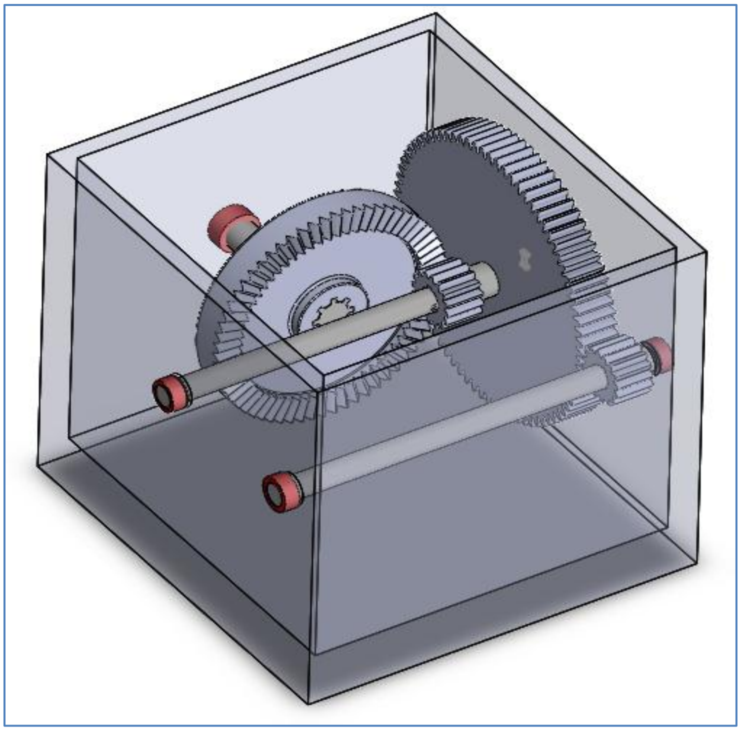


Figure 3-11 CAD model of face pair and spur pair transmission [13]

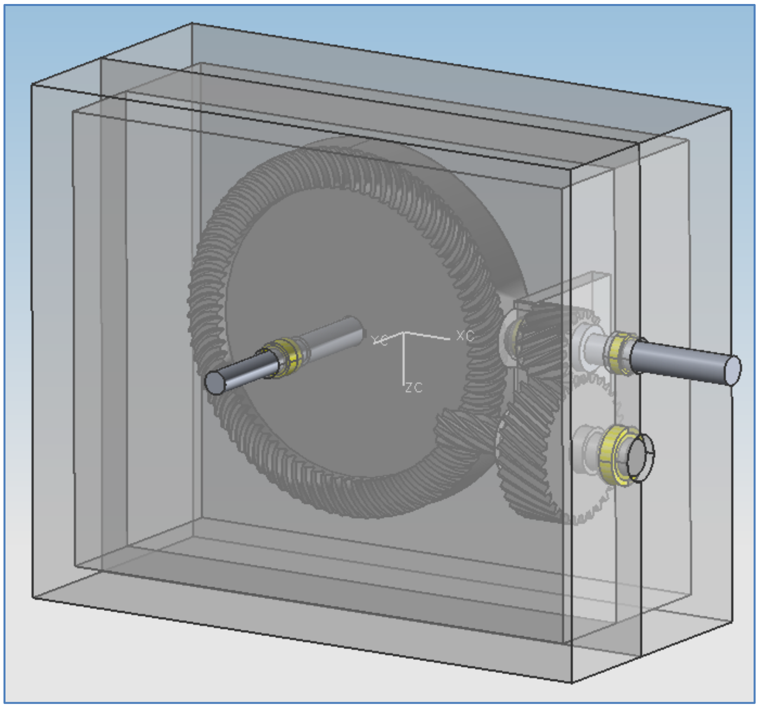


Figure 3- 12 CAD model of hypoid pair and helical pair transmission [13]

Four alternative designs, all capable of meeting the specified requirements, were generated. From the rationale it can be estimate that the simplest design was worm wheel gear box and the most complex design was two stage-hypoid and helical gear set. The complexity of other two remaining designs, two stage-bevels and spur gearbox and two stage-face and spur gearbox were in between the worm- wheel gearbox and hypoid-helical gear box. The reasons why these designs were selected for the experiment are as follows:

1. To check whether complexity measure can show the difference in the results according to the estimated intricacy of the design.
2. These designs will ensure that the comparison is being done between same kinds of design for a given set of specification. For instance worm gears are compared to helical gears.

3. These designed are created using well-known procedures and manufactured by standard processes. Therefore, uncertainty and unanticipated interactions were not present in this problem space.

The problem definition can be explained using artifact functional structure (Af) and artifact physical structure (Ap). In artifact functional structure all the relations between the design variables are used to create a graph. Artifact physical structure contains the connectivity information between the components of design solutions. However the information due to artifact physical structure is already present in the artifact functional structure, therefore a separate graph for artifact physical structure is not required in this study.

The design process complexity depends upon intrinsic design process structure (D_i) and extrinsic design process structure (D_e). The extrinsic design process complexity is introduced by organizational factors and choices made by the manufacturers, therefore not included in this study.

Similarly Manufacturing complexity depends upon intrinsic manufacturing structure (M_i) and extrinsic manufacturing structure (M_e). The extrinsic manufacturing process complexity is also introduced by organizational factors and choices made by the manufacturers, therefore it is also not included in this study.

The contribution of the remaining three components to complexity is not same. So weights should be applied to these complexity components and the equation of the overall complexity can be written as

$$\text{Complexity} = (w1)(Af) + (w2)(Di) + (w3)(Mi) \quad (1)$$

where w1, w2 and w3 are the weights assigned to artifact functional structure, design process structure and manufacturing process structure respectively.

3.6.4. Artifact functional structure results: Size, coupling and decomposability quantifiers were calculated from the Functional structures in Appendix B. The Table 3.1 below shows the results, which are also shown graphically for comparison.

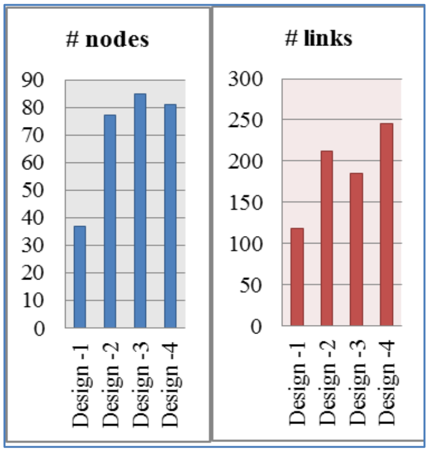
Table 3-1 Complexity metrics results for Artifact functional structures [13]

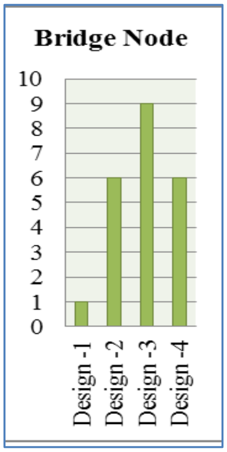
Metrics	Design1	Design2	Design3	Design4
# nodes	37	77	85	81
# links	118	212	185	246
# Bridge links	19	43	32	53
Average node degree (a_{AVG})	6.29	5.45	4.34	5.93
Max node degree (a_{MAX})	12	16	16	17
Connectedness	0.17	0.07	0.052	0.076
I_{VD}	654.1	1115.9	877.8	1378.1
N I_{VD}	0.095	0.030	0.019	0.033
Clustering coefficient (C_{AVG})	0.33	0.39	0.47	0.41
Mobility	277	479	382	573

3.6.5. Functional structure comparison charts

Size measures

Decomposability measures





Coupling measures

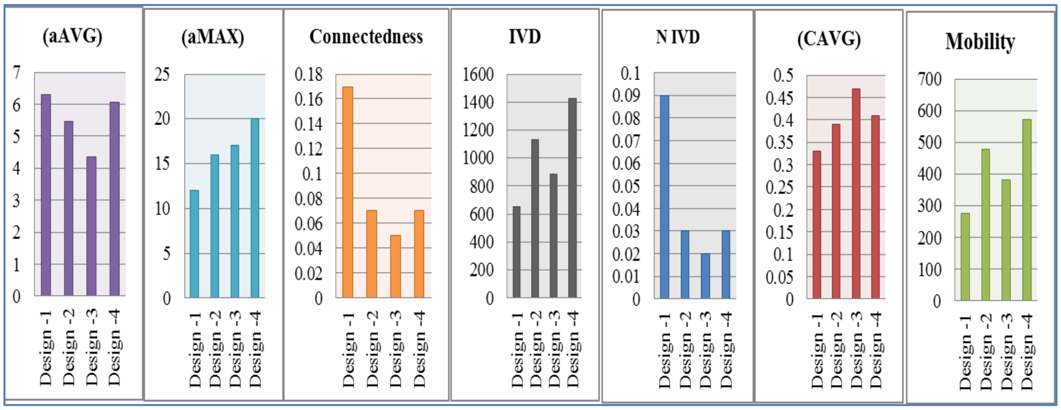


Figure 3-13 Bar charts for the size, decomposability and coupling measures of functional structure [13]

3.6.6. Design Process results: Appendix B contains the details of Process Structures for the four design alternatives. The complexity metrics results for design process structures are summarized in the Table 3-2 and Charts below.

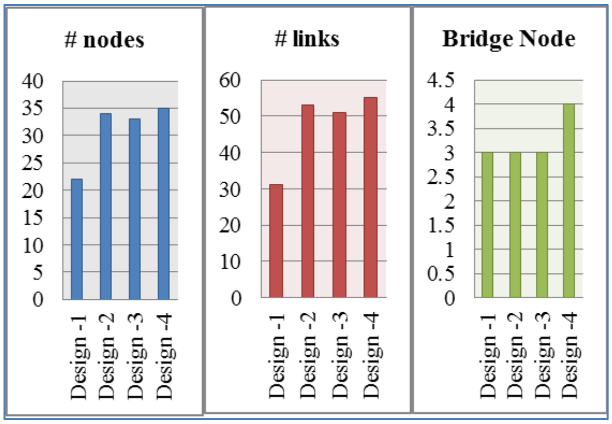
Table 3-2 Complexity metrics results for design process structures [13]

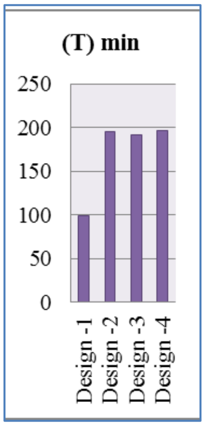
Metric	Design1	Design2	Design3	Design4
# nodes	22	34	33	35
# links	30	52	51	53
# cycles	11	23	23	23
#bridge links	2	4	4	4
Average node degree (a_{AVG})	2.78	2.94	2.97	2.92
Max Node degree (a_{MAX})	7	7	7	7
Connectedness	0.12	0.07	0.08	0.07
I_{VD}	107.40	197.02	195.02	199.78
N I_{VD}	0.04	0.02	0.02	0.02
Clustering coefficient (C_{AVG})	0.61	0.59	0.59	0.60
Total Process Time (min)	99	195	192	196
Mobility	46	88	84	92

3.6.7. Design Process Comparison Charts

Size measures

level of effort





Coupling measures

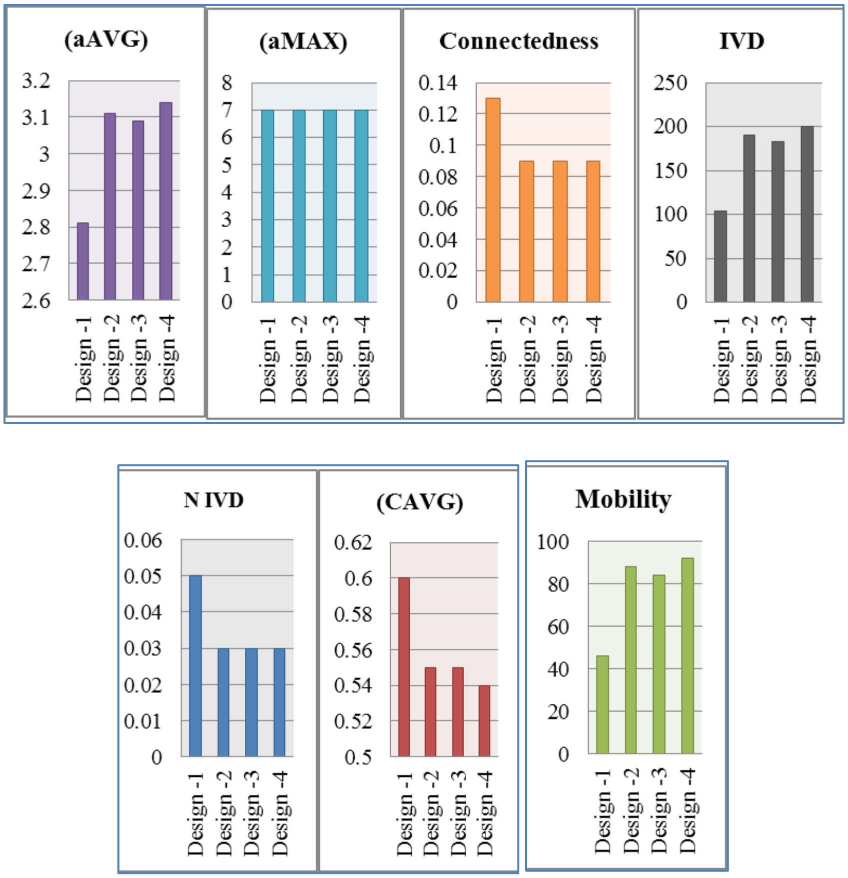


Figure 3- 14 Bar charts for the size, decomposability and coupling measures of design process structure [13]

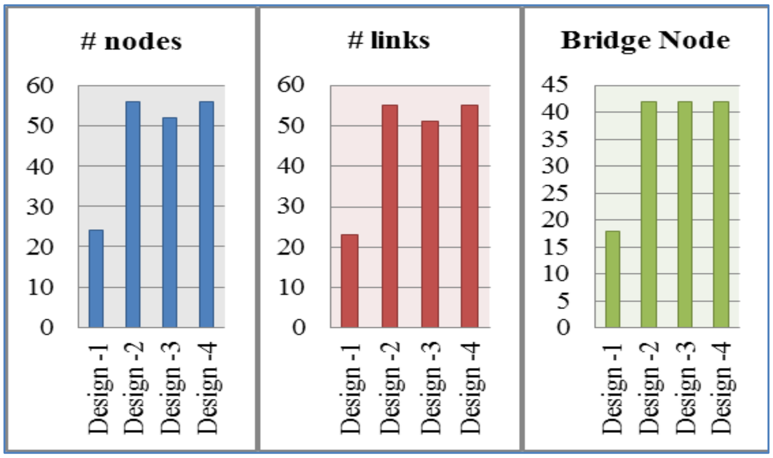
3.6.8. Manufacturing structures results: Appendix C contains the details of Process Structures for the four design alternatives. The complexity metrics results for manufacturing process structures are summarized in the Table 3-3 and Charts below.

Table 3-3 Complexity metrics results for manufacturing process structures [13]

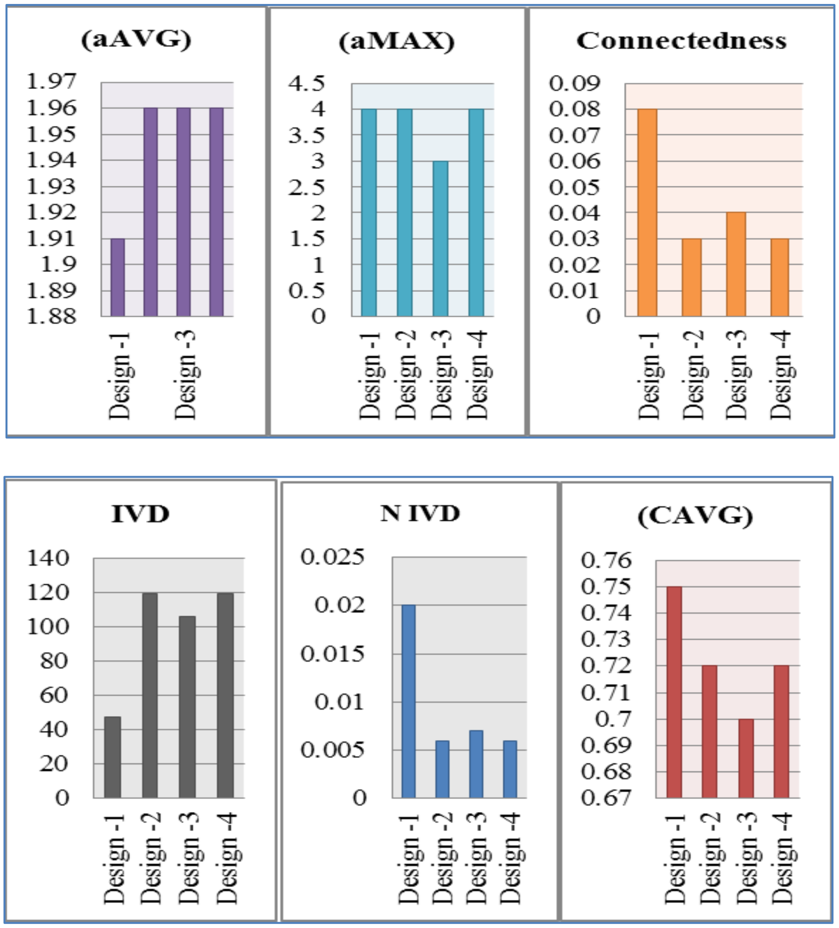
Metric	Design1	Design2	Design3	Design4
# nodes	24	57	52	57
# links	23	55	52	55
# Bridge links	9	22	18	22
Average node degree (a_{AVG})	2.04	2.07	2.17	2.07
Maximum node degree	4	4	4	4
Connectedness	0.08	0.03	0.03	0.03
I_{VD}	53.50	133.01	133.01	133.01
N I_{VD}	0.02	0.007	0.0088	0.0072
Clustering coefficient (C_{AVG})	0.68	0.68	0.65	0.68
Total Time Estimate (T)	37.62	265.44	155.18	265.44
Mfg Cost estimate	489.07	2,760.75	1,789.25	2,760.75
Mobility	18	50	46	50

3.6.9. Manufacturing Process Comparison Charts

Size and decomposability measures



Coupling measures



Cost, Time

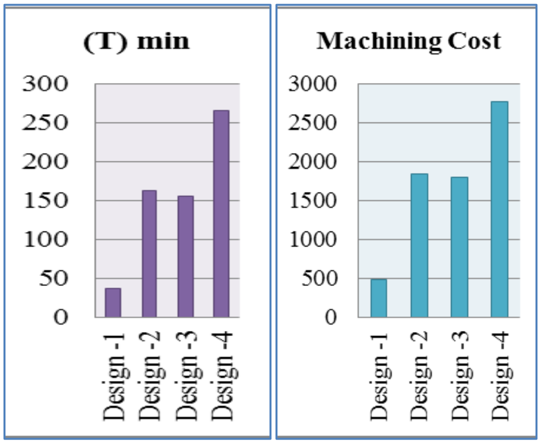


Figure 3- 15 Bar charts for the size, decomposability and coupling measures of manufacturing process structure [13]

3.7 Correlations: Table 3-4 shows correlation of metrics for all three structures.

Table 3-4 Correlation tables [13]

	# nodes (FS)	# links (FS)	Bridge Node (FS)	(aAVG) (FS)	(aMAX) (FS)	Connectedness (FS)	IVD (FS)	N IVD (FS)	(CAVG) (FS)	Mobility (FS)	# nodes (PS)	# links (PS)	# Bridge Nodes (PS)	(aAVG) (PS)	Connectedness (PS)	IVD (PS)	N IVD (PS)	(CAVG) (PS)	(T) (min) (PS)	Mobility (PS)	# nodes (MS)	# links (MS)	# Bridge nodes (MS)	(aAVG) (MS)	(aMAX) (MS)	Connectedness (MS)	IVD (MS)	N IVD (MS)	(CAVG) (MS)	(T) (min) (MS)	Machining Cost ($) (MS)	Mobility (MS)
# nodes (FS)	1.00																															
# links (FS)	0.85	1.00																														
Bridge Node (FS)	0.95	0.64	1.00																													
(aAVG) (FS)	-0.64	-0.14	-0.85	1.00																												
(aMAX) (FS)	0.87	0.94	0.71	-0.24	1.00																											
Connectedness (FS)	-1.00	-0.81	-0.96	0.70	-0.82	1.00																										
IVD (FS)	0.69	0.97	0.43	0.11	0.90	-0.63	1.00																									
N IVD (FS)	-1.00	-0.82	-0.96	0.68	-0.83	1.00	-0.65	1.00																								
(CAVG) (FS)	0.88	0.54	0.97	-0.86	0.68	-0.90	0.33	-0.88	1.00																							
Mobility (FS)	0.74	0.98	0.49	0.04	0.90	-0.69	1.00	-0.70	0.38	1.00																						
# nodes (PS)	0.97	0.94	0.85	-0.46	0.90	-0.96	0.83	-0.96	0.74	0.87	1.00																					
# links (PS)	0.97	0.95	0.84	-0.45	0.90	-0.95	0.83	-0.96	0.74	0.87	1.00	1.00																				
# Bridge Nodes (PS)	0.33	0.69	0.10	0.41	0.76	-0.25	0.81	-0.26	0.12	0.76	0.44	0.45	1.00																			
(aAVG) (PS)	0.97	0.94	0.85	-0.46	0.90	-0.96	0.83	-0.96	0.75	0.86	1.00	1.00	0.45	1.00																		
Connectedness (PS)	-0.99	-0.89	-0.90	0.57	-0.86	0.98	-0.74	0.99	-0.81	-0.79	-0.99	-0.99	-0.33	-0.99	1.00																	
IVD (PS)	0.97	0.95	0.84	-0.44	0.91	-0.95	0.84	-0.96	0.73	0.88	1.00	1.00	0.46	1.00	-0.99	1.00																
N IVD (PS)	-0.99	-0.89	-0.90	0.57	-0.86	0.98	-0.74	0.99	-0.81	-0.79	-0.99	-0.99	-0.33	-0.99	1.00	-0.99	1.00															
(CAVG) (PS)	-0.97	-0.95	-0.85	0.45	-0.93	0.95	-0.83	0.96	-0.77	-0.87	-1.00	-1.00	-0.49	-1.00	0.98	-1.00	0.98	1.00														
(T) (min) (PS)	0.98	0.90	0.89	-0.54	0.87	-0.98	0.76	-0.98	0.79	0.81	0.99	0.99	0.36	0.99	-1.00	0.99	-1.00	-0.99	1.00													
Mobility (PS)	0.97	0.95	0.84	-0.44	0.90	-0.95	0.84	-0.96	0.73	0.87	1.00	1.00	0.45	1.00	-0.99	1.00	-0.99	-1.00	0.99	1.00												
# nodes (MS)	0.97	0.93	0.85	-0.48	0.87	-0.96	0.80	-0.96	0.73	0.85	1.00	1.00	0.39	1.00	-0.99	1.00	-0.99	-0.99	1.00	1.00	1.00											
# links (MS)	0.97	0.93	0.85	-0.48	0.87	-0.96	0.80	-0.96	0.73	0.85	1.00	1.00	0.39	1.00	-0.99	1.00	-0.99	-0.99	1.00	1.00	1.00	1.00										
# Bridge nodes (MS)	0.99	0.89	0.90	-0.57	0.86	-0.98	0.74	-0.99	0.81	0.79	0.99	0.99	0.33	0.99	-1.00	0.99	-1.00	-0.98	1.00	0.99	0.99	0.99	1.00									
(aAVG) (MS)	0.99	0.89	0.90	-0.57	0.86	-0.98	0.74	-0.99	0.81	0.79	0.99	0.99	0.33	0.99	-1.00	0.99	-1.00	-0.98	1.00	0.99	0.99	0.99	1.00	1.00								
(aMAX) (MS)	-0.45	0.06	-0.70	0.91	-0.15	0.49	0.28	0.47	-0.81	0.24	-0.22	-0.21	0.33	-0.23	0.33	-0.20	0.33	0.25	-0.30	-0.20	-0.22	-0.22	-0.33	-0.33	1.00							
Connectedness (MS)	-0.94	-0.95	-0.80	0.41	-0.87	0.93	-0.84	0.94	-0.68	-0.88	-0.99	-0.99	-0.42	-0.99	0.98	-0.99	0.98	0.98	-0.99	-0.99	-1.00	-1.00	-0.98	-0.98	0.14	1.00						
IVD (MS)	0.95	0.94	0.81	-0.43	0.87	-0.94	0.83	-0.95	0.69	0.87	1.00	1.00	0.41	0.99	-0.98	1.00	-0.98	-0.98	0.99	1.00	1.00	1.00	0.98	0.98	-0.16	-1.00	1.00					
N IVD (MS)	-0.98	-0.91	-0.87	0.52	-0.87	0.97	-0.78	0.98	-0.77	-0.82	-1.00	-1.00	-0.36	-1.00	1.00	-0.99	1.00	0.99	-1.00	-0.99	-1.00	-1.00	-1.00	-1.00	0.27	0.99	-0.99	1.00				
(CAVG) (MS)	-0.94	-0.61	-1.00	0.86	-0.70	0.96	-0.40	0.95	-0.98	-0.46	-0.83	-0.82	-0.08	-0.83	0.89	-0.82	0.89	0.84	-0.87	-0.82	-0.83	-0.83	-0.89	-0.89	0.73	0.78	-0.79	0.86	1.00			
(T) (min) (MS)	0.83	0.98	0.63	-0.12	0.99	-0.78	0.96	-0.79	0.57	0.96	0.90	0.90	0.79	0.90	-0.84	0.91	-0.84	-0.92	0.85	0.90	0.87	0.87	0.84	0.84	0.00	-0.88	0.88	-0.86	-0.61	1.00		
Machining Cost ($) (MS)	0.87	0.98	0.68	-0.19	0.99	-0.82	0.94	-0.83	0.62	0.95	0.93	0.93	0.74	0.93	-0.88	0.94	-0.88	-0.95	0.89	0.93	0.90	0.90	0.88	0.88	-0.05	-0.91	0.91	-0.89	-0.66	1.00	1.00	
Mobility (MS)	0.97	0.93	0.85	-0.48	0.87	-0.96	0.80	-0.96	0.73	0.85	1.00	1.00	0.39	1.00	-0.99	1.00	-0.99	-0.99	1.00	1.00	1.00	1.00	0.99	0.99	-0.22	-1.00	1.00	-1.00	-0.83	0.87	0.90	1.00

3.8 Evaluation of complexity metrics with respect to the application and mathematical properties

Table 3-5 Evaluation of complexity metrics

	Holistic	Correlate reality	Consistent	Sensitive	Repeatable	Scalable	Monotonicity	Transitivity	Non-dimensional	Continuity	Invariant to affine transformations
Node	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Links	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
a_{AVG}	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
a_{MAX}	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
Conn.	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
I_{VD}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NI_{VD}	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
C.C.	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
Cost	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Time	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mob	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
B. link	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓

From the first experiment it was found the average node degree, maximum node degree, normalized information content and clustering were not satisfying the some application properties.

3.9 Inferences from the results

This experiment contrived a simple mechanical design problem in order to generate multiple alternatives that have clear advantages and disadvantages. These systems, however, are very standard, designed by well-known procedures and manufacturing processes. Therefore, many sources of complexity, such as uncertainty and unanticipated interactions, are not represented in this problem space. Even with this relatively simple synthetic problem we can see that certain quantifiers discriminate well between simple and complex systems and some do not.

From a cursory examination, it seems that Design1 is clearly the simplest because it has fewest parts, fewest interactions, least alignment issues and liberal tolerances. It is a very difficult to discriminate between Designs 2, 3, 4. All of those have 2 gear pairs. Designs 2, 3 each have a spur gear pair, so the discriminator between them is bevel versus face pair. So these designs should be fairly close in complexity. Design 4 uses both a hypoid and a helical arrangement; these gears require more complex machining and assembly alignment. Additionally, both radial and thrust loads require either additional bearings or more complex bearings. Therefore, we can say that Design1 appears to be least complex by far; Designs 2, 3, 4 have comparable complexity with Design 4 being the highest.

The simplest measures of size (#nodes, #links) appear to correlate well with our own engineering assessment of complexity in all domains (artifact, process and manufacturing). This is further confirmed by comparing to process development time and manufacturing cost. Pure bridge link count does not correlate with our own engineering assessment of complexity. Connectedness measures appear to have a negative correlation with complexity, a rather surprising result. The normalized information content measure also is negatively correlated while the non-normalized version is positively correlated to complexity.

Mobility also showed positive correlation with the cost. Node degree measures do not appear to discriminate the designs much at all.

3.10 Design and Implementation of Matlab code for unweighted complexity metrics:

The main requirements (functions) of the program are as follows:

1) Generate the graph in which nodes (variables) and links can be display and edit. The program should not only be able to read from the input file and but also from the interaction with the user.
2) Populate the adjacency matrix from the node and links in the graph.
3) Calculate the complexity metrics from the adjacency matrix.
4) Show the results and comparison charts of complexity metrics.
5) Save the adjacency matrix and results to a file.

3.10.1. Implementation: The whole program is divided into the following 5 functions:

1) *adj_matrix_gui*: This function creates the graph from the beginning or from the old saved file. A node can be created by left double click (LMB) and delete by right click (RMB). A link can be made between 2 nodes by first single click (LMB) on first node and then single click (LMB) on the second node. A link can be deleted by single right click (RMB). This function also generates the adjacency matrix from the graph. This function was derived by modifying the base function *Adjacency Matrix GUI* at the Mathwoks website [15].

2) *calc_metrics*: This function calculates all the complexity metrics from the adjacency matrix. This function also calls other functions, such as *bridgeNode, pruneingleafnodes, connection* etc., to compute the complexity matrices.

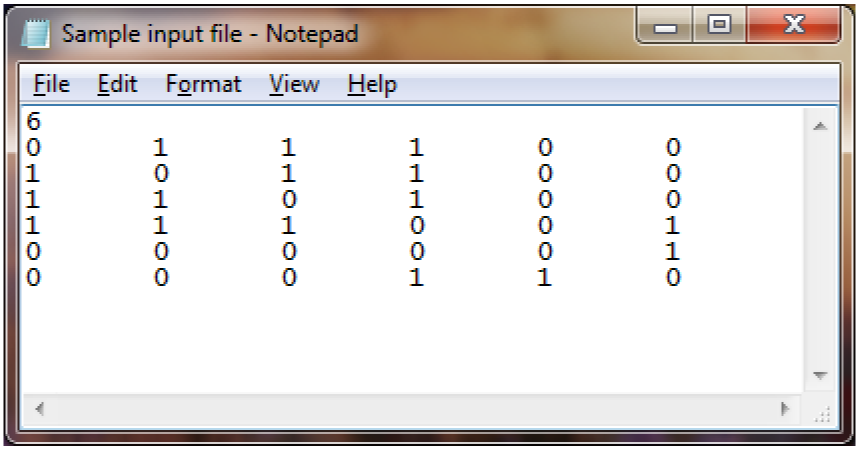
3) *OpenData:* This function is called by the function adj_matrix_gui to create the graph from the old existing file. This function reads the size i.e. number of nodes in the input file and creates a graph in which all the nodes are placed at the vertex of a polygon. The nodes are then connected by links according to the values present in the adjacency matrix. Nodes i and j are dependent if value at a cell a(i, j) of adjacency matrix is 1 and it is represents as a link plotted between the nodes. Nodes i and j are

independent if value at a cell a(i, j) of adjacency matrix is zero and it is represents as a no link between the nodes.

4) *Plot charts:* The role of this function is to read the results from function *calc_metrics* and display the results in the message box and bar-chart.

5) *OutPutVertex_Matrix_ToFile:* This function is used for saving the graph. It writes the number of nodes and adjacency matrix to the file.

3.10.2. Format of input file: First line of the input file has the information about the total number of nodes in the graph. Adjacency matrix is stored in the input file after the first line. A sample input file is shown in figure 3-16.



```
6
0          1          1          1          0          0
1          0          1          1          0          0
1          1          0          1          0          0
1          1          1          0          0          1
0          0          0          0          0          1
0          0          0          1          1          0
```

Figure 3- 16 sample input file for unweighted complexity matrices

The graphical representation for the above sample is given in figure 3-17

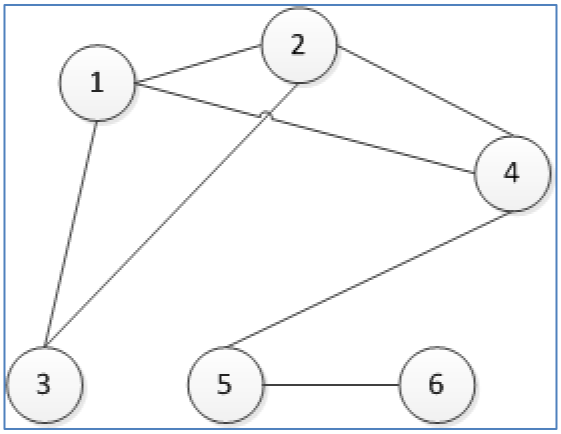


Figure 3- 17 Graph for the sample input file

The detail code of main file and all the function files, written in the MATLAB, is included in APPENDIX D.

CHAPTER 4

WEIGHTED COMPLEXITY METRICS

In the previous chapter an evaluation of a wide range of complexity measures relevant to various product life cycle phases was presented. It was shown that three domains would capture the overall complexity: functional structure, process structure and manufacturing structure. All the nodes and links in the experiment were assumed to be equal. However, in the real world problem variables (nodes) in a design have different importance. For example, consider two shafts, one rotating at 1000 rpm and other rotating at 50000 rpm. The designing of the latter shaft will not be simple because plenty of additional things need to consider e.g. dynamics (whirly speed), special bearings, coolant etc. It will also require keeping the frictional losses and temperature as low as possible in order to increase the efficiency of the system. Similarly all the relationships (links) are not equal in the real world problem. For example, finding the deflection of a beam in the elastic range is simple but in plastic range is not. Therefore, nodes and links should be assigned different weights depending upon how much it is difficult to compute them during design process.

In this chapter various methods of assigning weights nodes and links is explained. Second experiment “Hybrid powertrain architectures” is conducted in which different architectures of hybrid vehicle are generated for an IFV (Infantry Fighting Vehicle). The complexity metrics, using weighted nodes and links, were calculated for these different architectures.

4.1 Weighting schemes for variables (nodes):

All Variables in a real world design problems are not equally important, as were all links. Certain relationships cause greater problems than others [16]. Also, if we were looking at the system level and nodes were subsystems or components, they would be unequally important. There are three possible weighting schemes for associating weights with nodes:

1. Technological difficulty
2. Eigenvalue centrality
3. Sensitivity and percentage contribution to functional requirements

4.1.1 Technological difficulty method for Function structures

Nodes representing independent design variables can be rated based on the difficulty of achieving a particular parameter range (technological complexity).

Table 10-1 Node weight scheme for functional structure [16]

Parameter range	**weight**
Standard; conventional	1
Less common but achievable	2
Not found in any current machine; Beyond recommended range	3

Assigning Weights to nodes in process structures and manufacturing structure is more straightforward. The nodes are tasks, which can be rated based on expertise needed, time required, and expense of tool or method.

Table 4-2 Node weight scheme for process and manufacturing structure [16]

Weight	Expertise	Time (relative)	Tool
1	routine	low	none
2	moderate	medium	general purpose
3	highly specialize	high	special purpose

The links can be weighted based on the number of times they are traversed, i.e. the number of iterations needed. This approach can be used to both design and manufacturing process structures. The onus of assigning weights based on tech complexity would fall on domain experts constructing the function and process models.

4.1.2 Eigenvalue centrality method

In this method eigenvalues from adjacency matrix of the graph are calculated and then the eigenvector for the maximum eigenvalue gives the centrality of every node in the graph. According to Phillip Bonacich [17], the absolute values of the eigenvector indicate how important a node is in the graph and therefore these values can be used as weights for nodes in the graph.

The process of calculating weight of nodes by the eigenvector centrality method is given below:

1. Functional graph for a given problem is produced from the variables list and their relations. A random functional graph is shown on the next page in figure 4.1

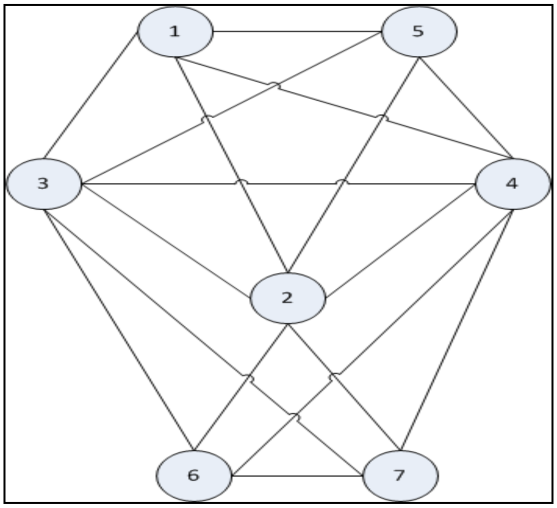


Figure 4-1 Functional graph for a problem

2. Adjacency matrix is generated from the functional graph.

$$A= \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

3. Characteristic equation is given by

$$\det\{[A]-\lambda[I]\}=0$$

Where A is adjacency matrix, λ is eigenvalue, det is determinant of the matrix and I is identity matrix. The characteristic equation is shown below.

$$\det \begin{pmatrix} -\lambda & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & -\lambda & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -\lambda & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -\lambda & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -\lambda & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -\lambda & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & -\lambda \end{pmatrix} = 0$$

4. Eigenvalues are found by solving the characteristic equation for λ. For Beam problem eigenvalues are -2.3478, -1.00, -1.00, -1.00, 0.00, 0.5305 and 4.8173.
5. For the maximum eigenvalues, a set of simultaneous linear equations are solved to determine the corresponding eigenvectors. For beam problem calculation is shown below:

$$\begin{pmatrix} -4.8173 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & -4.8173 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -4.8173 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -4.8173 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -4.8173 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -4.8173 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & -4.8173 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = 0$$

By solving above equation following eigenvector is found [-0.3511, -0.4467, -0.4467, -0.4467, -0.3511, -0.2782, 0.2782]. The absolute values of this eigenvector are used as weights for the nodes in the graph. Node numbers with weights are shown below in Table 4.3.

Table 4-3 Node weights of the functional structure [16]

Node Number	Node Weights
1	0.3511
2	0.4467
3	0.4467
4	0.4467
5	0.3511
6	0.2782
7	0.2782

4.1.3 Sensitivity based weighting

The importance of variables is related to design goals, functional requirement or components of optimization objective functions [18]. Then the sensitivity and/or percent contribution of that goal to each independent variable is computed. Consider a functional requirement A that is dependent on variables d_i:

$$A = f(d_1, d_2, ..., d_n)$$

Requirement A can be linearized about the variables' nominal values d_i, using the Taylor's Series expansion:

$$A \approx f(\overline{d_1}, ..., \overline{d_n}) + \sum_{i=1}^{n} (\frac{\partial f}{\partial d_i} \cdot \Delta d_i)$$

where $\frac{\partial f}{\partial d_i}$ are the sensitivity (same as $\frac{\partial A}{\partial d_i}$)

This assumes small perturbations about the nominal. The variance of A is obtained as:

$$\sigma_A = \sqrt{\left(\frac{\partial f}{\partial d_1}\right)^2 \sigma_{d_1}^2 + \left(\frac{\partial f}{\partial d_2}\right)^2 \sigma_{d_2}^2 + \ldots + \left(\frac{\partial f}{\partial d_n}\right)^2 \sigma_{d_n}^2}$$

The percentage contribution of d_i is computed as

$$\%C = \frac{(\partial f / \partial d_i)^2 {\sigma_i}^2}{{\sigma_A}^2} \times 100\%$$

Sensitivities and percentage contribution (%C) can be used as node weights.

This idea can be implemented in different ways. For example, we can use DOE and conduct a factorial experiment at 2 or 3 values (max, min, mean) for each independent variable [19].

4.1.4 Comparison of Node weight methods

Weights based on technological rating involve domain expertise and subjectivity. Different people might arrive at different results. It is probably most likely to correlate best with expert ratings and be the most meaningful. Weighting schemes are different for different types of structures (function, process, and manufacturing). It would be difficult to automate. It does offer the advantage of assigning weights to links independent of nodes.

Eigenvalue centrality uses connectivity information already present in the un-weighted graphs. There is no new information added. In that respect it would be similar to other coupling measures in the aggregate. Perhaps it is only usefulness might lie in determining an efficient design process sequence. It is easy to automate for any type of graph regardless of node/link meaning.

Sensitivity measures depend entirely on parametric relations, so do not require any subjective judgment. They take into account what the design goals are that we really care about. They require the nodes in the graph to be classified into: Result node, independent variables and dependent variables. If there are multiple Result nodes, each needs to have its own DOE done separately. This method seems to be only suited for functional structures. Stacking of variability in manufacturing can be studied this way but that is related to the quality of the artifact not just its complexity. Based on the above analysis technological difficulty has been selected as the basis for assigning weights to the nodes in the functional structure.

4.2 Assigning weights to the links

Link weights should be based on the mathematical nature of the relationship such as degree to which known in advance, static or dynamic, linear or nonlinear, stable or unstable, single mode or multi-modal (relation changes in different parameter values) and synchronized or unsynchronized [16].

These factors could be individually weighted and an aggregate score computed. Alternatively, we could arrange them in a hierarchy and rate the branches, as shown below:

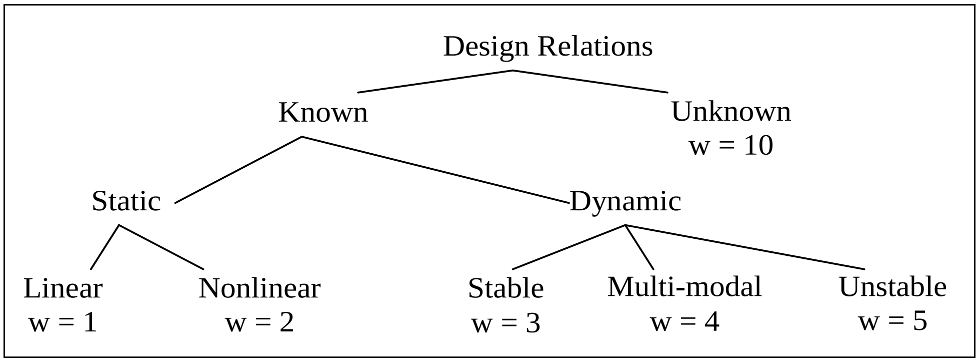


Figure 4-2 Decision tree for link weighing scheme [16]

The leaves in this tree have been given arbitrary weights. Correct weighting has to be determined by calibration. For bi-partite representation we can apply the link weights to the relation nodes instead of the links. The interpretation of node and link weights for each domain is shown in the Table 4-4.

Table 4-4 Interpretation for each domains weighing method [16]

Structure	**Nodes**	**Links**	**Node wt.**	**Link wt.**
Function	Design variable	Parametric relation	Feasibility range	Linearity, stability of relation
Process	Design, analysis task	task dependencies	Task skill, difficult	# iterations
Manufacturing	Manufacturing operation	Operation sequence	Process capability	Re-work

4.3 Weighted Candidate Metrics definitions

Unweighted complexity metrics are modified weighted complexity metrics by incorporating node weights and link weights as follows. Some metrics measure size complexity, coupling complexity, or solvability. Others may measure combine size and coupling or size and solvability. Both weighted and unweighted versions of each metrics are classified according to the type of complexity they measure. Classification of complexity metrics is as shown below.

Size measures

The size of the design problem can be measured by # nodes, # link and total size. The formulae for the Size measures are given below in Table 1

Table 4-5 Size complexity metrics [37]

Size complexity metrics		
S. No	***Metric***	***Measure***
1	# Node	N
2	# links	L
3	Total Size	N+L

Note: N is the number of nodes; L is the number of links.

Solvability measures:

Solvability measures indicate how much it is difficult to solve a given problem. Two solvability measures are average node weight and average link weight. Average node weight is calculated by dividing total node weight with # nodes. Similarly average link weight is calculated by dividing total link weight with # links.

Table 4-6 Solvability complexity metrics [37]

Solvability complexity metrics		
S. No	***Metric***	***Measure***
1	Average node weight	$\frac{\sum_{i=1}^{N} w_i}{N}$
2	Average link weight	$\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}}{L}$

Coupling measures

Coupling measures indicates how design parameters are related to each other. There are several coupling measures:

Average node degree: It is define as the ratio of sum of all the node degree to the total number of links.

Connectedness (graph density): It is define as the ratio of number of links to the maximum weighted links possible in the graph.

Information content of the node degree: it is a modified form of Shannon's Information content in which probability is replaced by weighted node degree of nodes.

Normalized Information content of the node degree: It is given by the ratio of Information content of the node degree to the Information content of the complete graph (maximum node degree).

Average clustering coefficient: it is a measure of interaction among the set of nodes in the graph. It is given by the ratio of sum of all clustering coefficient to the # nodes. $C_{AVG} = (\sum_{i=1}^{N} C_i) / N$

Where Ci is the clustering coefficient and is defined as the ratio of the number of links Li between the neighbors of the node i, and the maximum no of links Li (max) =k (k-1)/2, in a complete graph that can be formed by the nearest neighbors (k) of that node. $C_i = \frac{2L_i}{k_i(k_i - 1)}$

Mobility: it is the "degree-of-freedom" of the structure. It is a measure of how much the structure is over or under-constrained.

Weighted Connectedness: It is define as the ratio of total link weight to the maximum weighted links possible in the graph.

Weighted Average clustering coefficient: It is given by the ratio of sum of all weighted clustering coefficient to the # nodes. Weighted clustering coefficient is given by

$$C_i = \frac{L_i}{5.k_i(k_i - 1)}$$

Table 4-7 Coupling complexity metrics [37]

Coupling complexity metrics		
S. No	***Metric***	***Quantifier***
1	Average node degree	$\frac{\sum_{i=1}^{N} a_i}{N}$
2	Connectedness	$\frac{2L}{N(N-1)}$
3	Information content	$\sum_{i=1}^{N} a_i\ log_2\ a_i$
4	Normalized info content of node degree	$\frac{\sum_{i=1}^{N} a_i \log_2 a_i}{N(N-1)\log(N-1)}$
5	Avg clustering coefficient	$\frac{\sum_{i=1}^{N} 2Li/k_i(k_i-1)}{N}$
6	Mobility	$3(N-L-1)+\mathrm{N}$
7	Wt. connectedness	$\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}}{N(N-1)}$
8	Weighted Average Clustering coefficient	$\frac{\sum_{i=1}^{N} 2Li/5k_i(k_i-1)}{N}$

Where a_i is the number links in/out of node i and $\emptyset_i = \sum_{j=1}^{a_i} w_{ij}$

<u>Size + solvability</u>

These measures give the weighted size of the design problem. It is given by total node weight, total link weight and Total weighted size.

Table 4-8 Size + solvability complexity metrics [37]

Size + solvability		
S. no	***Metric***	***Quantifier***
1	Total node weight	$\boldsymbol{\sum_{i=1}^{N} w_i}$
2	Total link weight	$\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\ ,\ i < j$
3	Total Weighted size	$\sum_{i=1}^{N} w_i + \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}$

Coupling + solvability

Weighted information content: it is derived from Information content of node degree in which node degree (a_i) is replaced by weighted node degree ($\emptyset_i$) of nodes.

Normalized Weighted information content: It is given by the ratio of weighted Information content of the node degree to the Information content of the complete weighted graph (maximum node degree).

Weighted average node degree: The weighted average node degree is calculated as a ratio of sum of all weighted node degrees to the #nodes.

Wt. mobility: it is derived from unweighted mobility in which degree of freedom per node is replaced with the total node weight (w_i).

Table 4-9 Coupling + solvability complexity measures [37]

Coupling + solvability		
S. no	***Metric***	***Quantifier***
1	Weighted information content	$\sum_{i=1}^{N} w_i(\emptyset_i\, log_2\, \emptyset_i)$
2	Normalized Weighted information content	$\frac{\sum_{i=1}^{N} w_i(\emptyset_i \log_2 \emptyset_i)}{3N\{10(N-1)[\log 10(N-1)]\}}$
3	Weighted average node degree	$\frac{\sum_{i=1}^{N} w_i}{N}$
4	Wt. mobility	$3(N-L-1)+\sum_{i=1}^{N} w_{\mathrm{i}}$

4.4 Experiment 2: Hybrid Powertrain

4.4.1 Objective of experiment: The equations derived in the previous section should be apply to an experiment to find out the feasibility of the complexity measures and separate the genuine complexity metrics from unfair complexity metrics. Hybrid power train of a GCV (Ground control vehicles) developed by Vanderbilt was selected for the experiment because it was not a standard design problem and the scope of this problem is assembly (system) level. Complexity metrics for only artifact functional structures are computed for this experiment.

4.4.2 Design specifications: The design specification of the experiment are:

1. Carrying capacity: It should be able to carry 12 soldiers (3 crew and 9 squad members).

2. Mobility requirements: (a) the max off road speed must be more than 30 mph and the maximum on-road speed must be more than 80mph. (b) It must accelerate to the speed of 30 mph from the 0 mph in only 8 seconds.

   (c) The maximum obstacle height that the vehicle could cross over is 24 inches.

   (d) The minimum turning radius (tractive effort) of the vehicle should be less than 0.7 m.

   (e) The range of the vehicle should be more than 300 miles.

   (f) The fuel efficiency of the vehicle must be more than 2 mpg.

   (g) The vehicle must exert ground pressure of less than17 psi.

   (h) The speed of the vehicle at 60% slope and 10% slope must be more than 4.1mph and 17 mph respectively.
3. Survivability: it should be able to survive the passive blast.
4. Transportability: Vehicle must be transferable by C-17 aircraft, rail and ship.

4.4.3 Alternative designs: Based on the current literature, there could be five alternative hybrid powertrain designs:

1. Conventional Powertrain: is the most common powertrain used in more than 90% of the total automotive used in the world. They are very common because they cover more distance when fueled completely than any other single source powertrain e.g. full electric drive. In this powertrain engine runs the transmission which supply the mechanical power to the wheels with the help of several important equipment such as driveshaft, differential and half shafts.

2. **Parallel hybrid:** Parallel hybrid powertrain uses only one device that acts as both motor and generator. This device also works as a starter motor. Both engine and motor drives the wheels with the help of transmission. In parallel hybrid power train engine always runs and provides a constant power supply to generator. It has 3 modes of working:

    a. Slow speed: At slow speed engine provide power to the transmission and addition power is provided by the motor/generator unit.

    b. Cruising: in this stage engine drives both transmission and generator unit. Current produced by the generator is used to charge the battery.

    c. Braking: in this stage transmission acts as a power source to generator which further charges the battery.

3. **Series-Parallel hybrid:** It combines the advantages of series and hybrid powertrains. At slow speed it works as a series hybrid and at high speed it works as parallel hybrid. In this configuration both engine and motor can run the vehicle independently. Series-parallel hybrid has 4 mode of working:

    a. Slow speed: Working of this powertrain is same as the series powertrain at slow speed. Motor drives the power splitter at slow speed. Engine is shut down at the slow speed.

b. Accelerating: during acceleration both engine and motor drives the power splitter.

c. Cruising: while driving at cruising engine gives power to both wheels and generator which further charges the battery.

d. Braking: during braking power splitter drives the generator which further charges the battery.

4. Series hybrid: it is the simplest hybrid configuration in which only motor runs the vehicle. In this configuration engine drives the generator which can charges the battery or provides electric energy directly to the motor. Battery, generator, engine and motor are controlled by electric control unit (ECU). This power train does not require any transmission. Series hybrid cars have 4 modes of working:

a. Slow speed: In this mode, motor drives the transmission using the electric power from the battery only. Engine and generator work in this mode only if the battery is discharged i.e. voltage is very low from the battery.

b. Acceleration: In this mode motor is run by the current given by both battery and generator.

c. Cruising: in this mode, engine drives the generator, which delivers the current to the motor and if surplus current is being generated then it also charges the battery.

d. Break: In this mode, motor acts as a generator and converts the energy lost during the deceleration into current which further charges the battery. IC engine shuts stops during this mode to safe the fuel.

5. All electric powertrain: This powertrain uses a huge capacity battery to the electric motor. This power train does not require any transmission but its range is quite limited. Some variants also use regenerative braking to charge the battery.

Every alternative design can also have 2 variants because of two different types of wheel drives i.e. 4x4 wheel drive and 6x6 wheel drive. The schematic diagrams have two kinds of flows: Energy (full lines) and signal (dashed lines). Mechanical, electrical/electronic and weapon systems are color coded green, blue, red, respectively, as shown in the figure-3. The schematic diagrams of Hybrid power train architectures are can be viewed in next 10 pages.

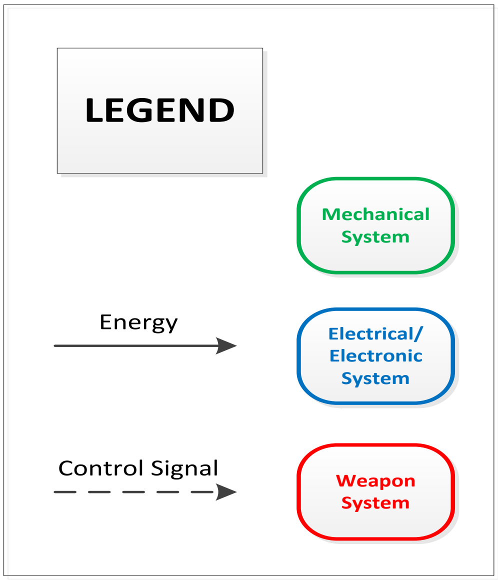


Figure 4-3 Legend for architectures [37]

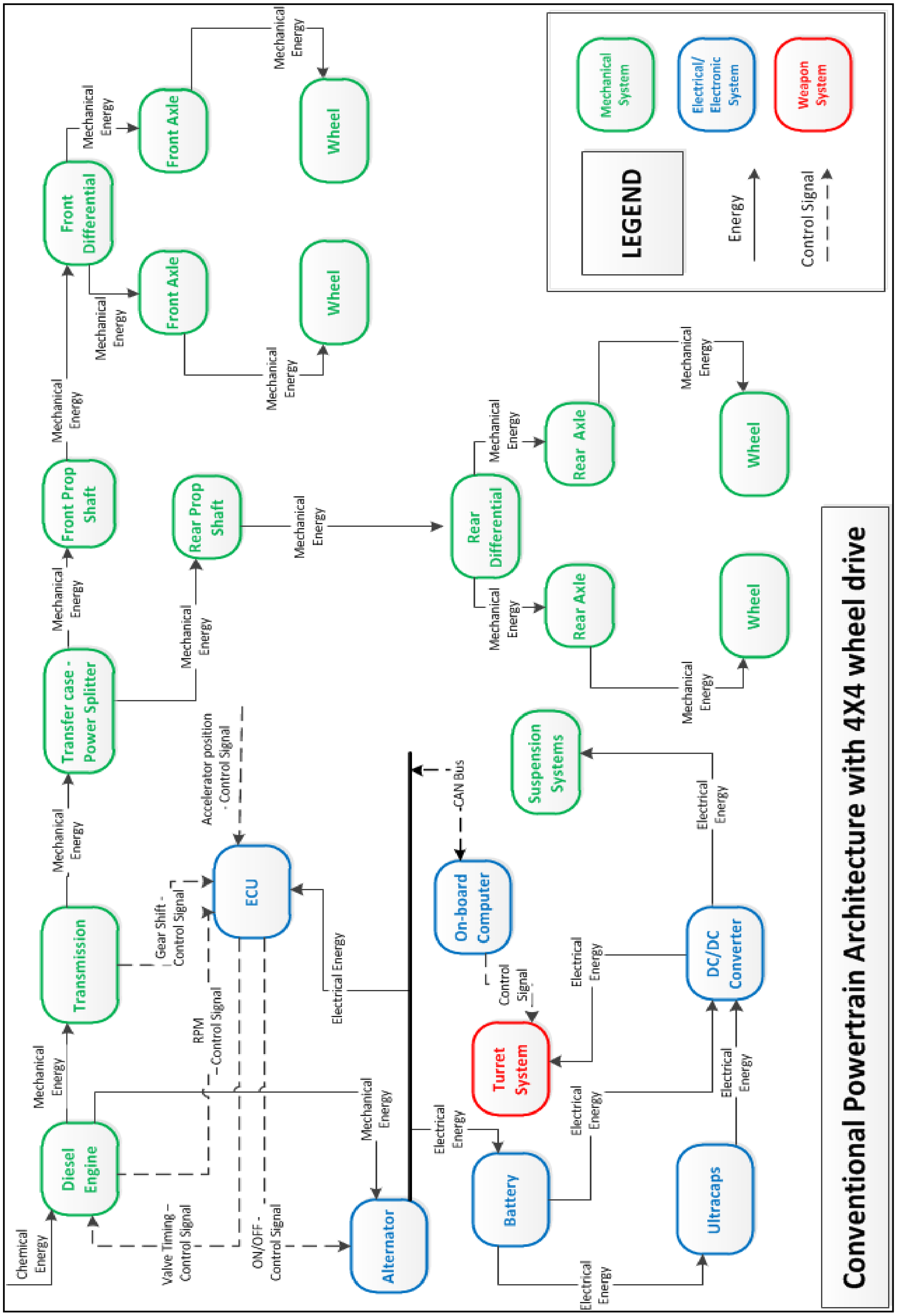


Figure 4-4 Conventional Powertrain Architecture with 4x4 wheel drive [37]

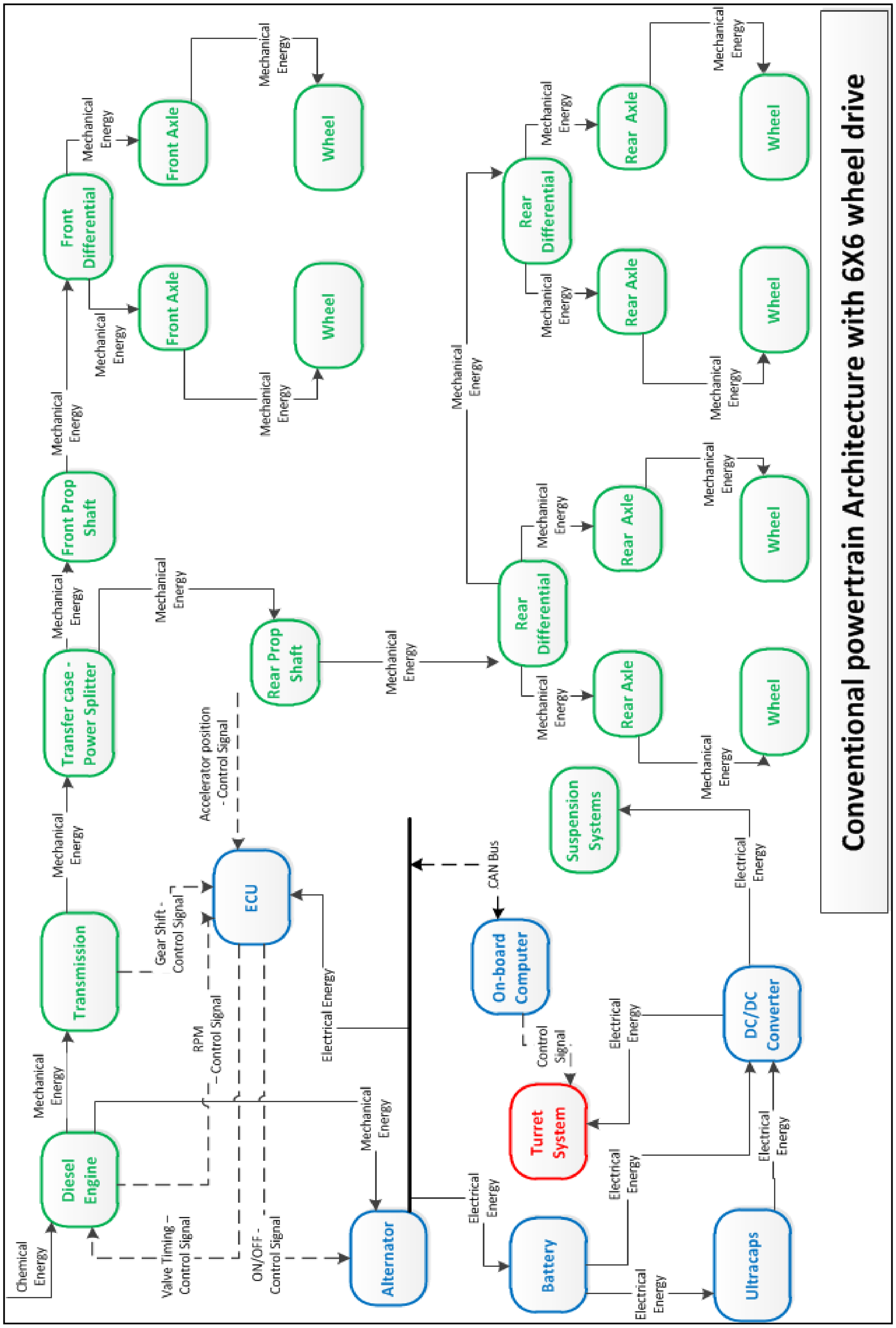


Figure 4-5 Conventional Powertrain Architecture with 6x6 wheel drive [37]

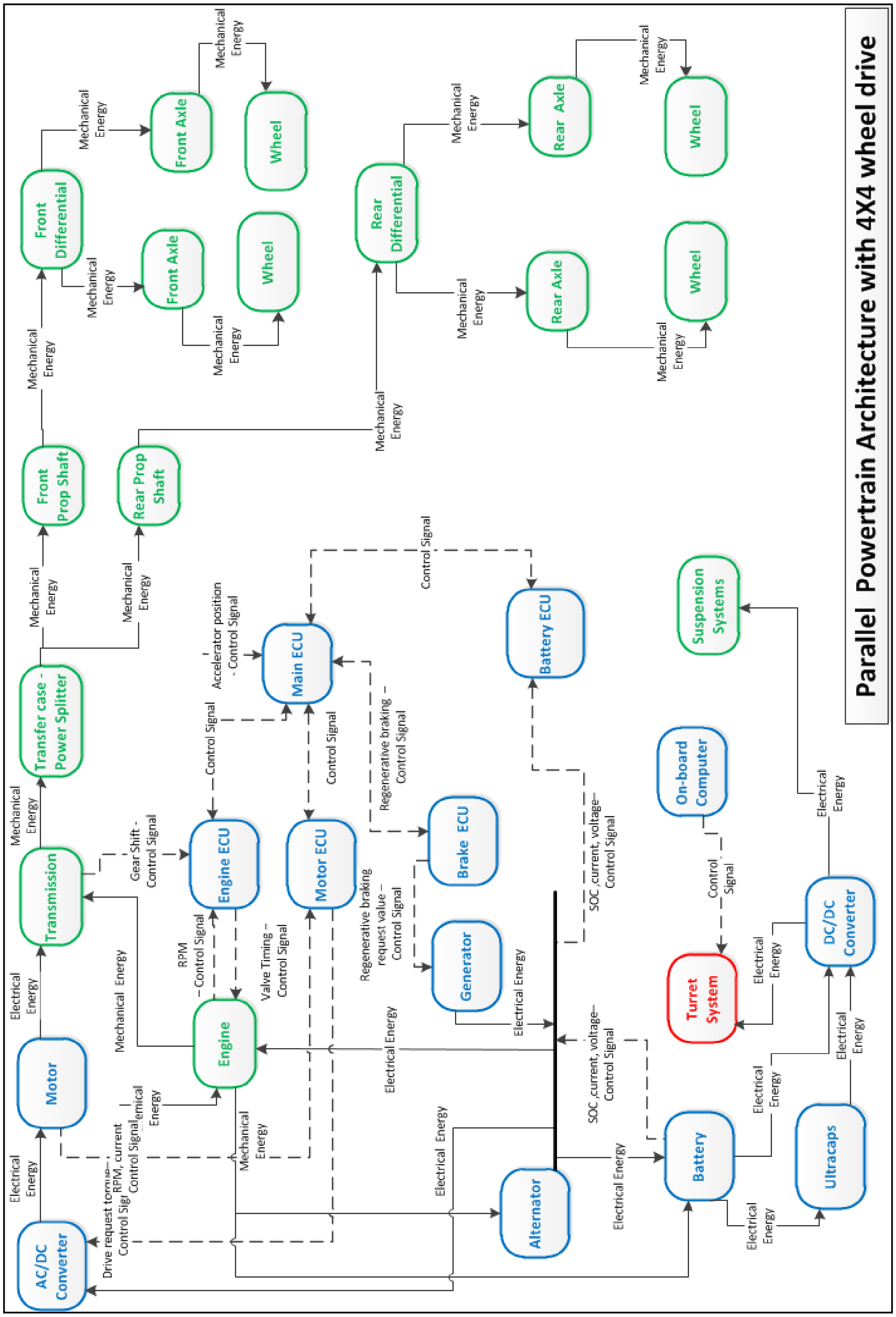


Figure 4-6 - Parallel Powertrain Architecture with 4x4 wheel drive [37]

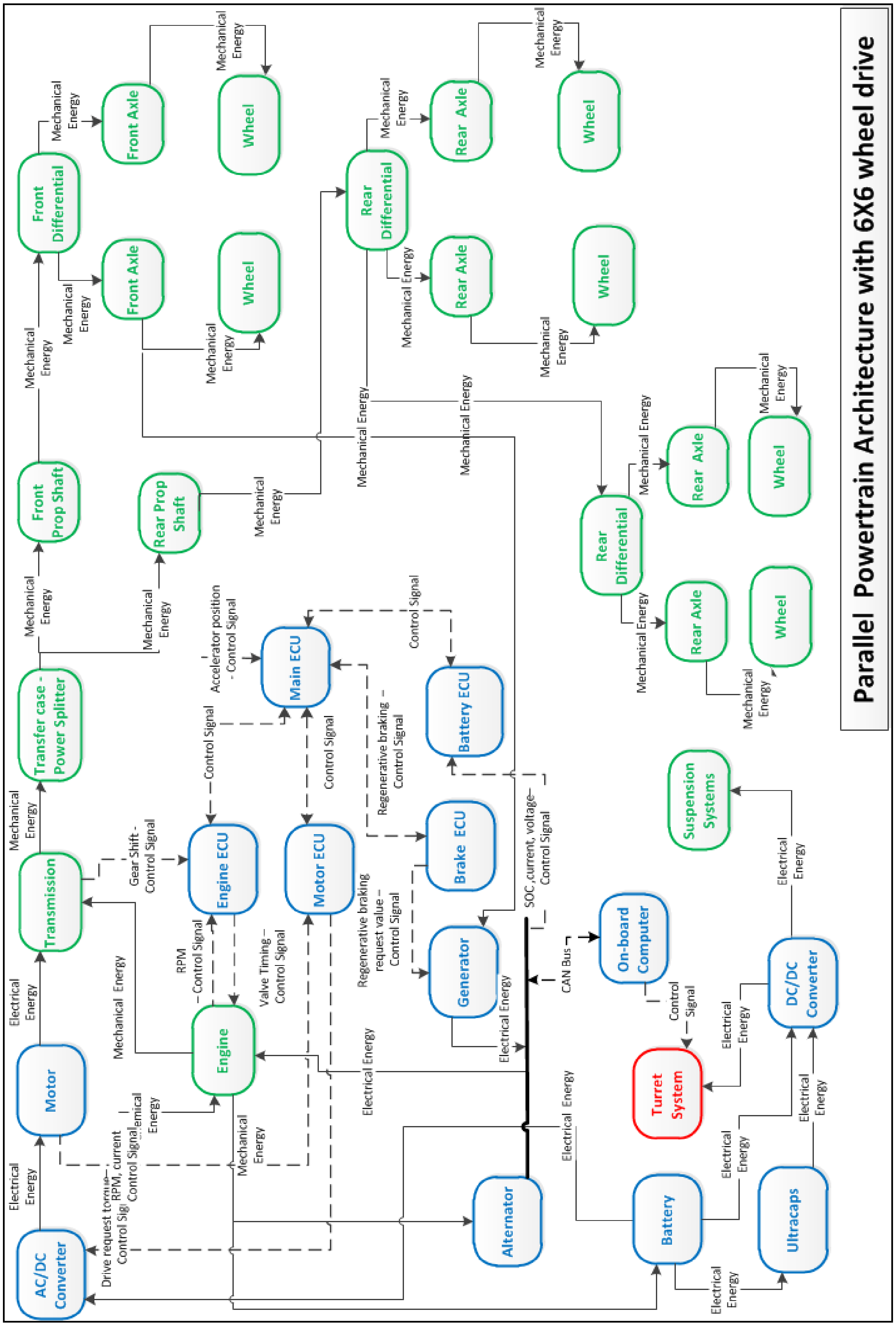


Figure 4-7- Parallel powertrain Architecture with 6x6 wheel drive [37]

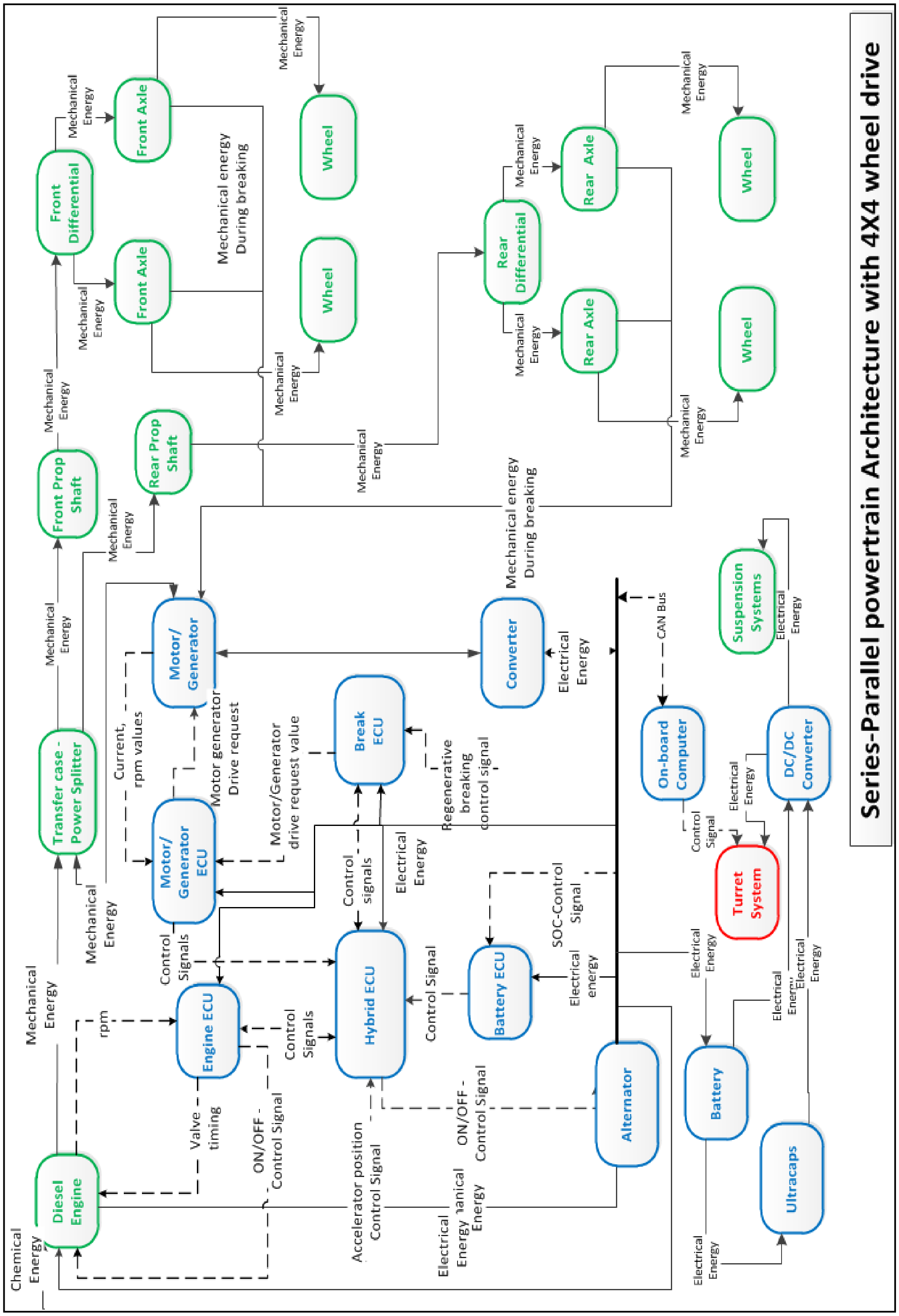


Figure 4-8- Series-Parallel powertrain with 4x4 wheel drive [37]

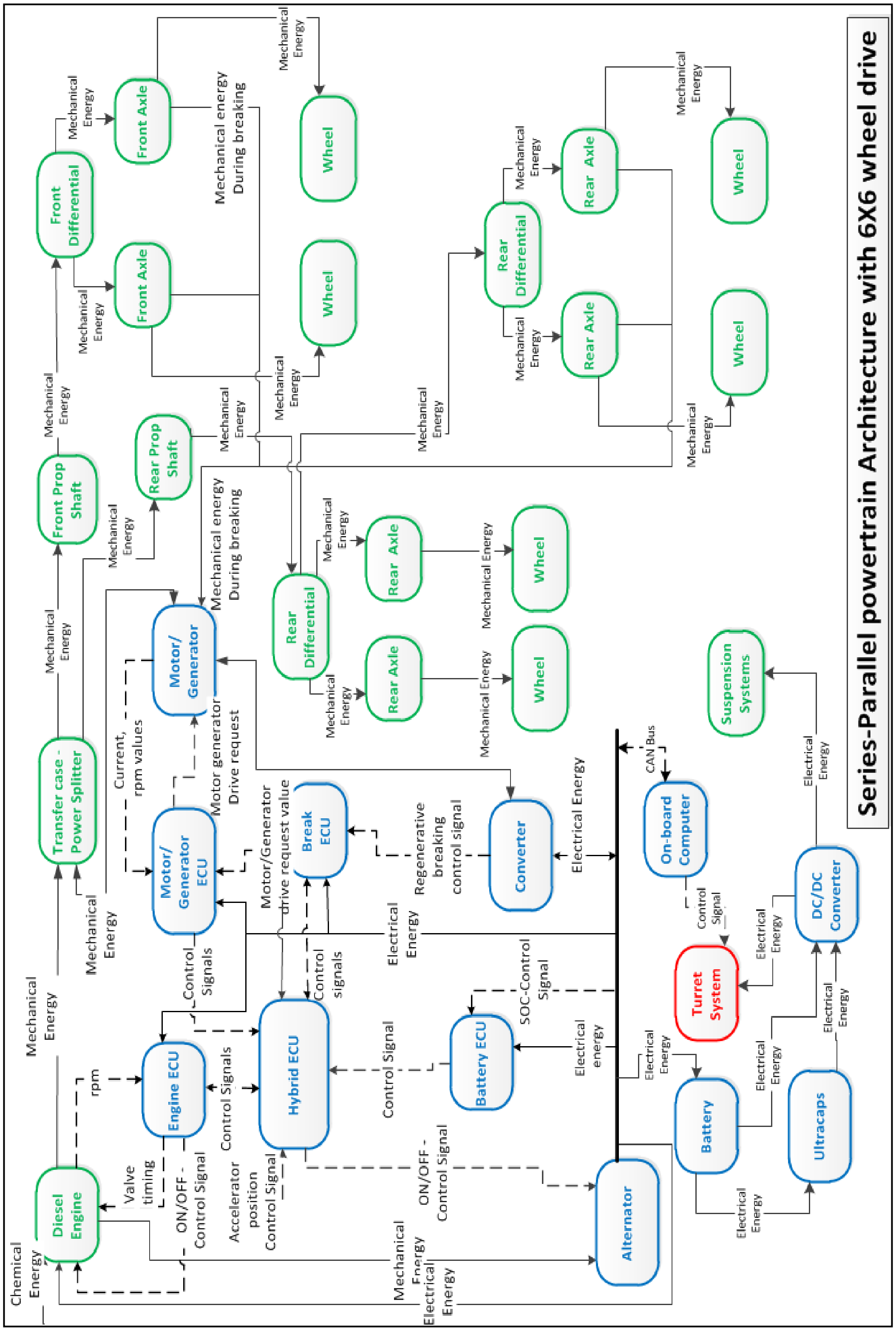


Figure 4-9-Series-Parallel powertrain with 6x6 wheel drive [37]

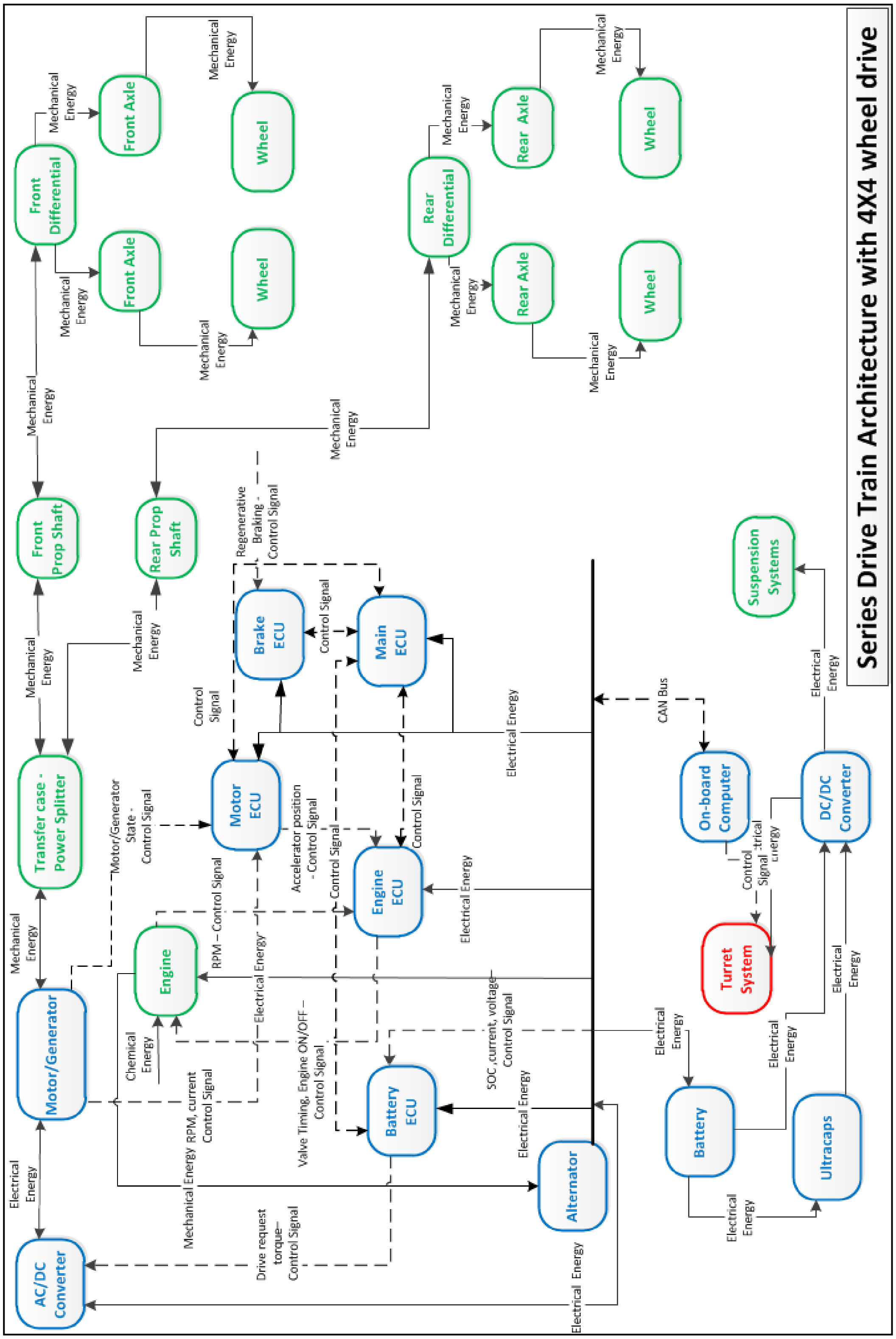


Figure 4-10- Series Powertrain Architecture with 4x4 wheel drive [37]

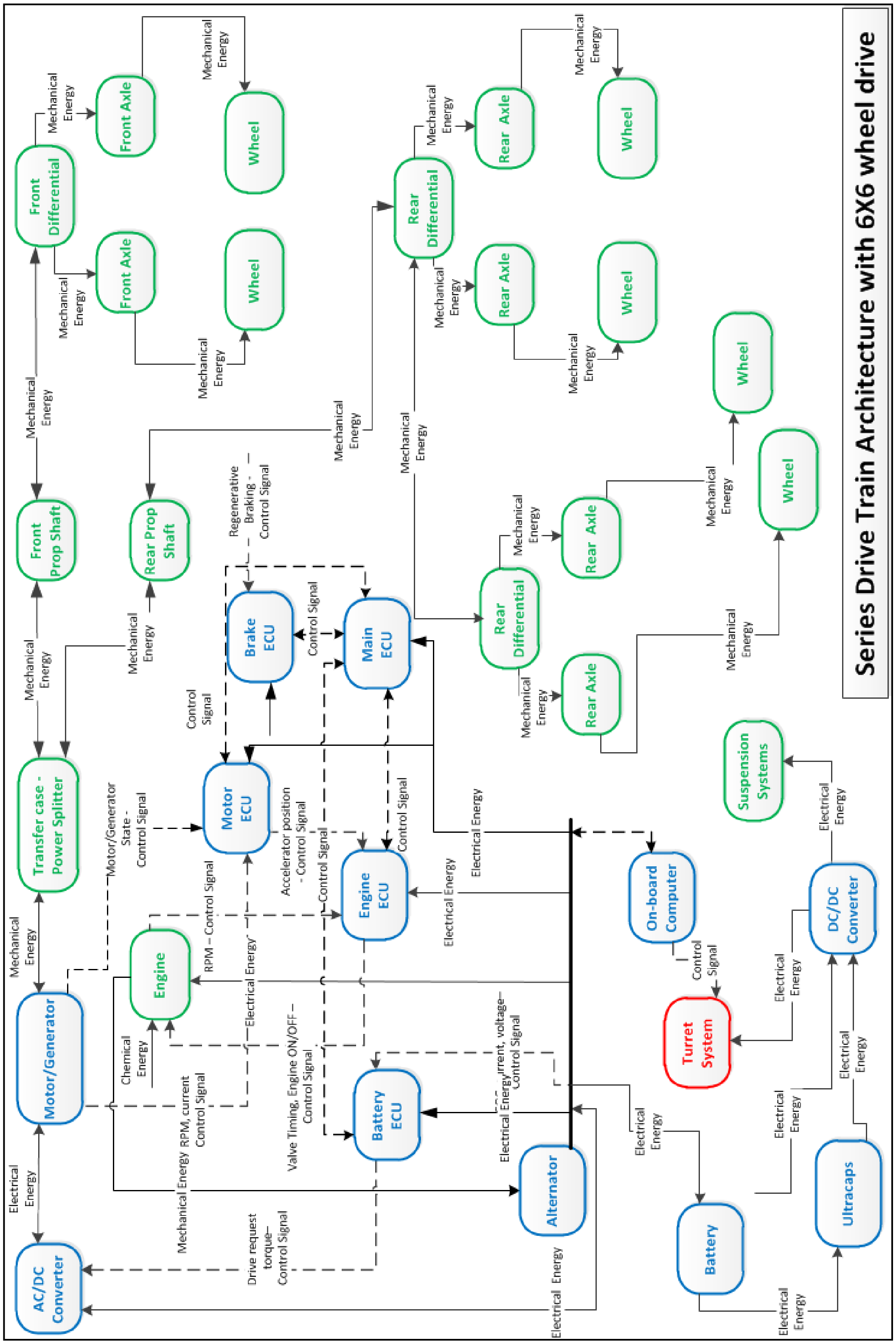


Figure 4-11- Series Powertrain Architecture with 6x6 wheel drive [37]

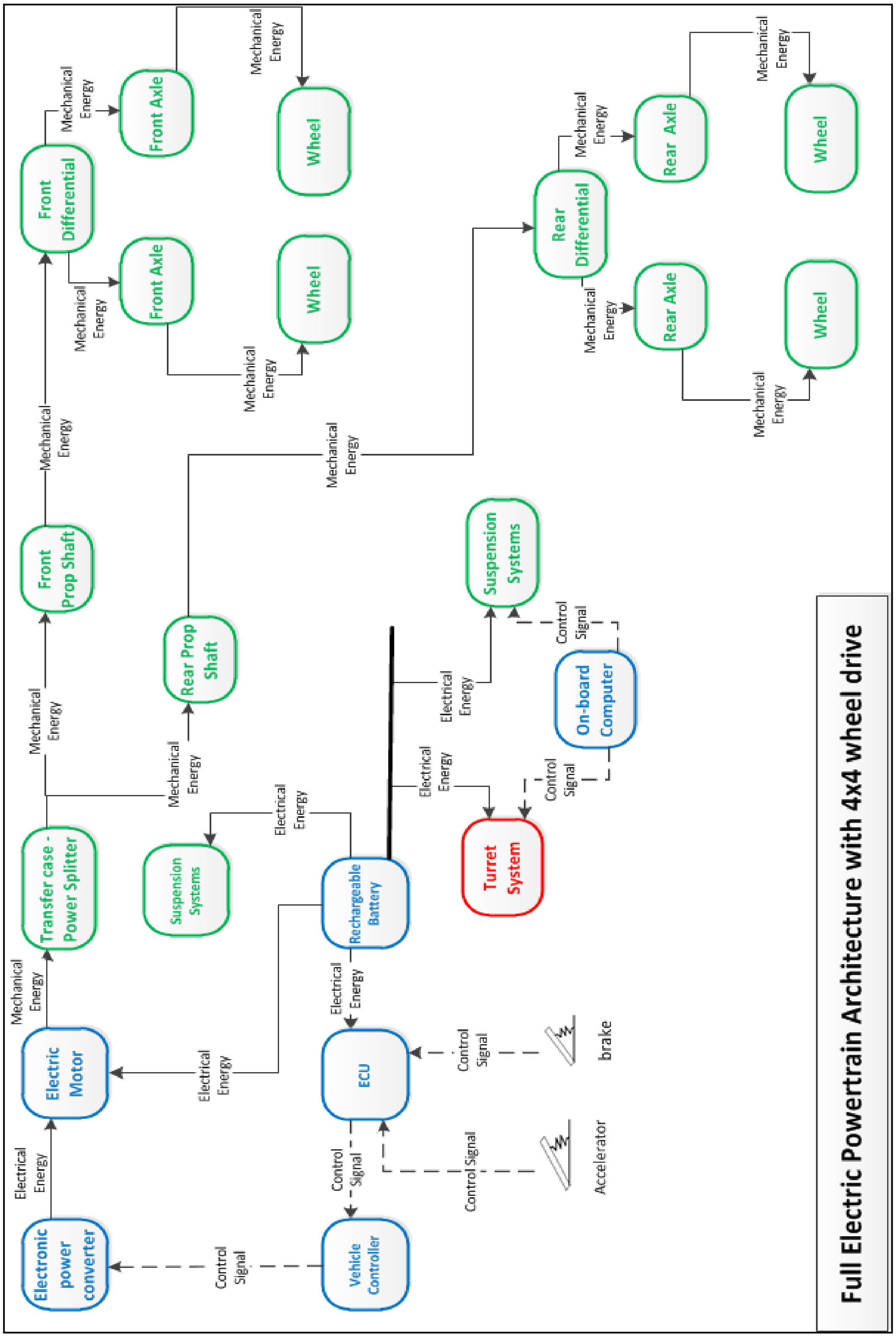


Figure 4-12-Full electric powertrain architecture with 4x4 wheel drive [37]

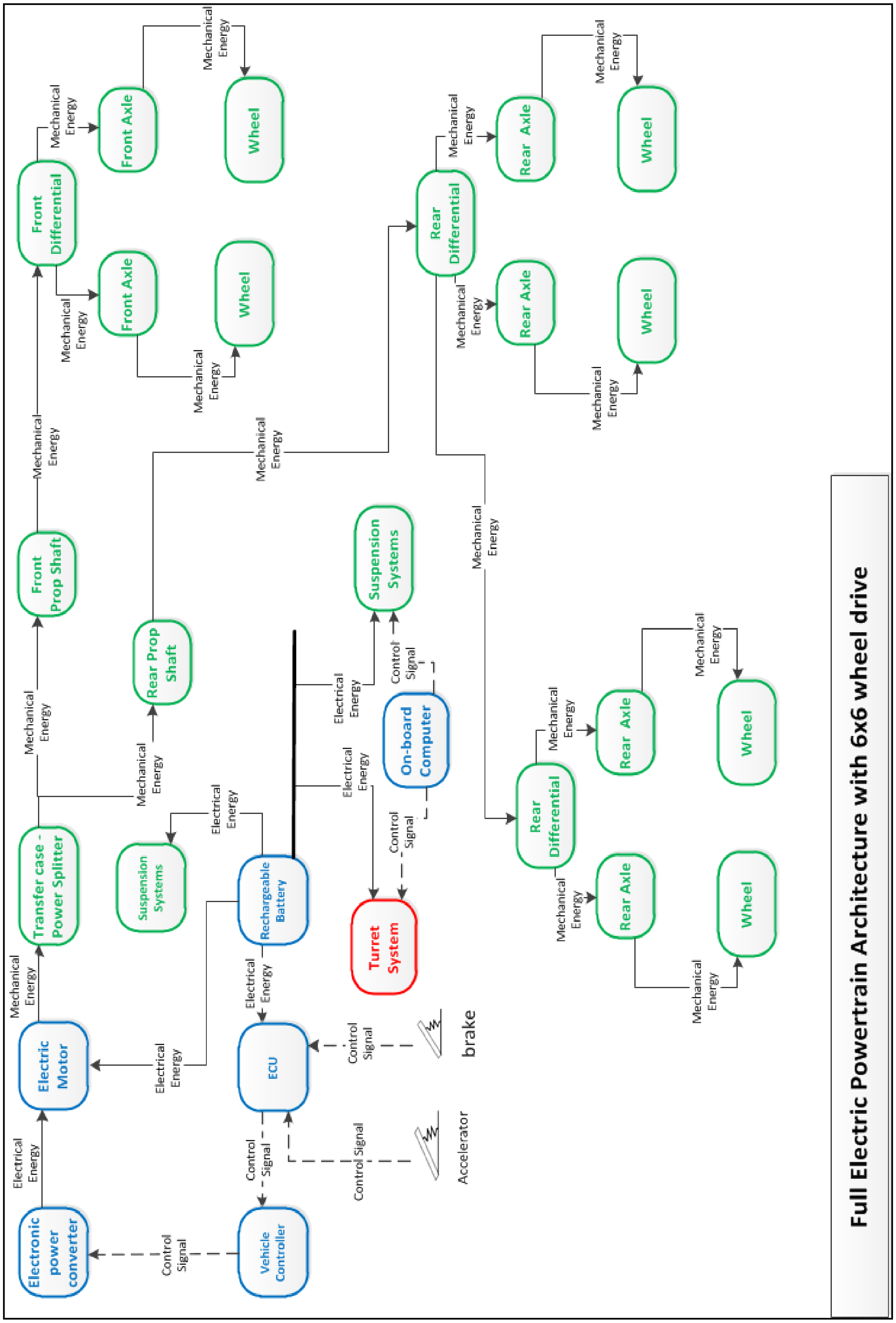


Figure 4-13- Full electric powertrain architecture with 6x6 wheel drive [37]

From the various Hybrid power train architectures, parallel hybrid powertrain was selected because it satisfies all functional requirements and combines the advantages of both parallel and series drive trains and therefore gives better results for both cruising stage and low speed stage. Full electric power train was eliminated because it will require a very large battery set which will add more weight to the design and render this alternative unfeasible. The conventional powertrain was not selected because it could not satisfy the range (300 miles) and fuel efficiency (2 mpg) requirements. Series-Parallel architecture was not selected because it requires a large battery and motor which will increase the total weight and affects the range of the vehicle. Series drivetrains gives good results at the low speed, but at cruising stage it becomes less efficient because of the more power loss due to converting mechanical power into electrical energy and then converting electrical energy back into mechanical energy.

Every system (block) labeled in the diagram is mapped to physical component or assembly. Some systems can be replaced with several components e.g. engine system can have 6 different type of engine. This situation leads to several design alternatives for parallel hybrid powertrain architectures. However some components are not compatible with each other e.g. caterpillar transfer case is compatible with only C9 and C7 caterpillar engines. Therefore some design alternatives for the Parallel hybrid powertrain gets eliminated. The total count of the feasible design alternatives comes out to be 522. The table 4.10 shows the list of possible components for each system in the hybrid power train architectures.

Table 4-10 components for each system in the schematic diagram [37]

System (Block)	**Components**
Differential	ZF end transmission (BK)
	ZF tru transmission (BK-DUA)
Engine	3616_Diesel (6169-7268 bhp @ 900-1000 rpm)
	C7_Diesel (210-325 bhp @ 2100 rpm)
	C9_Diesel (325-450 bhp @ 2100 rpm)
	C13_Diesel (385-520 bhp @ 1800-2100 rpm)
	C15_Diesel (475-580 bhp @ 1800-2100 rpm)
	C18_Diesel (600-700 bhp @ 1800-1900 rpm)
	D6V8 International (400-425 bhp @ 3000rpm)
ISG (integrated starter generator)	VU_ISG_V1 (130 KW)
	VU_ISG_V2 (195 KW)
	VU_ISG_V3 (253 KW)
	VU_ISG_V4 (130 kW)
Transfer case	Transfer case 455 (Gear ratio 3.088:1) aluminum case
	Transfer case 485 (Gear ratio 3.088:1) ductile iron case
Battery	Saft_HEMV_5 (220 V , 450A)
	Saft_HEMV_3 (220 V , 270A)
	Saft_HEMV_1 (220 V , 90A)
Transmission	Transmission CX28 (400 hp)
	Transmission CX31 (600 hp)
Super capacitor	PBM-27-64.8, 4x6, 24 cells, 64 V
	PBM-166/48.6, 3x6, 18 cells, 48 V
Converters	BDC2
	BDC4
	DC02
	PCM
	IPU160

4.5 Weighted Complexity metrics results

GME tool (developed by Vanderbilt) is used to generate 522 design alternatives for Parallel hybrid power trains. The design Alternatives has the components as nodes and connection between those components as links. The adjacency matrix created by GME tool were based on all the components a series-parallel hybrid vehicle uses i.e. drive train, chassis etc. For our complexity study we considered only power train components and those components have to be extracted from the entire list of components for each design alternative.

Table 4-11 shows a Reference matrix with all different components found in GME model for power train and a reference matrix of which components connects to which, to build Adjacency matrix for all design configurations. Multiple components were not considered e.g. there were 4 half shafts but only one is present in the reference matrix. Also there were some components (intermediate electrical connections etc.) which were not identified and taken into account.

Table 4-11 Reference matrix [37]

	(BK)	(BK-DUA)	ECU	Engine	ISG	Battery	Transfer case	Converters	Transmission for tires	Intermediate Drive Shaft	Half Shaft	Primary Drive Shaft
BK	Node wt	0	0	0	0	0	0	0	0	1	1	0
(BK-DUA)	0	Node wt	0	0	0	0	0	0	0	1	1	0
ECU	0	0	Node wt	0	1	1	0	0	0	0	0	0
Engine	0	0	0	Node wt	1	0	1	1	0	0	0	0
ISG	0	0	1	1	Node wt	0	0	0	1	0	0	0
Battery	0	0	1	0	0	Node wt	0	0	0	0	0	0
Transfer case	0	0	0	1	0	0	Node wt	0	1	1	0	1
Converters	0	0	0	1	0	0	0	Node wt	0	1	0	0
Transmission	0	0	0	0	1	0	1	0	Node wt	0	0	1
Intermediate Drive Shaft	1	1	0	0	0	0	1	1	0	Node wt	0	0
Half Shaft	1	1	0	0	0	0	0	0	0	0	Node wt	0
Primary Drive Shaft	0	0	0	0	0	0	1	0	1	0	0	Node wt

Links weight: The roles of components remain the same because only parallel hybrid powertrain is considered, but the component type is different as values of parameters/variables of components are not same. As the type of Components changes, its compatibility with the other components also changes i.e. it may be somewhat compatible, entirely compatible or not compatible at all with other components. This criterion is used to assign weights to the different interacting types of components to assign weights to the components. Weight 1 is given to the components that are compatible and weight 2 for somewhat compatible and 3 for incompatible. The compatibility data has been obtained from the specifications of different components (dependent on availability of the information about each component). Specs of all the components were not available so all those links were given a weight 1 and rest were assigned weights as per the given specification. There were no Alternatives for Drive shafts present so it is assumed the drive shafts used of any type will be compatible to all types of system used. Hence all the links to drive shaft are weighted as 1.

Table 4-12 Link weight based on compatibility [37]

Link From	Link to	Link wt.
BK	Intermediate shaft	1
BK	Half Shaft	1
BK-DUA	Intermediate shaft	1
BK-DUA	Half Shaft	1
ECU	Any ISG variant	1
ECU	Any Battery variant	1
Engine C7 or C9 Diesel	Transfer case (455 or 484)	1
Engine C13 diesel	Transfer case (455 or 484)	2
Engine (any)	Converter (any)	1
Engine(any)	ISG_V4	1
Engine C7 Diesel	ISG_V2	2
Engine C13 or C9	ISG_V2	1
Engine C7 Diesel	ISG_V3	3
Engine C9 Diesel	ISG_V3	2
Engine C13 Diesel	ISG_V3	1
Engine(any)	ISG_V1	1
ISG(any)	Transmission for tires (any)	1
Transfer Case	Transmission for tires	1
Transfer Case	Intermediate Drive shaft	1
Transfer Case	Primary Drive shaft	1
Converter	Intermediate Drive shaft	1
Transmission for tires	Primary Drive shaft	1

Node Weights: Node Weights are based on the power ratings, available types and information available.

a) Both differentials operate at nearly same torque and speed. The only difference is in their dimensions. So both are assigned weight 1.

b) Only one kind of ECU is available so the weight of ECU is taken as 1.

c) Engine weights are based on power ratings 100-350hp = 1 and > 350 hp =2

d) ISG weights are also based on power rating 100-200hp = 1 and > 200 hp =2

e) Battery weights are dependent upon the available types and Information. Weight 1 and 2 are given to batteries that provide 150 A and 450 A current respectively.

f) Transfer cases weights are dependent upon the material. Weight 1 is assigned for aluminum casing (for medium duty vehicles) and weight 2 is assigned for iron casing (for heavy duty vehicle).

g) Data for converters is not available so all of their weights were assigned 1.

h) Transmission for tires weight is also dependent on power rating and the types available. If power is less than 500hp, weight 1 is assigned otherwise (if power is greater than 500 hp) weight 2 is assigned.

i) Superchargers weight depends upon their cell numbers and voltage values. Super capacitor that 24 cells and gives 64V is given weight 2.

Table 4-13 Node weight based on parameter values [37]

Component	**Component Type**	**Weight**
Differential	ZF end transmission (BK)	1
	ZF tru transmission (BK-DUA)	1
ECU	ECE Hybrid Controller	1
Engine	3616_Diesel (6169-7268 bhp @ 900-1000	2
	C7_Diesel (210-325bhp @ 2100 rpm)	1
	C9_Diesel (325-450 bhp @ 2100 rpm)	1
	C13_Diesel(385-520 bhp @ 1800-2100	2
	C15_Diesel(475-580 bhp @ 1800-2100	1
	C18_Diesel (600-700 bhp @ 1800-1900	1
	D6V8 International (400-425 bhp @	2
ISG	VU_ISG_V1 (130 KW)	1
	VU_ISG_V2 (195 KW)	1
	VU_ISG_V3 (253 KW)	2
	VU_ISG_V4 (130 kW)	1
Battery	Saft_HEMV_5_One (220 V , 150A)	1
	Saft_HEMV_5_Three (220 V , 450A)	2
Transfer case	Transfer case 455 (Gear ratio 3.088:1)	1
	Transfer case 485 (Gear ratio 3.088:1)	2
Converters	BDC2	1
	BDC4	1
	DC02	1
	DC07	1
	dc12	1
	IPU160(2)	1
	ipu160	1
	PCM	1
	PCM-hr	1
Transmission for tires	Transmission CX28 (400 hp)	1
	Transmission CX31 (600 hp)	2
Super capacitor	PBM-27-64.8, 4x6, 24 cells, 64 V	2
	PBM-166/48.6, 3x6, 18 cells, 48 V	1

4.5.1 Cost computation of design alternatives

Design alternatives can also be evaluated on the basis of cost. The total cost in general is a function of procurement cost, manufacturing cost, assembly cost and service cost. Since most of the hybrid powertrain components are COTS (component off-the-shelf), we only consider the procurement cost. All other costs are assumed to be equal for all design alternatives. A master bill-of-material for all the possible configurations in the GME is obtained. The configurations contain components that are both powertrain and non-powertrain. Only powertrain components are considered for cost computation in this study. The cost of each of the individual component is obtained from web catalogs of parts and only parts stores of the OEMS. The master BOM with components is shown in table 4-14.

Table 4-14 Cost of the hybrid powertrain components [37]

Component	Component Type	Price ($)
Differential	BK	4800
	BK-DUA Front Mid	5000
	BK-DUA Rear Mid	5000
ECU	ECE Hybrid Controller	1000
Engine	3616_Diesel	40000
	C7_Diesel	11075
	C9_Diesel	13000
	C13_Diesel	15500
	C15_Diesel	18000
	C18_Diesel	22000
	D6V8 International	12000
ISG	VU_ISG_V1	11000
	VU_ISG_V2	11500
	VU_ISG_V3	12000
	VU_ISG_V4	12500
Battery	Saft_HEMV_5_One	2500
	Saft_HEMV_5_Three	4000
Transfer case	TC455	4500
	TC485	5000
Converters	BDC2	3000
	BDC4	3500
	DC02	4000
	DC07	4500
	dc12	6000
	IPU160	7000
	PCM	7500
	PCM-hr	8000
Transmission	CX28	13900
	CX31	15000

The BOM for each design alternative is extracted from the GME. The design alternative cost is computed by referring the master BOM and assigning component cost for those components present in the BOM. The design configuration 1 BOM and the cost of each component are shown below:

Table 4-15 Cost computation of Design alternative 1 [37]

S.no.	Hybrid powertrain components	Cost
1	C13_Diesel	15500
2	ECE_Hybrid_Controller	1000
3	455 (transfer case)	4500
4	Saft_HEMV_5_One	2500
5	VU_ISG_V1	11000
6	Saft_HEMV_5_Three	4000
7	SAFT HEMV-5 Rack	5500
8	BK-DUA Rear Mid	5000
9	BK (differential)	4800
10	BK (differential)	4800
11	BK-DUA Front Mid	5000
12	CX31 (transmission)	15000
Total cost		78600

A MATLAB program listed in Appendix E computes the cost of every the design Alternative. Total cost of all configurations is given in Appendix G. A graphical representation for comparing design alternatives with respect to complexity and cost is shown below. The complexity value for each alternative is obtained by simply summing up the values of all the complexity metrics. Since the designs are the alternatives of same parallel hybrid powertrain there is not much difference in the cost and complexity values. Therefore many design points are plotted close to or over each other in the plot.

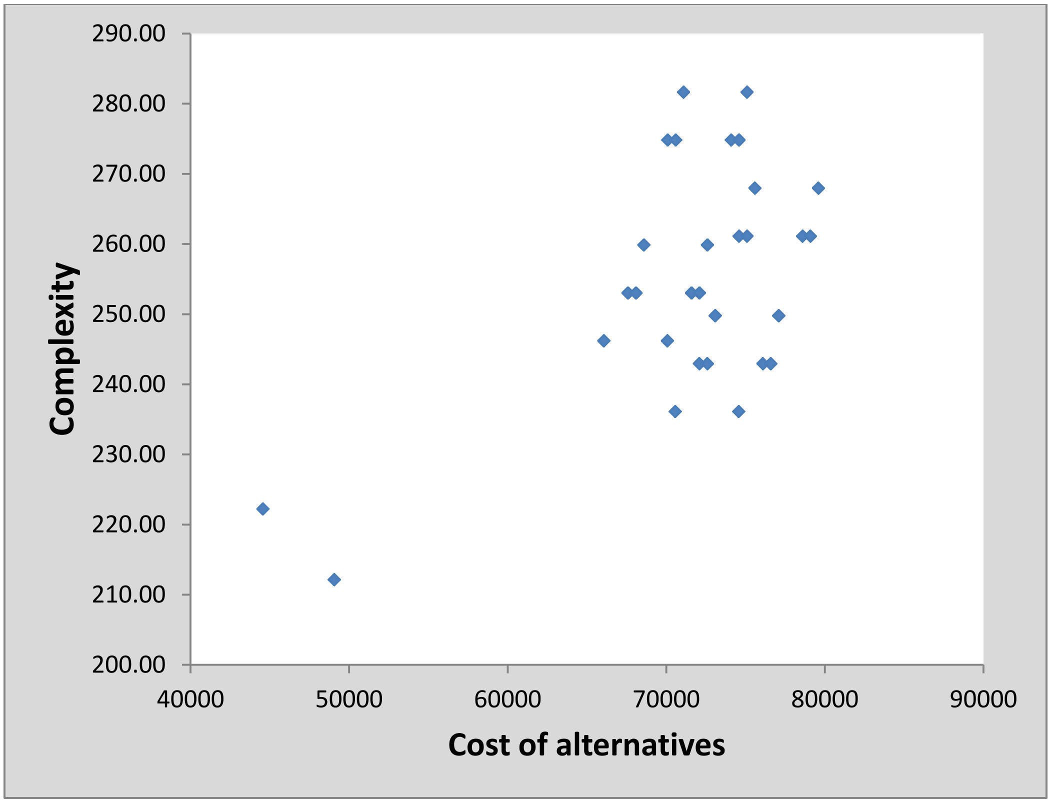


Figure 4-14 Cost versus performance plot

4.5.2 Evaluation of complexity metrics with respect to the application and mathematical properties

Table 4-16 Evaluation of complexity metrics

	Holistic	Correlate reality	Consistent	Sensitive	Repeatable	Scalable	Monotonicity	Transitivity	Non-dimensional	Continuity	Invariant to affine transformations
Node	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Links	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
a_{AVG}	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓

a_{MAX}	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
Conn.	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
I_{VD}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NI_{VD}	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
C.C.	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
Cost	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Time	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mob	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Total size	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Node wt	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Link wt	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
wt conn.	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
wt CC	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
Wt I_{vd}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
wt NI_{vd}	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓
wt mob	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Wt node degree	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓

4.5.3 Inferences from the results

As all of these 522 design configurations are the alternatives of a same system (Parallel Hybrid drive train) so there is not a big difference in the complexity metrics' results. Also in place of design variables the components and there inter compatibility is used to define the nodes and links among them so it gives a complexity measure of Assembly (i.e. physical structure).

The best way to find out the authenticity of the complexity metrics is to check how they relate with the cost of the alternatives. A correlation study is performed between the complexity metrics and cost of the alternatives. The result of the correlation study is shown in figure 4-5.

	links (L)	N+L	Avg Node weight	Avg link weight	Avg Node degree	Connectedness	lvd	Nivd	Mobility	WeightedConnectedness	weighted CC	nodes weight(N)	links weight (L)	Total Weighted size	wt lvd	wt Nivd	wt avg node degree	wt Mobility	Cost
links (L)	1.00																		
N+L	1.00	1.00																	
Avg Node weight	0.42	0.42	1.00																
Avg link weight	0.16	0.16	0.64	1.00															
Avg Node degree	1.00	1.00	0.42	0.16	1.00														
Connectedness	1.00	1.00	0.42	0.16	1.00	1.00													
lvd	1.00	1.00	0.42	0.16	1.00	1.00	1.00												
Nivd	1.00	1.00	0.42	0.16	1.00	1.00	1.00	1.00											
Mobility	1.00	1.00	0.42	0.16	1.00	1.00	1.00	1.00	1.00										
WeightedConnectedness	0.47	0.47	0.71	0.95	0.47	0.47	0.47	0.47	0.47	1.00									
weighted CC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00								
nodes weight(N)	0.42	0.42	1.00	0.64	0.42	0.42	0.42	0.42	0.42	0.71	0.00	1.00							
links weight (L)	0.47	0.47	0.71	0.95	0.47	0.47	0.47	0.47	0.47	1.00	0.00	0.71	1.00						
Total Weighted size	1.00	1.00	0.42	0.16	1.00	1.00	1.00	1.00	1.00	0.47	0.00	0.42	0.47	1.00					
wt lvd	0.32	0.32	0.93	0.84	0.32	0.32	0.32	0.32	0.32	0.86	0.00	0.93	0.86	0.32	1.00				
wt Nivd	0.32	0.32	0.93	0.84	0.32	0.32	0.32	0.32	0.32	0.86	0.00	0.93	0.86	0.32	1.00	1.00			
wt avg node degree	0.47	0.47	0.71	0.95	0.47	0.47	0.47	0.47	0.47	1.00	0.00	0.71	1.00	0.47	0.86	0.86	1.00		
wt Mobility	0.71	0.71	0.94	0.56	0.71	0.71	0.71	0.71	0.71	0.73	0.00	0.94	0.73	0.71	0.84	0.84	0.73	1.00	
Cost	0.82	0.82	0.38	0.37	0.62	0.32	0.87	0.78	0.85	0.60	0.00	0.38	0.60	0.82	0.86	0.72	0.60	0.80	1.00

Figure 4-15 Correlation table [37]

Form the correlation study it was found that Mobility, Information content, weighted information content and total weighted size are the metrics that somewhat represents a good measure for complexity. As the correlation value of these metrics with cost is more than 0.8, these metrics correlated well to the reality. Average clustering coefficient (a measure of the interaction among the nodes and does not takes into account the weight of links or nodes) shows bad correlation values which do not affect the complexity. The Weighted Avg. node degree and weighted Connectedness also have a fair correlation with the cost as their correlation value is more than 0.6. Average node weight, average link weight and connectedness are correlating very poorly with cost because the correlation values for these measures are less than 0.4.

4.6 Design and Implementation of Matlab code for weighted complexity metrics: The program is divided into following five subtasks (functions):

1) Generate the graph in which nodes (variables) and links can be display and edit. The values of weights assigned to the nodes and links could be introduced by importing an input text file or from the interaction with the user.
2) Populate the adjacency matrix from the node and links in the graph.
3) Calculate the weighted and unweighted complexity metrics from the adjacency matrix.
4) Show the results and comparison charts of complexity metrics.
5) Save the adjacency matrix and results to a file so that it can be read later.

4.6.1 Weighted adjacency matrix: The adjacency relation (a_{ij}) is specified by non-zero value if two nodes are connected and value 0 if the nodes are not connected. The matrix containing all adjacency relations in a graph is called adjacency matrix. If n is number of nodes then size of the adjacency matrix is (n x n). Undirected graphs have adjacency matrixes that are symmetrical with respect to their main diagonal. On the other hand the adjacency matrixes of directed graphs are not symmetric.

Unlike the unweighted adjacency matrix, the diagonal of weighted adjacency matrix contains non-zero values. The diagonal of the weighted adjacency matrix contain the information about the nodes weights. The non-diagonal elements of the weighted adjacency matrix contain the information about the link weights.

4.6.2 Implementation: The weighted complexity program is divided into the following 5 functions:

1) adj_matrix_gui: This function creates the graph from the beginning or from the old saved file. A node can be created by left double click (LMB) followed by entering the value of node weight in the command window. A link can be made between 2 nodes by first single click (LMB) on first node and then single click (LMB) on the second node followed by entering the value of link weight in the command window. A node or link can be deleted by single right click (RMB) on them. This function also generates the adjacency matrix from the graph. This function was built by updating the base function, *Adjacency Matrix GUI*, present at the Mathwoks website.

2) calc_metrics: This function calculates all the complexity metrics from the weighted adjacency matrix. This function also calls other functions, such as

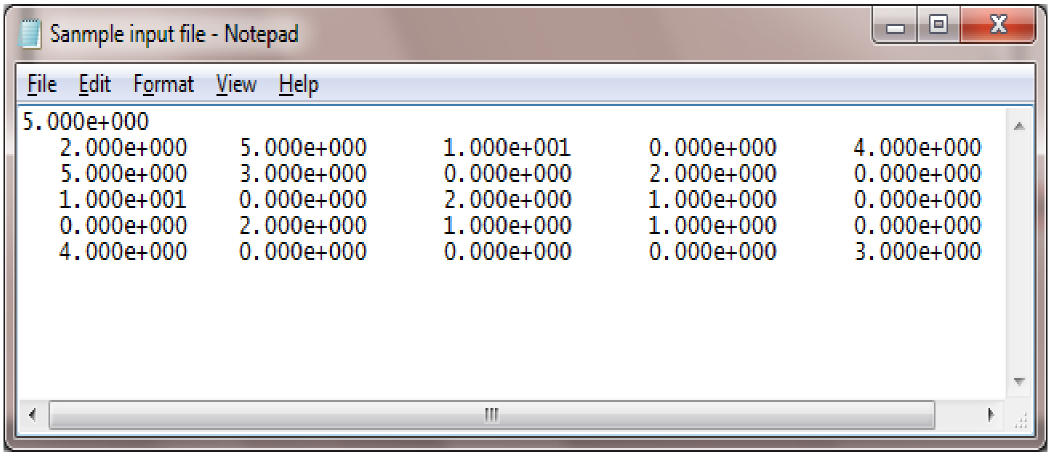
bridgeNode, pruneingleafnodes, connection etc., to calculate the weighted complexity matrices.

3) OpenData: This function is called by the function adj_matrix_gui to create the graph from the old existing file. This function reads the size i.e. number of nodes in the input file and creates a graph in which all the nodes are placed at the vertex of a polygon. The nodes are then connected by lines in the graph according to the values present in the adjacency matrix. Nodes i and j are dependent if value at a cell a(i, j) of adjacency matrix is non-zero and it is represents as a link is plotted between nodes. Nodes i and j are independent if value at a cell a(i, j) of adjacency matrix is zero and it is represents as a no link between the nodes.

4) Plot charts: The role of this function is to read the results from function calc_metrics and display the results in message box and also create a bar-chart of the results.

5) OutPutVertex_Matrix_ToFile: This function is used for saving the graph. It writes information about the number of nodes, weight of the nodes, number of links and weight of the links to the output file.

4.6.3 Format of input file: First line of the input file has the information about the total number of nodes in the graph. Weighted adjacency matrix is stored in the input file after the first line. Node weights are stored at the diagonal of the weighted adjacency matrix. Link weights are stored at the non-diagonal cells. A sample input file is shown below in figure 4-6.



```
5.000e+000
   2.000e+000     5.000e+000     1.000e+001     0.000e+000     4.000e+000
   5.000e+000     3.000e+000     0.000e+000     2.000e+000     0.000e+000
   1.000e+001     0.000e+000     2.000e+000     1.000e+000     0.000e+000
   0.000e+000     2.000e+000     1.000e+000     1.000e+000     0.000e+000
   4.000e+000     0.000e+000     0.000e+000     0.000e+000     3.000e+000
```

Figure 4-16 sample input file for weighted complexity matrices

The detail Matlab code of main file and all the function files is included in APPENDIX E.

CHAPTER 5

COMPLEXITY DUE TO MIXED DISCIPLINES

To this point we considered that variables from all the possible disciplines and domain as a part of one whole functional structure. The limitation of this method is that some variables are shared by more than one discipline and these variables are more important than others. Assigning weights to the all variables according to its parameter value can give accurate value of complexity metrics for a single discipline but not for the overall complexity.

Consider an aircraft engine that provides thrust to the aircraft. Designing the engine is very complex affair because it not only has components such as compressors, turbine, combustion chambers, nozzle, afterburners but also noise suppression system, lubrication system, pollution control system. Therefore it encompasses the disciplines such as heat transfer, fluid dynamics, aerodynamics, acoustics and electrical. Some variables like pressure of the air are shared by different system like compressors, combustion chamber, turbines and afterburners. There is no doubt that these variables are more important than other variables in the complete engine design. Therefore there is need of a procedure that can compute complexity metrics of individual disciplines, interaction between the disciplines and the overall complexity of a given design problem.

5.1 Functional graph representation for multi-domain artifact

The development of a typical cyber-physical product takes account of the principles, formulae and applications of several disciplines. Some Possible disciplines are:

1. Structural
2. Heat transfer (Thermodynamics)
3. Fluid dynamics
4. Electrical
5. Computer science (Soft-wares and programs)

There are 2 methods for representing the multi-domain graph:

5.1.1. L3D (Layers in three dimensions) method for multi-domain graph representation: In this method, every discipline will have its own separate layers in the space. For example, if there are n disciplines then there will be n number of layers. All the variables (nodes) related to a particular relationship will be on a unique discipline layer and links will connect these variable based on the relationship information. For example, a graph is shown in figure 5 for the product which has 3 disciplines.

Limitations:

a) If a variable is common to two or more than two discipline then there will be ambiguity about in which layer that variable should be place.

b) This representation scheme would be very difficult to develop in C++.

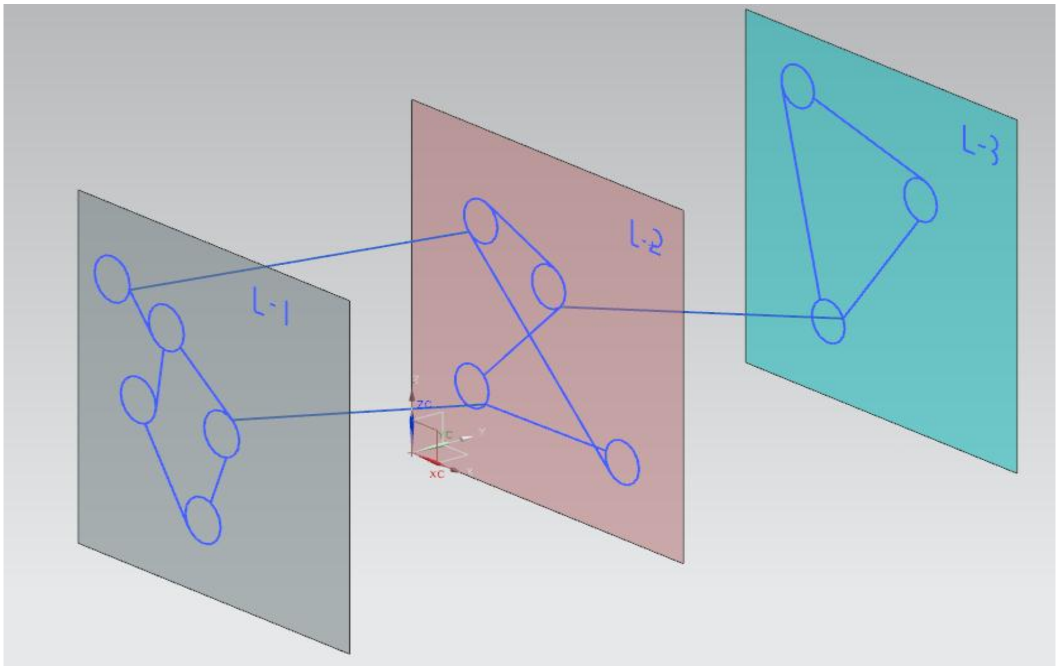


Figure 5-1 Layered representation of discipline oriented graph

5.1.2. B2D (Blocks in two dimensions) method for multi-domain graph representation: In this method, disciplines will be represented by a blocks in two dimension. Every discipline is represented by a unique block. These blocks can also overlap with each other so that variables common in two or more disciplines can be placed in the area that is common to those blocks. An example of this type of representation scheme is shown below in figure 6.

Advantages:

1. This type of representation will be easy to implement as compare to Layer type representation.
2. Node shared by two or more discipline can be easily represented without duplicating with this method.

3. This representation will be easy to interpret because it doesn't have to rotate so that one can see the node and links.

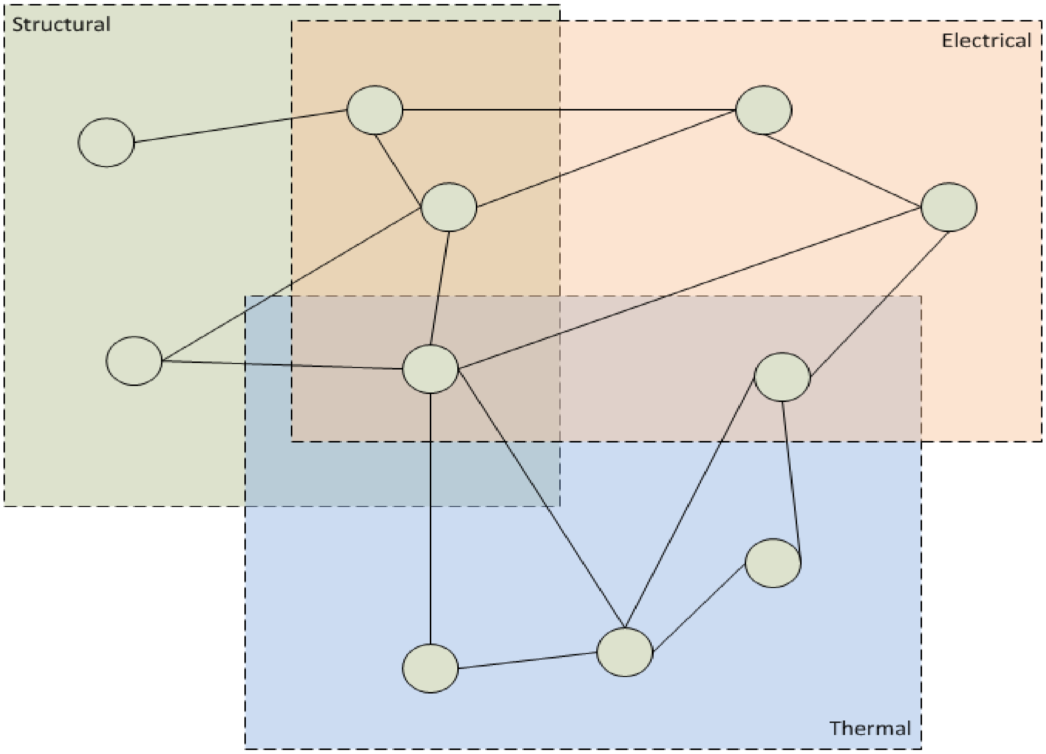


Figure 5-2 Block representation scheme

5.2 Multi-domain interaction complexity

By using B2D representation, complexity matrices can be computed for the entire structure as well as for the individual discipline. From these complexity metrics size, coupling and solvability of a product can be easily calculated. Diversity in design i.e. number of different disciplines involved is one more measure of complexity which has not been computed yet.

According to the survey conducted on the experts, it was found that the core reasons for the diversity complexity are the number of different disciplines involved in the design process and the interaction between the disciplines. As the number of disciplines for a product increases, the product becomes more and more complex. The complexity metrics for an individual discipline can be calculated easily by separately constructing the adjacency matrix for that discipline. Now the main problem is to calculate the interaction between the disciplines. It can be easily computed by using the set theory.

For example, suppose all the variables in a functional chart belong to two disciplines A and B as shown in the figure 7.

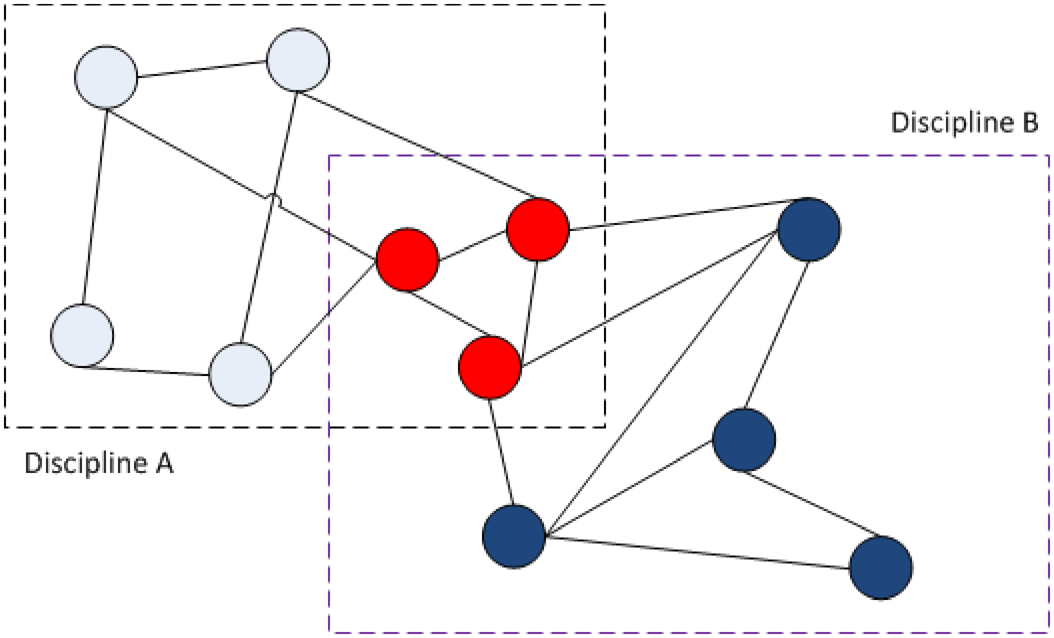


Figure 5-3 Random graph with two domains

Assume the complexity metrics for these individual disciplines is given by nA and nB. If we compute the complexity metrics of the entire functional graph, as we did in the experiment 1-mechanical transmission and experiment 2- hybrid powertrain architecture, then we will get the combined complexity i.e. n(AUB). The interaction between the disciplines i.e. n(A∩B) can be computed by summing up the complexity of all the individual disciplines complexity and then subtracting the combined complexity.

$$\mathbf{n(A \cap B) = (nA + \boldsymbol{n}B) - n(AUB)}$$

So far we have calculated the combined matrix in the experiment 1-mechanical transmission and experiment 2- hybrid powertrain architecture. This combined complexity should not be confused with the overall complexity of the design process. The overall complexity of the design process can be measured by summing up the complexity of all the individual disciplines or the sum of combined complexity and interaction complexity.

$$\mathbf{Overall\ complexity = nA + \boldsymbol{n}B = n(AUB) + \ n(A \cap B)}$$

For the figure 1, if we look at the size metrics for nodes then

Nodes in the graph (size complexity) for Individual discipline

nA = 7 nodes

nB = 7 nodes

Combined complexity n(AUB)= 11

Interaction complexity n(A ∩ B) = 3

Overall complexity= 7+7 = 11+3 = 14

Similarly for the links

Nodes in the graph (size complexity) for Individual discipline nA = 10 links

nB = 11 links

Combined complexity n(AUB) = 18

Interaction complexity $n(A \cap B) = 3$

Overall complexity= 10+11= 18+3 = 21

Similarly interaction complexity for all the complexity metrics can be calculated by this method. One important property of interaction complexity is it should be more than equal to zero i.e.

Interaction complexity $n(A \cap B) >= 0$

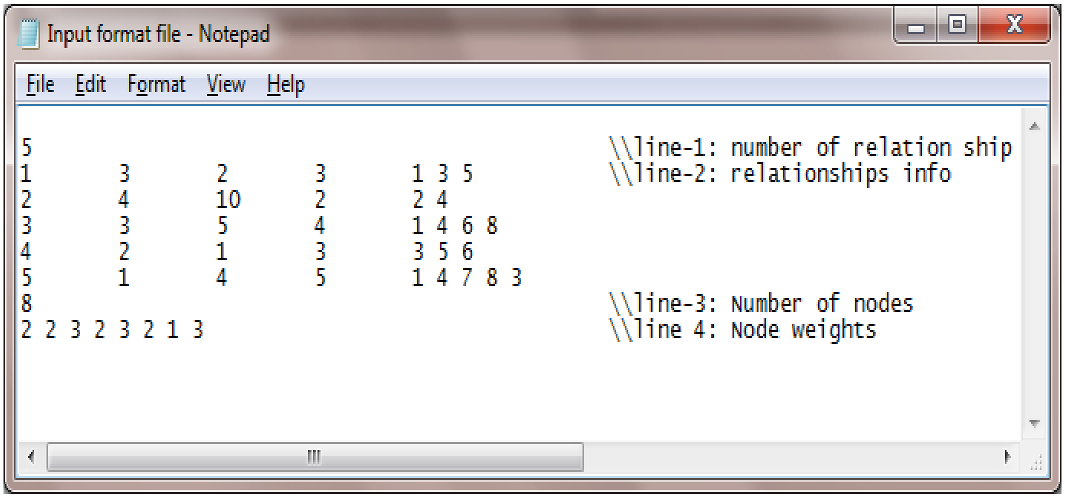
This property can be used to identify the good complexity measures from the bad ones. If any complexity measure gives the negative value for the interaction complexity then that complexity measure is not a genuine complexity metrics.

5.3 Design of software for functional graph representation of mixed disciplines:

The adjacency matrix of weighted complexity metrics was created based on the type of connection between the nodes i.e. link weight and variable values i.e. node weight. For creating the functional graph, information about the discipline type of the relationships is also required. Moreover the adjacency matrix of a design problem can be very big which makes it difficult to read and understand. So a new compact method to define the matrix is to use the relation (formulae) that connects the nodes (variables). Besides node and link information, additional information required for this new representation is the discipline type of relationship.

Input file format

The input file format will have information about the number of relation, relationship number, relationship discipline, links weight, # of nodes in a relationship, node # and node weight. A possible file format is shown below in figure 1. The first value of the line-1 will give the number of relationships which is 5 in this example. A loop is formed using this number to read the next 5 lines. These lines will have relationship number, relationship_discipline #, link's weight, number of node in the relation and node number respectively. Line 3 will give total number of nodes and line-4 will give the node's weight respectively.



```
5                       \\line-1: number of relation ship
1   3   2   3   1 3 5   \\line-2: relationships info
2   4   10  2   2 4
3   3   5   4   1 4 6 8
4   2   1   3   3 5 6
5   1   4   5   1 4 7 8 3
8                       \\line-3: Number of nodes
2 2 3 2 3 2 1 3         \\line 4: Node weights
```

Figure 5-4 Input file format

Relationship_disciplines# is determined by the scheme given in table 5-1.

Table 5-1 determining the Relationship_disciplines# from the relationship type

Relationship type	**Relationship_disciplines #**
Structural	1
Thermodynamics	2
Fluid	3
Electrical	4
Software (computer science)	5

CHAPTER 6

CONCLUSION

The complexity of the product, process or problem should be estimated as early as possible in the product development phase, otherwise it will result in higher development time and cost. Complexity must also be considered in trade-offs involving performance and cost. At least, it should be made apparent to decision makers what they are paying for additional complexity to get marginal improvements in performance.

A holistic approach for measuring complexity is mentioned in this thesis. The proposed complexity metrics look at product architecture, process complexity and manufacturing complexity. Complexity metrics are grouped into three categories: system size, degree of coupling between the elements at different development stages, technological complexity that comes from technological gaps. Both data structure and fast algorithms for complexity calculations for large cyber physical systems were also provided. A rational basis for assigning weights to entities and relations for metric calculations was developed.

Only two experimental studies have been conducted due to limited time and resources. The lack of project and manufacturing data from real programs prevented us from more comprehensive validation. Moreover, for experiment 2 the metrics were calculated only at the component connectivity level, that is, they only considered physical complexity and not functional.

The weighted metrics correlate well with cost and expert ratings. The only exception is the weighted average cluster co-efficient and it should be eliminated. Total weighted size, weighted Mobility, and weighted information content appear to be best suited for measuring complexity.

6.1 Future Work: complexity measurement is a very intricate but practical research area. Although a lot of work has been done in the complexity measurement research but it has solved only few issue of the research. Future work may include the following aspects:

1) **Defining the abstraction levels for different stages of design:** In the first experiment the complexity metrics are calculated at component level and all the design variables for every component were considered. But in the second experiment complexity metrics are calculated at system level. Since many components of the shelf (COTS) were used in hybrid power-train, all design variables for every component were not considered. Design variables that interact at system level were considered for the complexity metrics calculations. Therefore there is a need of defining the abstraction levels and at which variables can be considered and eliminated for the complexity metrics calculations.
2) **Develop an algorithm for computing the number of cycles in graph:** Number of cycles in artifact functional structure and design process structure is a decent candidate of complexity metrics. As the number of links increases, the cycle computation of a graph takes a more and more

computer memory and time. There are several algorithms for computing the number of cycles in the graph, such as backtracking algorithm and Depth first search (DFS) algorithm, but none of them is very efficient. So an algorithm needs to be developed that can compute the number of cycles very quickly and accurately in functional structure and design process graphs.

3) **Development of the software that can compute the complexity of the mixed disciplines:** A design of the software to calculate complexity of the mixed disciplines was discussed in the fifth chapter. However the interactive part of the proposed software is very intricate to develop in Matlab. This is because the software will require object oriented programming which is not present in Matlab. Therefore this software can only be developed in a language that can support object oriented programming such as Visual C++ or JAVA.

4) **Assigning weights to the complexity metrics to get one overall value of complexity:** while plotting cost versus complexity plot in the experiment 2, all the complexity metrics are simple added i.e. all complexity metrics are given weight 1. However some complexity metrics, such as mobility and information content, are more significant than others. So a technique of assigning weights to complexity metrics need to be develop so that significant complexity metrics can contribute more to the complexity results than non-significant complexity metrics.

REFERENCES

[1] U.S. Government Accountability Office, GAO-09-326SP Defense, “Acquisitions: Assessments of Selected Weapon Programs”, 2009, http://www.gao.gov/new.items/d09326sp.pdf

[2] Teseon, “TESEON - Complexity Management”, http://www.teseon.com

[3] Braha, D., and Oded Maimon, 1998, “The Measurement of a Design Structural and Functional Complexity” IEEE Transactions. Systems, Man and Cybernetics, Part A. Syst. Humans, 28_4_, pp. 527–535.

[4] F.G. Wikie, B.A. Kitchenham, “Coupling measures and change ripple in C++ application software”, Journal of Systems and Software, Volume 52 Issue 2-3, June 1 2000

[5] Briand, L.C., Daly, J.W., Wust, J., 1996. “A unified framework for coupling measurement in object-oriented systems”. Technical Report number ISERN-96-146, Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany.

[6] Bashir, H., and Thomson, V., 2001, “Models for Estimating Design Effort and Time,” Design Stud., 22_2_, pp. 141–155.

[7] Li Chen, and Simon Li, S., 2005, “Analysis of Decomposability and Complexity for Design Problems in the Context of Decomposition,” Journal of Mechanical Design, 1274, pp. 545–557.

[8] Summers J, Shah J, “Mechanical engineering design complexity metrics: Size, coupling and solvability”, J. Mechanical Design, ASME Transactions, V132, Feb., 2010.

[9] Nam P Suh, 1990, “The Principles of Design”, 1st edition: Oxford University Press, New York, NY.

[10] C. E. SHANNON and W. WEAVER, “The Mathematical Theory of Communication”. University of Illinois Press, 1949.

[11] Shah J, George Runger, “Misuse of information-theoretic dispersion measures as design complexity metrics”.

[12] Marc H. Meyer and Kathleen Foley Curley, “An Applied Framework for Classifying the Complexity of Knowledge-Based Systems”

[13] Srinath Balaji, Gurpreet Singh, Jami Shah, “Experiment 1: Mechanical Transmissions Complexity Analysis”, DARPA-META2 Project, Technical Report, ASU/DAL/META/2011-01R3

[14] D. Bonchev and G. Buck. Complexity in Chemistry, Biology and Ecology, chapter 5 Quantitative Measures of Network Complexity, pages 191–235, Springer, New York, 2005.

[15] Steve Chuang, “Adjacency Matrix GUI”, http://www.mathworks.com/matlabcentral/fileexchange/6937-adjacency-matrix-gui

[16] Srinath Balaji, Gurpreet Singh, Jami Shah, Peng Zhao, Prashant P, “Report 2: Complexity Metrics for Cyber-Mechanical Systems”, DARPA-META II Project, Technical Report, ASU/DAL/META/2011

[17] Phillip Bonacich, “Some unique properties of eigenvector centrality”, Department of Sociology, University of California at Los Angeles, Los Angeles, CA 90095, United States (2007)

[18] Jaroslaw Sobieszczanski Sobieski, “On the sensitivity of complex internally coupled systems”, NASA Tech Memo 100537, 1988.

[19] Douglas C. Montgomery, Design and analysis of experiments [Book] (7th edition)

[20] Lindemann, Udo, Maurer, Maik, Braun, Thomas, “Structural complexity management- An Approach for the Field of Product Design”, Springer, 2009

[21] El Haik, Kai Yang, “The component of complexity in engineering design” IIE transection, 31, pp. 925-934, 1999.

[22] Frizelle G, “An entropic measurement of complexity in manufacturing operations”, Technical report, Department of manufacturing operations, proceeding of the royal society London, Vol. 457, 2001.

[23] A calinescu, J Efstathiou, J Schirn and J Bermejo,” Applying and assessing two methods for measuring complexity in manufacturing”, Journal of operational research society, pp. 723-733, 1998.

[24] Deshmukh, Joseph J. Talavage, “Complexity in manufacturing systems, Part 1: Analysis of static complexity”, IIE Transactions (1998) 30, pp. 645-655

[25] Boothroyd, Dewhurst, Knight, “product design for manufacture and assembly”, Marcel Dekker, 1994

[26] Rodriguez-Toro1 C Tate S Jared G and Swift K, “Complexity metrics for design (Simplicity + Simplicity = Complexity)”, Proc. Instn Mech. Engrs Vol. 217 Part B: J. Engineering Manufacture.

[27] Banker, R.D., Datar,S.M., Kekre, S. Mukhopadhyay, “Cost of product and process complexity”, Measures of manufacturing excellence, Harward business school press, Cambridge, Massachusetts, pp269-290, 1990

[28] László Sors, László Bardócz, István Radnóti, “Plastic molds and dies, Published by Van Nostrand, Reinholdcompany and Akademiai Kiado, pp. 376-391, 1981

[29] Jami Shah, Paul K. Wright, “Developing Theoretical Foundations of DFM”, Proceedings of DETC2000: Design for manufacturing conference, Sept 2000, Baltimore, MD

[30] Corbett J., Dooner M., Meleka J., Pym C., “Design for Manufacture: Strategies”, Principles and Techniques, Addison Wesley, 1991.

[31] Trucks, H., 1987, Designing for Economical Production, Society of Manufacturing Engineers, Dearborn, MI.

[32] Boothroyd G., Dewhurst P., “Design for Assembly” .Handbook, ME Dept., Univ. of Massachusetts, 1983

[33] Esawi, A. M. K., and Ashby, M., “Cost Estimation for Process Selection””, Proceedings of ASME Design for Manufacture Conference, September 1999, Las Vegas, NV.

[34] Smith, C., “The Manufacturing Advisory Service: Web Based Process and Material Selection”. U C Berkeley Mechanical Engineering Depart., Ph.D. Thesis, August 1999.

[35] Elmaraghy H., Knoll L., Johns B., “Design of Assemblies of a DC Motor”, Design for Manufacture: Strategies, Principles and Techniques”, Addison Wesley, 1991.

[36] Dept. of Navy, “Producibility Measurement Guidelines”, Final Draft, Dec 1991.

[37] Srinath Balaji, Prashant Mohan, Gurpreet Singh, Mahmood Dinar, Jami Shah, “Complexity Report 3”, DARPA-META II Project, Technical Report, ASU/DAL/META/2011

[38] Skiena, S. “Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica”, MA: Addison-Wesley, p. 213, 1990.

[39] Whitney, H. (1932), "Congruent graphs and the connectivity of graphs", American Journal of Mathematics 54 (1): 150–168

[40] S.G.Shirinivas, S.Vetrivel, Dr. N.M.Elango “Applications of Graph Theory In Computer Science An Overview”, International Journal of Engineering Science and Technology, Vol. 2(9), 2010, 4610-4621

APPENDIX A

FUNCTION STRUCTURES (NETWORK REPRESENTATION) OF TRANSMISSION ALTERNATIVES (REF [13])

I. Design variable list and node numbers

Design 1

Variable	#
b	1
b_{max}	2
b_{min}	3
K_s	4
K_m	5
K_v	6
V_s	7
V_g	8
HP_{out}	9
F_{gt}	10
F_{wa}	11
F_{wt}	12
F_{ga}	13
F_{wr}	14
F_{gr}	15
e	16
HP_{in}	17
Brg -2	18
Brg -1	19
n_w	20
n_g	21
m_G	22
N_w	23
N_g	24
C	25
d_w	26

Design 2

Variable	#
Brg – 2	1
Key – 2	2
d_{s3}	3
Brg - 3	4
Key - 3	5
HP_{in}	6
n_{p1}	7
n_{g2}	8
m_G	9
S_{N2}	10
d_{s2}	11
$(F_r)_{p2}$	12
$(F_a)_{p2}$	13
$(F_a)_{g2}$	14
$(F_r)_{g2}$	15
S_{N3}	16
T_{p1}	17
m_{G1}	18
m_{G2}	19
n_{p2}	20
γ_{p2}	21
γ_{g2}	22
$(F_t)_{g1}$	23
b_{L2}	24
ψ_2	25
$(F_t)_{p2}$	26

Design 3

Variable	#
HP_{in}	1
n_{p1}	2
m_G	3
n_{g2}	4
T_{p1}	5
m_{G1}	6
m_{G2}	7
n_{p2}	8
h_3	9
w_3	10
$(F_t)_{p1}$	11
V_{p1}	12
d_{p1}	13
N_{p1}	14
P_{d1}	15
N_{g1}	16
n_{g1}	17
r_1	18
V_{g1}	19
HP_{inter}	20
S_{N2}	21
T_{p2}	22
d_{p2}	23
N_{p2}	24
N_{g2}	25
V_{p2}	26

Design 4

Variable	#
S_{N1}	1
$d_{s1,}$Fit	2
T_{p1}	3
n_{p1}	4
m_G	5
n_{g2}	6
w_1	7
h_1	8
d_1	9
HP_{IN}	10
m_{G1}	11
m_{G2}	12
N_{g1}	13
P_{d1}	14
N_{p1}	15
n_{g1}	16
r_{p1}	17
V_{p1}	18
r_{g1}	19
V_{g1}	20
S_{N2}	21
V_{p2}	22
d_{g2}	23
P_{d2}	24
n_{p2}	25
N_{g2}	26

d_g	27
P_d	28
p_x	29
L_w	30
λ_w	31
ψ_g	32
μ	33
Key	34
H_g	35
S_N	36
d_{shaft}	37

$(F_t)_{p1}$	27
V_{p1}	28
d_{p1}	29
N_{p1}	30
P_{d1}	31
N_{g1}	32
n_{g1}	33
r_1	34
V_{g1}	35
HP_{inter}	36
T_{p2}	37
$(\sigma_b)_{nom}$	51
K_t	52
p_f	53
η_{ex}	54
E	55
ν	56
C_p	57
J	58
K_a	59
K_v	60
K_m	61
(σ_b)	62
S_{tbf}	63
$\eta_{bending}$	64
I	65
(σ_{sf})	66
Key -1	67
Bearing -	68
S_f	69

P_{d2}	27
m_o	28
m_i	29
D_o	30
D_i	31
ψ_3	32
d_{s3},Fit	33
d_3	34
S_{N1}	35
d_{s1}, Fit	36
ψ_1	37
b_3	51
$(\sigma_b)_{nom}$	52
Y_3	53
P_{d3}	54
d_{g2}	55
K_t	56
S_{N3}	57
w_1	58
h_1	59
d_1	60
S_f	61
S_u	62
q	63
$(\sigma_b)_{act}$	64
K_f	65
$(\sigma_b)_{eq-CR}$	66
d_2	67
w_2	68
S_u	69

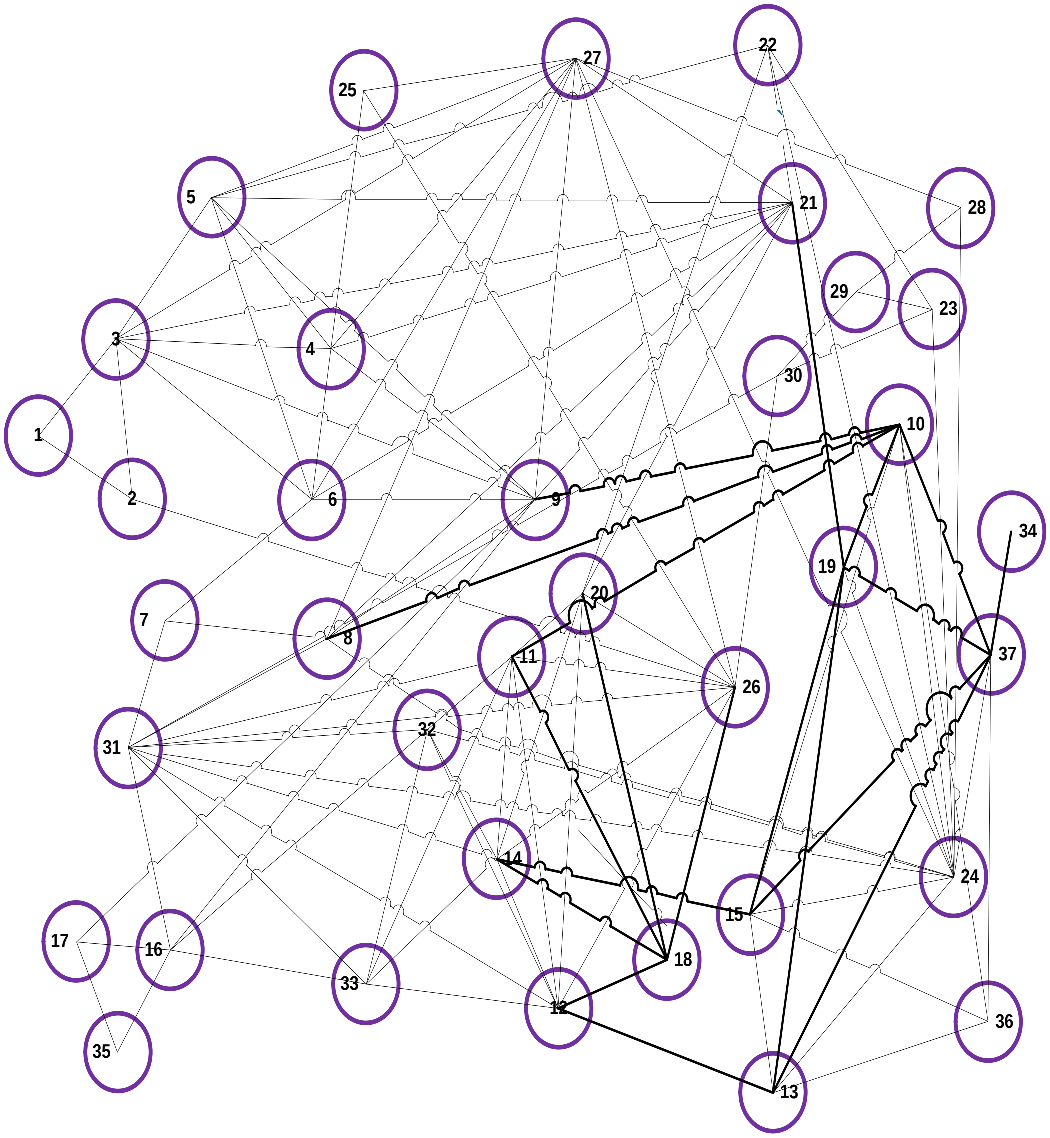
b_{g2}	27
C_{d2}	28
C_{offset}	29
Γ_2	30
P_t	31
P_n	32
C	33
ψ	34
d_{p1}	35
d_{g1}	36
b_1	37
$K_{v(hypoi}$	51
K_o	52
J_p	53
θ_{p2}	54
$\eta_{bending}$	55
S_{tbf}	56
$\sigma_{b(helical)}$	57
K_a	58
$K_{v(helical}$	59
$K_{m(helica}$	60
K_I	61
J	62
m_p	63
C_p	64
σ_{sf}	65
w_2	66
h_2	67
d_2	68
$\eta_{bending}$	69

S_u	70
q	71
$(\sigma_b)_{act}$	72
K_f	73
$(\sigma_b)_{eq\text{-}CR}$	74
S_{sf}	75
$\eta_{surface\text{-}}$	76
η_d	77

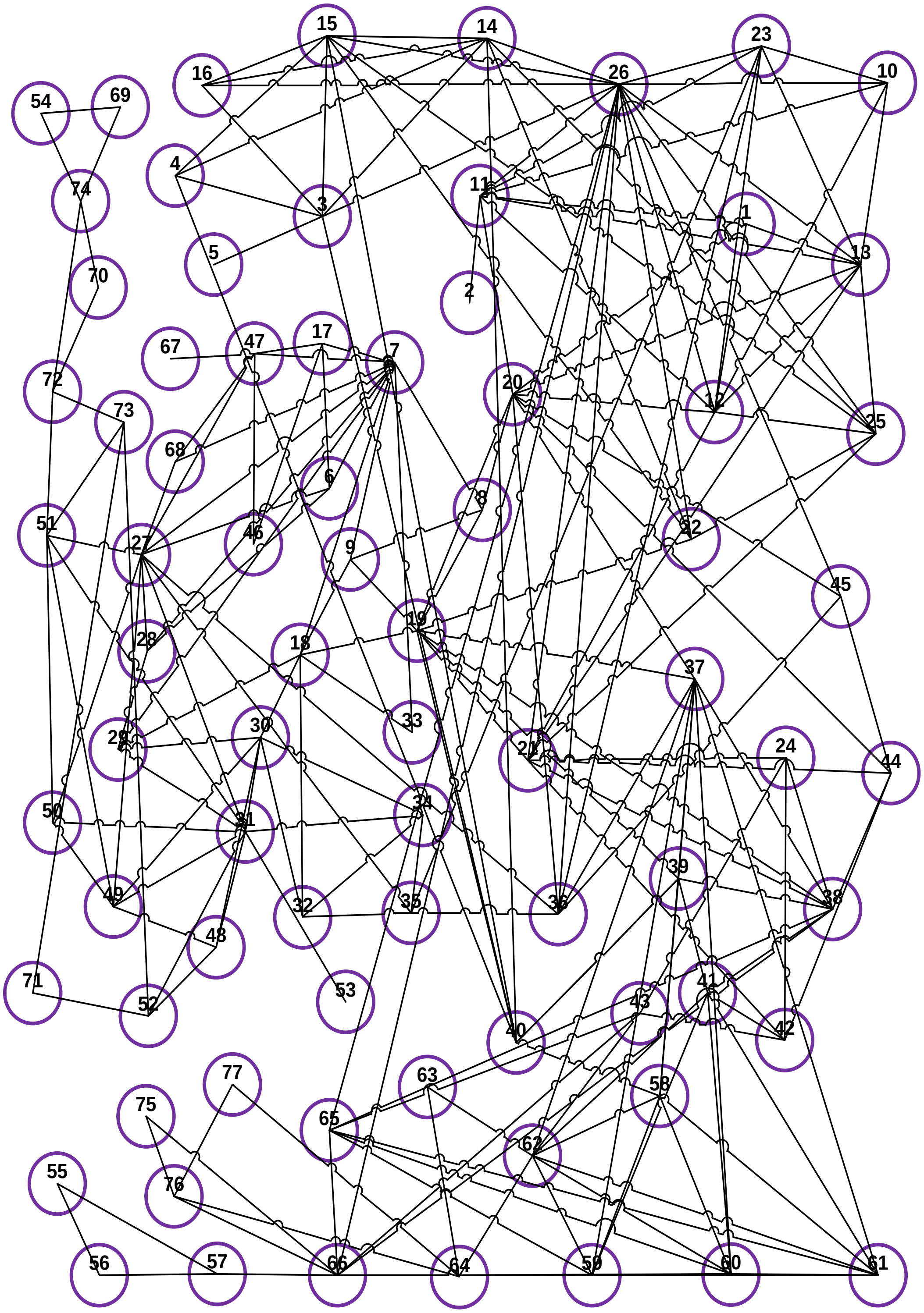
$(\sigma_b)_{act}$	70
K_f	71
q	72
$(\sigma_b)_{eq\text{-}CR}$	73
η_{ex}	74
S_f	75
S_u	76
$(\sigma_b)_{act}$	77
K_f	78
q	79
$(\sigma_b)_{eq\text{-}CR}$	80
η_{ex}	81
S_f	82
Brg -1	83
Brg -2	84
Brg -3	85

S_{tbf}	70
d_{s3}, Fit	71
S_{N3}	72
Bearin	73
Bearin	74
Bearin	75
$\eta_{surface\text{-}}$	76
S_{sf}	77
w_3	78
h_3	79
d_3	80
N_{p2}	81

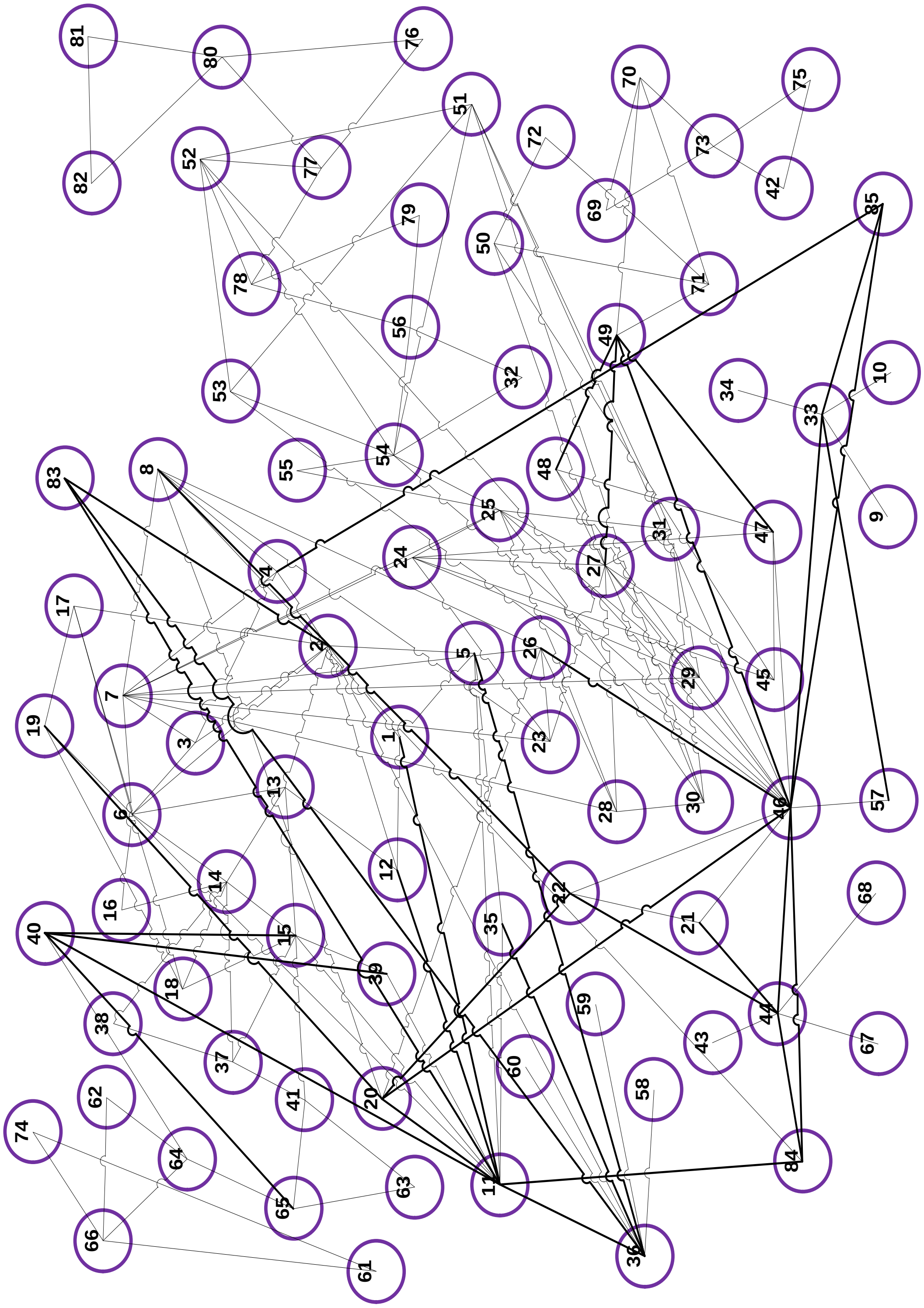
II. Alternative -1 (Ref [13])



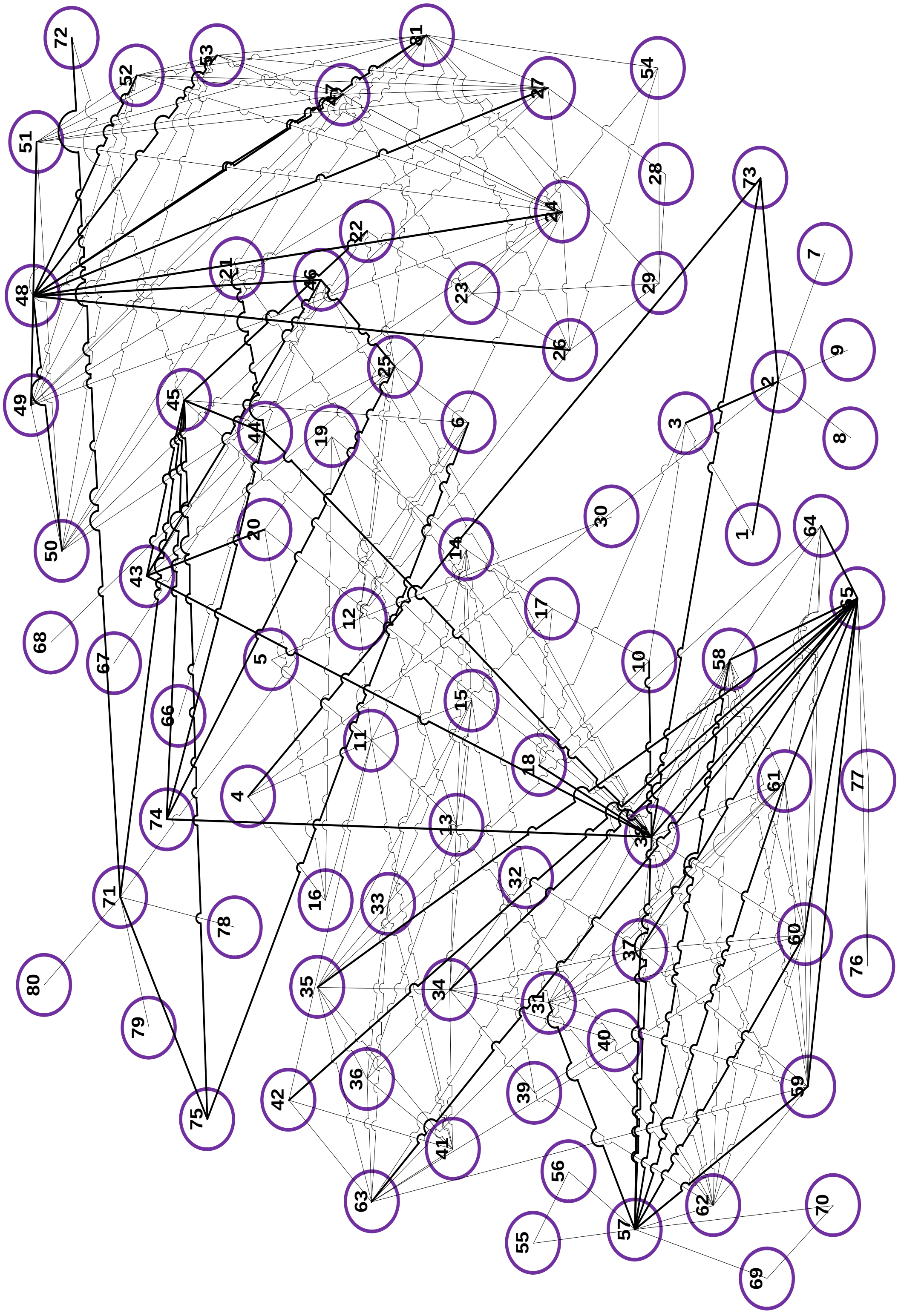
III. Alternative -2 (Ref [13])



IV. Alternative -3 (Ref [13])



V. Alternative -4 (Ref [13])



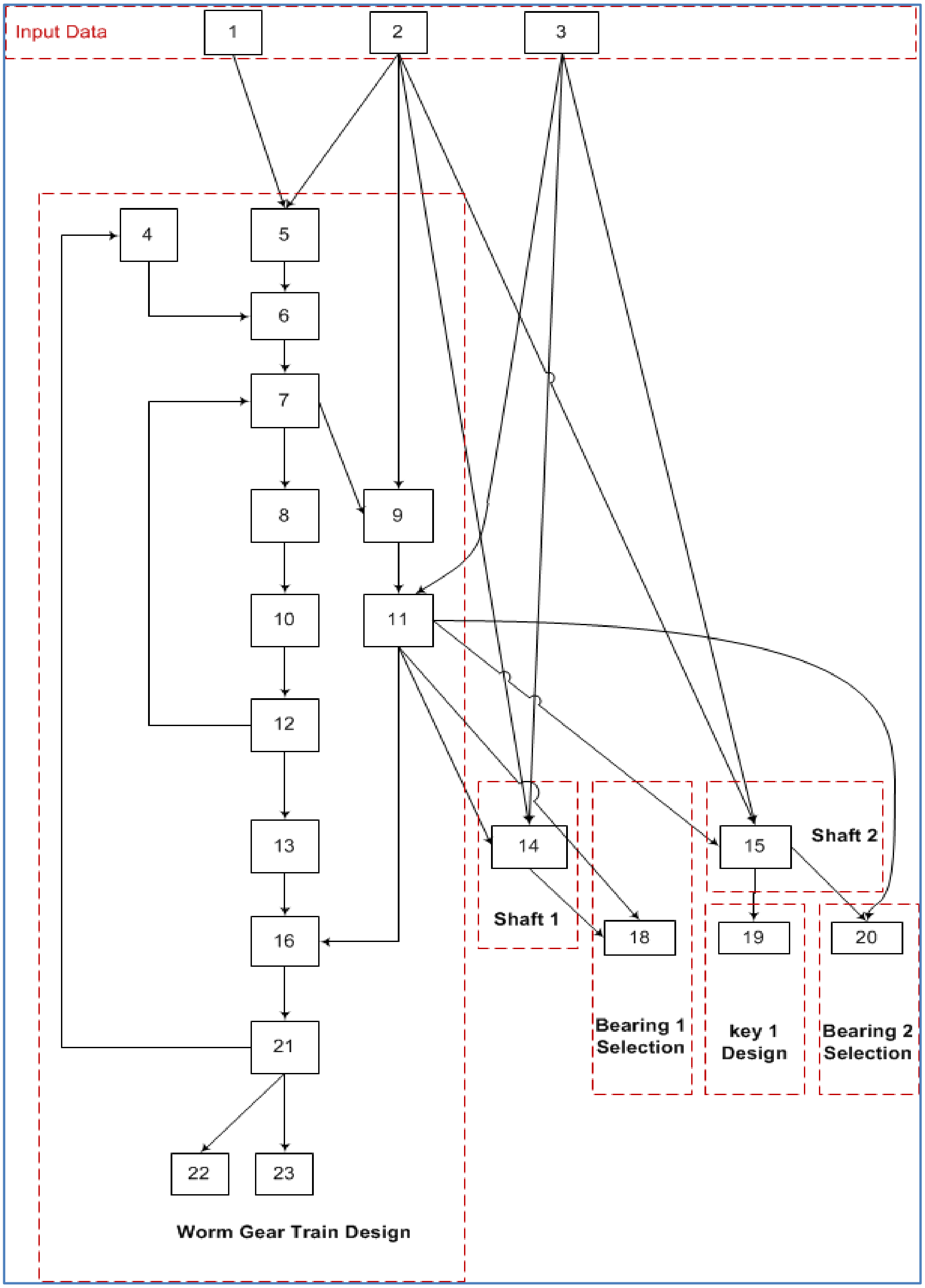
APPENDIX B

DESIGN PROCESS STRUCTURE GRAPHS OF TRANSMISSION ALTERNATIVES (REF [13])

I. Variable list of alternative-1 (Ref [13])

S.no	Variable
1	Input Speed (n_w)
2	Output speed (n_g)
3	Horse power (HP_{in})
4	Select material for Gears
5	Overall Gear ratio (m_G)
6	Select # of worm threads (N_w) (starts)
7	# of worm gear(wheel) teeth (N_g)
8	Pitch diameters of worm and worm gear (d_w, d_g)
9	Pitch line velocity of worm gear train(V_g)
10	diametrical pitches or module of worm gear(P_d)
11	Tangential, Radial and axial force components (F_{gt},F_{wa},F_{wr},F_{wt},F_{ga}, F_{gr})
12	Are Axial and diametrical pitch matching with standard value?
13	Calculate pressure angle (ψ_g), lead angle (λ_w), face width (b)
14	Diameter for Shaft1 (d_{wr} a function of d_w)
15	Diameter for Shaft2 (d_{shaft})
16	Calculate face width (b)
18	Bearing Selection for shaft 1 (Bearing -1)
19	Key Design for shaft 2 (Key)
20	Bearing Selection for shaft 2(Bearing -2)
21	Design safety factor
22	Efficiency (e)
23	Heat generation and Frictional losses (H_g, μ)

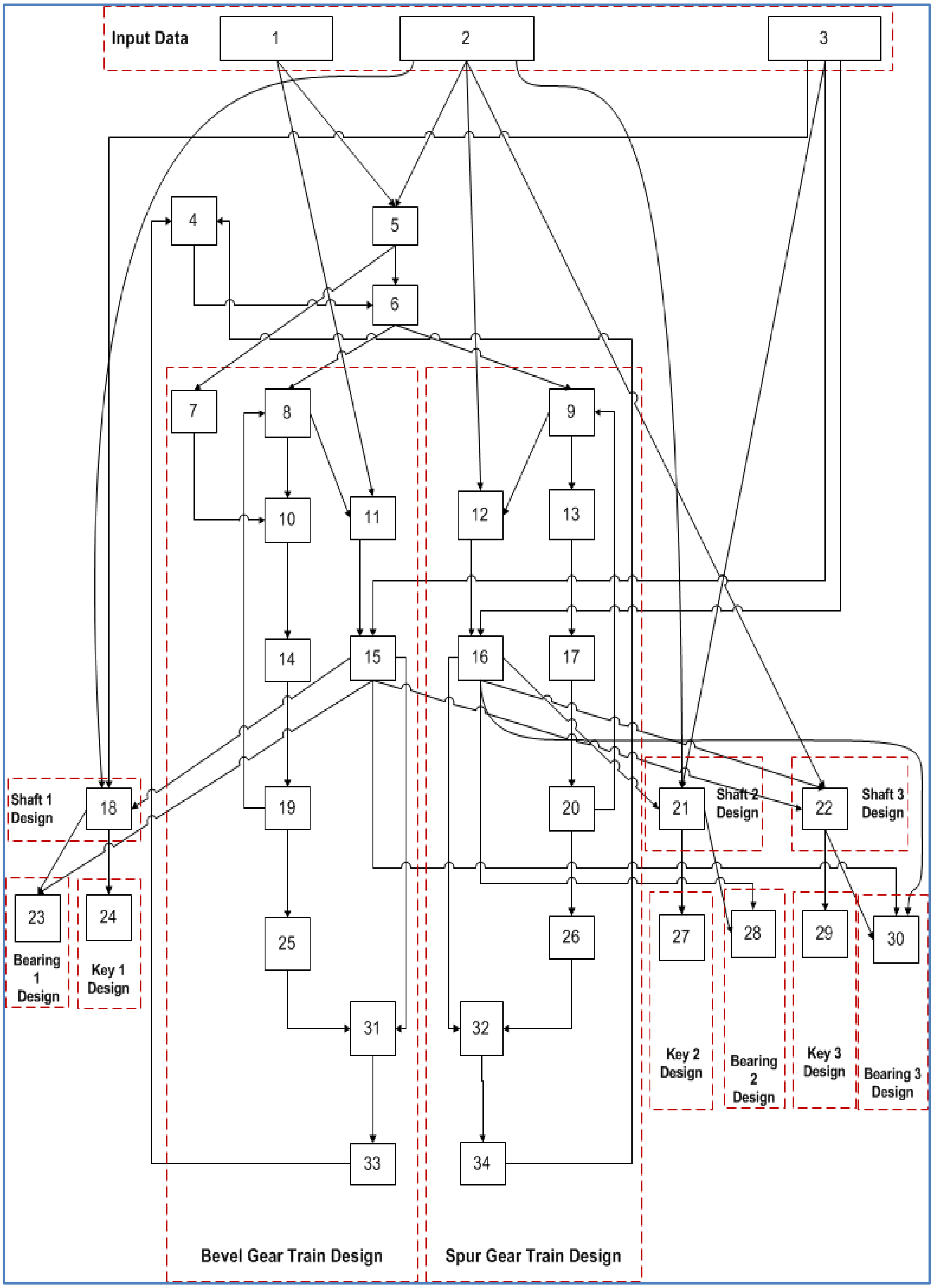
II. Process structure of alternative 1 (Ref [13])



III. Variable list of Alternative-2 (Ref [13])

S. No	Variable
1	Input Speed (n_{p1})
2	Output speed (n_{g2})
3	Horse power (HP_{in})
4	Select material for Gears
5	Overall Gear ratio (m_G)
6	Select gear trains and split the gear ratio (m_{G1}, m_{G2})
7	Pitch cone angle (γ_{p2}, γ_{g2})
8	Pitch diameters of Bevel gear train (d_{p2})
9	Pitch diameters of Spur gears (d_{p1})
10	No of teeth on bevel Gear train ($N_{p2,}$ N_{g2})
11	Pitch line velocity of Spur gear train (V_{p1}, V_{g1})
12	Pitch line velocity of bevel gear train (V_{p2})
13	No of teeth on Spur Gear (N_{p1}, N_{g1})
14	Diametric pitch or module of bevel gear train (P_{d2})
15	Tangential, Radial and axial force for bevel gear ($(F_t)_{p2}$, $(F_r)_{p2}$,$(F_a)_{p2}$,$(F_r)_{g2}$,$(F_a)_{g2}$)
16	Tangential force for spur gear ($(F_t)_{p1}$, $(F_t)_{g1}$)
17	Diametric pitch of Spur gear (P_{d1})
18	Determine diameter of Shaft 1 (d_{s1})
19	Is diametrical pitch matching with standard value?
20	Is diametrical pitch matching with standard value?
21	Determine diameter of Shaft 2 (d_{s2})
22	Determine diameter of Shaft 3 (d_{s3})
23	Bearing Selection for shaft 1 (Bearing -1)
24	Key Design for shaft 1 (Key-1)
25	Calculate all parameters such as pressure angle (ψ_2), face width (b_2), for bevel
26	Calculate all parameters such as pressure angle (ψ_1), face width (b_1), for spur
27	Key Design for Shaft 2 (key-2)
28	Bearing Selection for Shaft 2 (Bearing-2)
29	Key Design for shaft 3 (key-3)
30	Bearing Selection for Shaft 3 (Bearing-3)
31	Tooth bending fatigue Stresses(σ_b) and surface fatigue contact stress (σ_{sf})for
32	Tooth bending fatigue Stresses (σ_b)for spur gear
33	Is the design safety factor achieved for bevel gear (η_d)?
34	Is the design safety factor achieved for spur gear (η_{ex})?

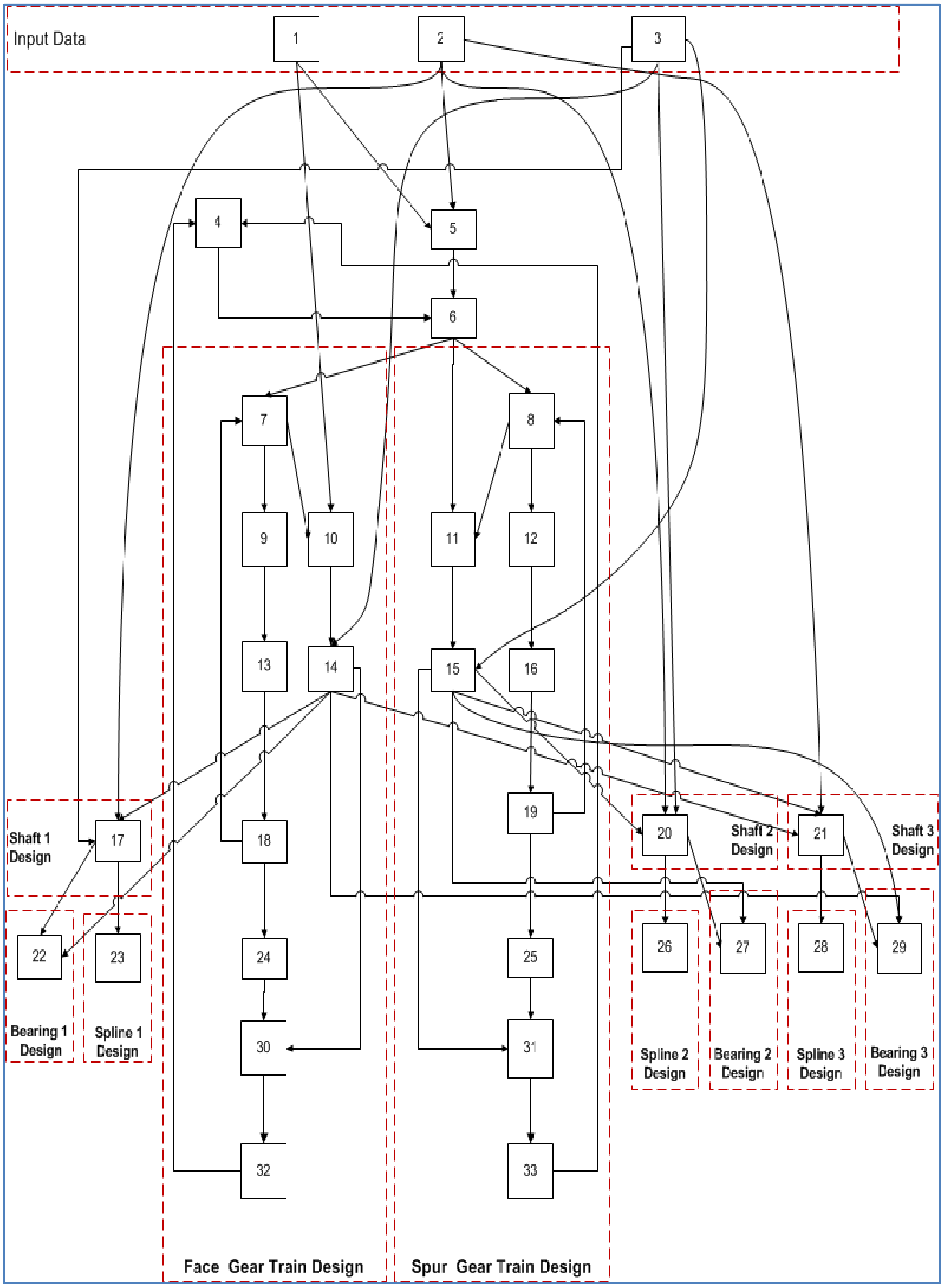
IV. Process structure of Alternative-2 (Ref [13])



V. Variable list of Alternative-3 (Ref [13])

S.no	**Variable**
1	Input Speed (n_{p1})
2	Output speed (n_{g2})
3	Horse power (HP_{in})
4	Select material for Gears
5	Overall Gear ratio (m_G)
6	Select gear trains and split the gear ratio (m_{G1}, m_{G2})
7	Pitch diameters of Face gear train (d_{p2}, d_{g2})
8	Pitch diameters of Spur gear (d_{p1})
9	No of teeth on Face Gear train (N_{g2}, N_{p2})
10	Pitch line velocity of Face gear train (V_{p2})
11	Pitch line velocity of Spur gear train (V_{p1}, V_{g1})
12	No of teeth on Spur Gear (N_{g1}, N_{p1})
13	Diametric pitch or module of Face gear train (P_{d2})
14	Tangential force (((F_t)$_{p1}$)
15	Tangential force ((F_t)$_{p2}$)
16	Diametric pitch or module of Spur gear (P_{d1})
17	Determine diameter of Shaft 1 (d_{s1}, Fit)
18	Is diametrical pitch matching with standard value?
19	Is diametrical pitch matching with standard value?
20	Determine diameter of Shaft 2 (d_{s2}, Fit)
21	Determine diameter of Shaft 3 (d_{s3}, Fit)
22	Bearing Selection for shaft 1 (Bearing -1)
23	Shaft 1 - spline dimensions (w_1, d_1, h_1)
24	Calculate all parameters such as pressure angle (ψ_1), face width (b_1)
25	Calculate all parameters such as pressure angle (ψ_2), face width (b_3 ,b_2)
26	Shaft2 - spline dimensions (w_2, d_2, h_2)
27	Bearing Selection for Shaft 2 (Bearing -2)
28	Shaft3 - spline dimensions (w_3, d_3, h_3)
29	Bearing Selection for Shaft 3 (Bearing -3)
30	Tooth bending fatigue Stresses (σ_b)
31	Tooth bending fatigue Stresses (σ_b)
32	Is the design safety factor achieved for face gear design? (η_{ex})
33	Is the design safety factor achieved for Spur gear design? (η_{ex})

VI. Process structure of Alternative-3 (Ref [13])

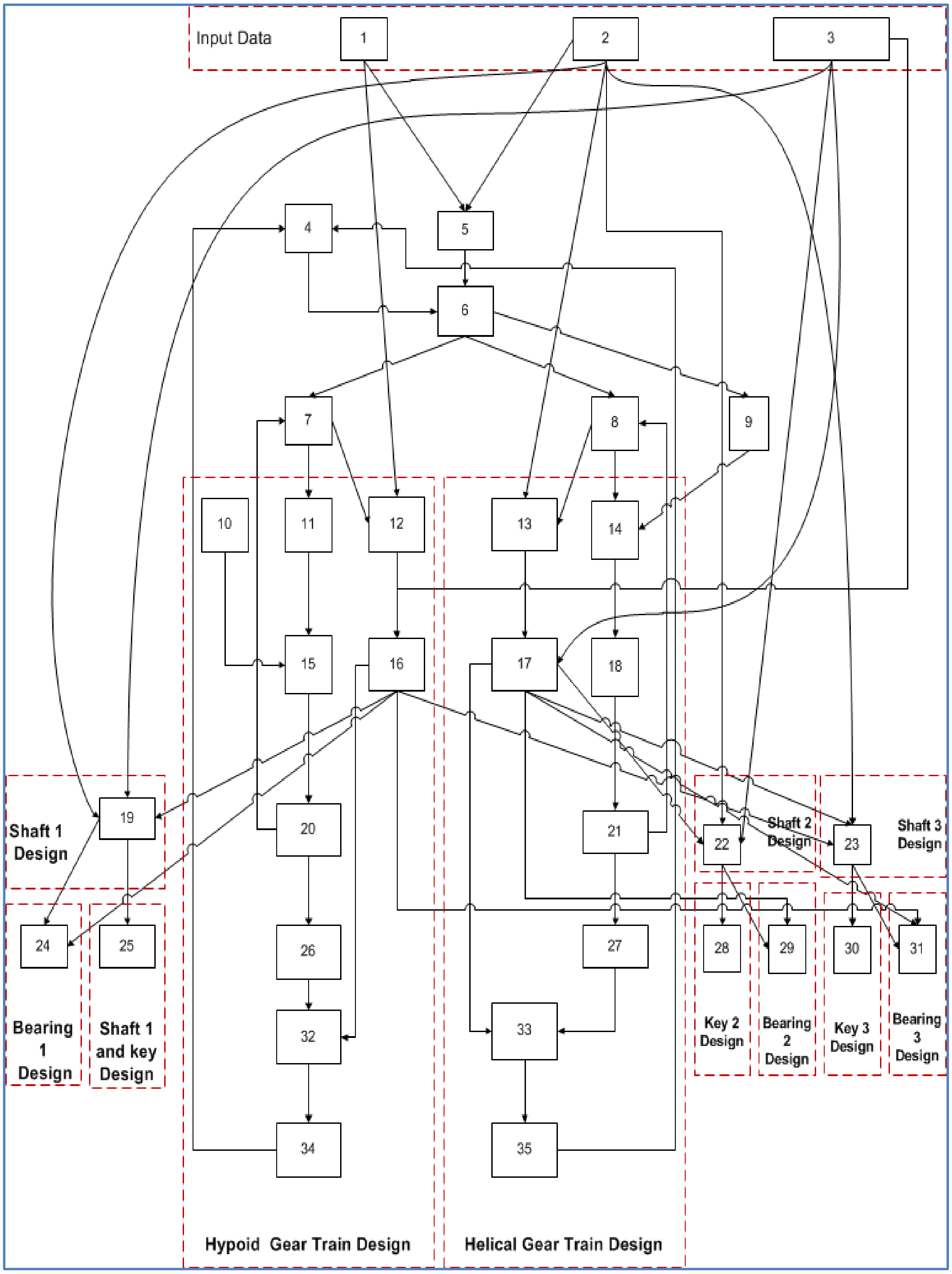


VII. Variable list of Alternative-4 (Ref [13])

S.no	Variable
1	Input Speed (n_{p1})
2	Output speed (n_{g2})
3	Horse power (HP_{in})
4	Select material for Gears
5	Overall Gear ratio (m_G)
6	Select gear trains and split the gear ratio (m_{G1}, m_{G2})
7	Pitch diameters of Hypoid gear train (d_{p2}, d_{g2})
8	Pitch diameters of helical gear (d_{p1}, d_{g1})
9	Select # of start in hypoid Gear (N_{p2})
10	Select helix angle (ψ)
11	No of teeth on Hypoid Gear train (N_{g2})
12	Pitch line velocity of hypoid gear train (V_{p2})
13	Pitch line velocity of helical gear train (V_{p1}, V_{g1})
14	No of teeth on helical Gear (N_{p1}, N_{g1})
15	Diametric pitch or module of Hypoid gear train (P_{d2})
16	Tangential force for Hypoid gear ($(F_t)_{p2}$)
17	Tangential force for helical gear (F_t)
18	Diametric pitch or module of helical gear (P_{d1})
19	Determine diameter of Shaft 1 ($d_{s1,}$Fit)
20	Is diametrical pitch matching with standard value?
21	Is diametrical pitch matching with standard value?
22	Determine diameter of Shaft 2 ($d_{s2,}$Fit)
23	Determine diameter of Shaft 3 ($d_{s3,}$Fit)
24	Bearing Selection for shaft 1 (Bearing -1)
25	Shaft 1 - spline dimensions (w_1, d_1, h_1)
26	Calculate all parameters such as helix angle(φ_t), pressure angle, face
27	Calculate all parameters such as pressure angle, face width (b_{g2})
28	Shaft2 - spline dimensions (w_2, d_2, h_2)
29	Bearing Selection for Shaft 2 (Bearing -2)
30	Shaft3 - spline dimensions (w_3, d_3, h_3)
31	Bearing Selection for Shaft 3 (Bearing -3)

32	Tooth bending fatigue stress $(\sigma_{b(hypoidal)})$ of Hypoid gear
33	Tooth bending fatigue Stress $(\sigma_{b(helical)})$& surface fatigue contact
34	Is the design safety factor achieved for hypoidal gear design? (η_{ex})
35	Is the design safety factor achieved for helical gear design?

VIII. Process structure of Alternative-4 (Ref [13])



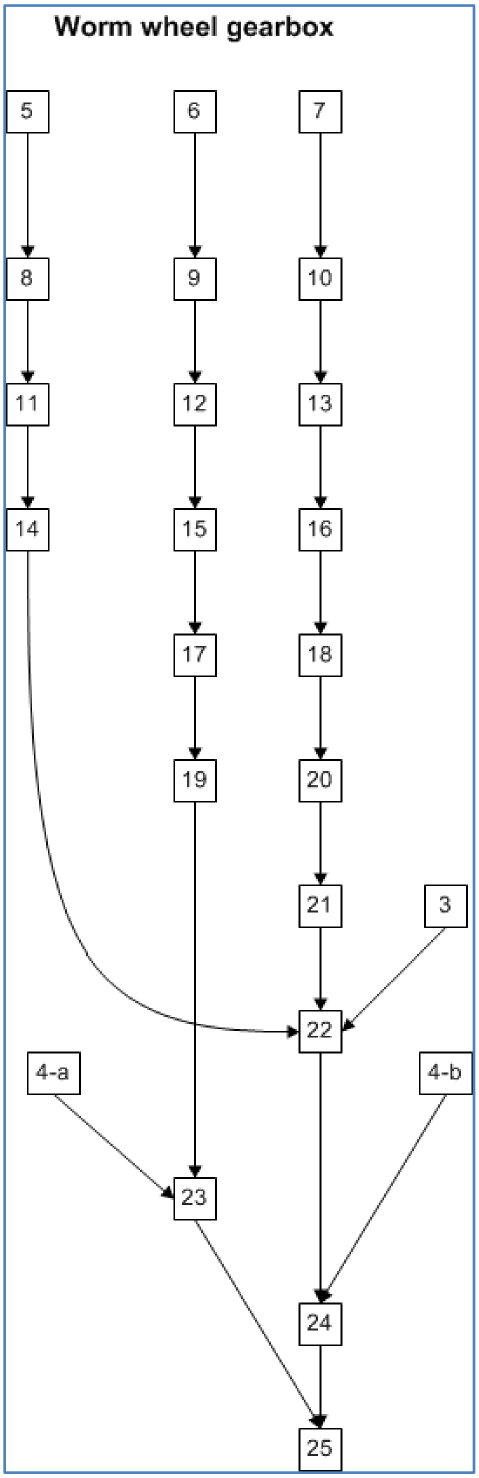
APPENDIX C

MANUFACTURING STRUCTURE GRAPHS OF TRANSMISSION ALTERNATIVES (REF [13])

I. Manufacturing processes list of Alternative-1 (Ref [13])

Number	**Variable**
3	Key-1 for shafts–1
4(a)	Select Bearings-1 for shafts – 1
4(b)	Select Bearings-2 for shafts –2
5	Select metal blank appropriate for Shaft -1: cast or forged
6	Select metal blank appropriate for Worm : cast or forged
7	Select metal blank appropriate for Helical Gear -1: cast or forged
8	Turn: Dimension of Shaft-1
9	Turn to the diameter of worm thread
10	Drill to the dimension of Shaft -2 Hole
11	Hob or Mill keyway for key-1
12	Turn to the diameter of shaft excluding worm area
13	Bore to the final dimension of Shaft-2 Hole
14	Shot Peen shaft-1
15	Gear Hob: Worm -1 Teeth
16	Broach keyway to the dimension of key -1
17	Gear Honing: Worm-1
18	Face Mill or Face Hob: Straight Helical Gear -1
19	Heat treat: Carburizing or Annealing
20	Gear Honing: Helical Gear -1
21	Heat treat: Carburizing or Annealing
22	Assemble Shaft-2, Helical Gear-1 and Key-1
23	Fit Bearing-1 on Worm Shaft
24	Fit Bearing-2 on Shaft-2
25	Mount Bearing -1 and Bearing -2 on the housing to mesh Hypoid Gears – 1,2

II. Manufacturing structure of alternative-1 (Ref [13])



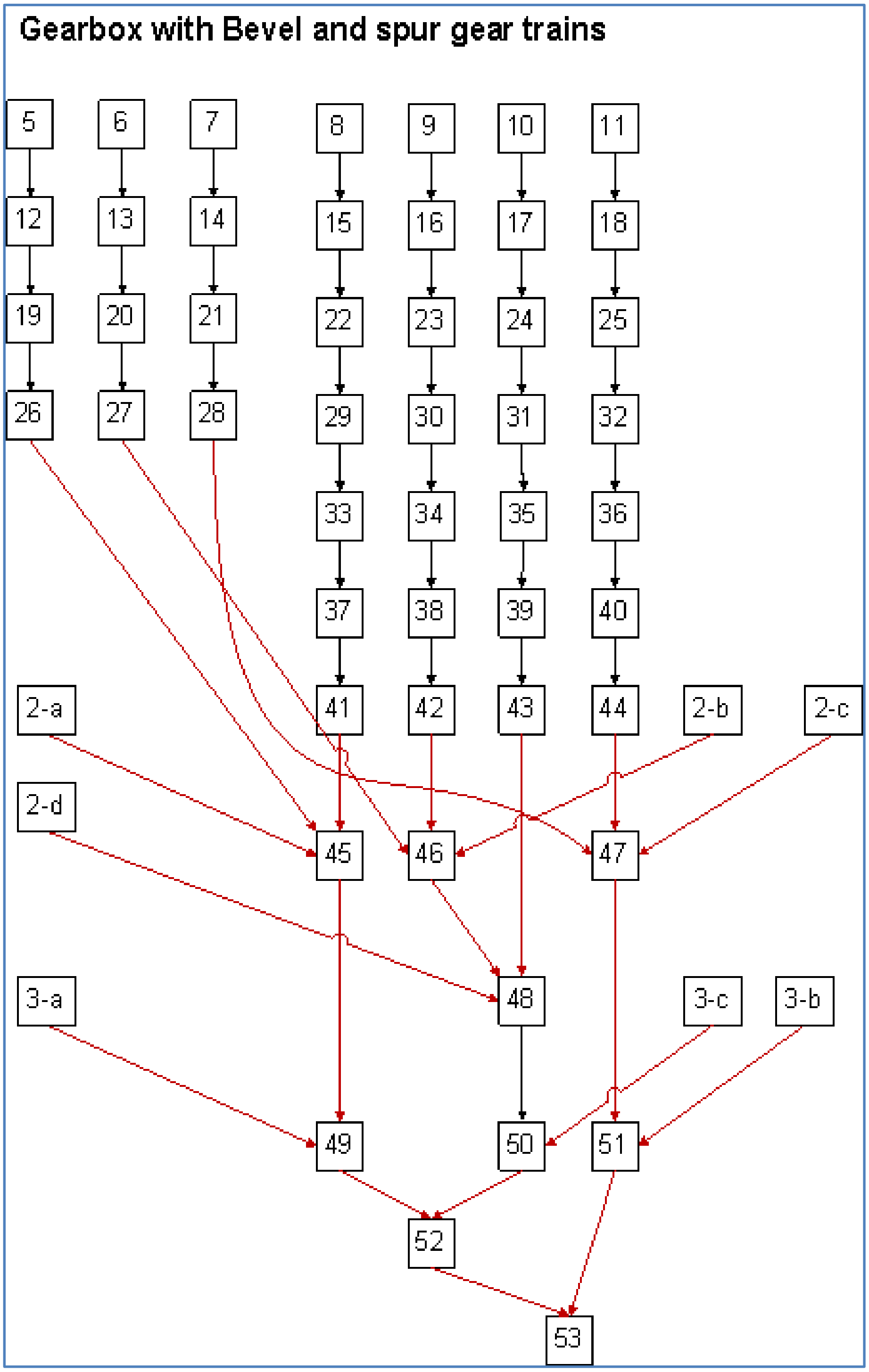
III. Variable list of Alternative-2 (Ref [13])

Number	Variable
2(a)	Select Key-1 for bevel gear– 1
2(b)	Select Key-2 for bevel gear– 2
2(c)	Select Key-3 for spur gear– 1
2(d)	Select Key-4 for bevel gear– 2
3(a)	Select Bearings-1 for shafts – 1
3(b)	Select Bearings-2 for shafts – 2
3(c)	Select Bearings-3 for shafts – 3
5	Select metal blank appropriate for Shaft -1: cast or forged
6	Select metal blank appropriate for Shaft -2: cast or forged
7	Select metal blank appropriate for Shaft -3: cast or forged
8	Select metal blank appropriate for Spur Gear -1: cast or forged
9	Select metal blank appropriate for Spur Gear -2: cast or forged
10	Select metal blank appropriate for Bevel Gear -1: cast or forged
11	Select metal blank appropriate for Bevel Gear -2: cast or forged
12	Turn: Dimension of Shaft-1
13	Turn: Dimension of Shaft-2
14	Turn: Dimension of Shaft-3
15	Drill spur gear-1 to the dimension of Shaft -1 Hole
16	Drill spur gear-2 to the dimension of Shaft -2 Hole
17	Drill bevel gear-1 to the dimension of Shaft -2 Hole
18	Drill bevel gear-2 to the dimension of Shaft -3 Hole
19	Hob or Mill keyway for key-1
20	Hob or Mill keyway for key-2
21	Hob or Mill keyway for key-3
22	Bore spur gear-2 to the final dimension of Shaft-1 Hole

23	Bore spur gear-2 to the final dimension of Shaft-2 Hole
24	Bore bevel gear-1 to the final dimension of Shaft-2 Hole
25	Bore bevel gear-2 to the final dimension of Shaft-3 Hole
26	Shot Peen shaft-1
27	Shot Peen shaft-2
28	Shot Peen shaft-3
29	Broach spur gear-1 keyway to the dimension of key -1
30	Broach spur gear-2 keyway to the dimension of key -2
31	Broach bevel gear-1 keyway to the dimension of key -2
32	Broach bevel gear-2 keyway to the dimension of key -3
33	Gear Hob: Spur Gear -1 Teeth
34	Gear Hob: Spur Gear -2 Teeth
35	Face Mill or Face Hob: Straight Bevel Gear -1
36	Face Mill or Face Hob: Straight Bevel Gear - 2
37	Gear Honing: Spur Gear -1
38	Gear Honing: Spur Gear -2
39	Gear Honing: Bevel Gear -1
40	Gear Honing: Bevel Gear -2
41	Heat treat Spur Gear -1: Carburizing or Annealing
42	Heat treat Spur Gear -2: Carburizing or Annealing
43	Heat treat Bevel Gear -1: Carburizing or Annealing
44	Heat treat Bevel Gear -2: Carburizing or Annealing
45	Assemble Shaft-1, Spur Gear-1 and Key-1
46	Assemble Shaft-2, Spur Gear-2 and Key-2
47	Assemble Shaft-3, Bevel Gear-2 and Key-3
48	Assemble Shaft-2, Bevel Gear-1 and Key-2
49	Fit Bearing-1 on Shaft-1
50	Fit Bearing-2 on Shaft-2

51	Fit Bearing-3 on Shaft-3
52	Mount Bearing -1 and Bearing -2 on the housing to mesh Spur Gears – 1,2
53	Mount Bearing -3 on the housing to mesh Bevel Gears – 1,2

IV. Manufacturing structure of alternative-2 (Ref [13])

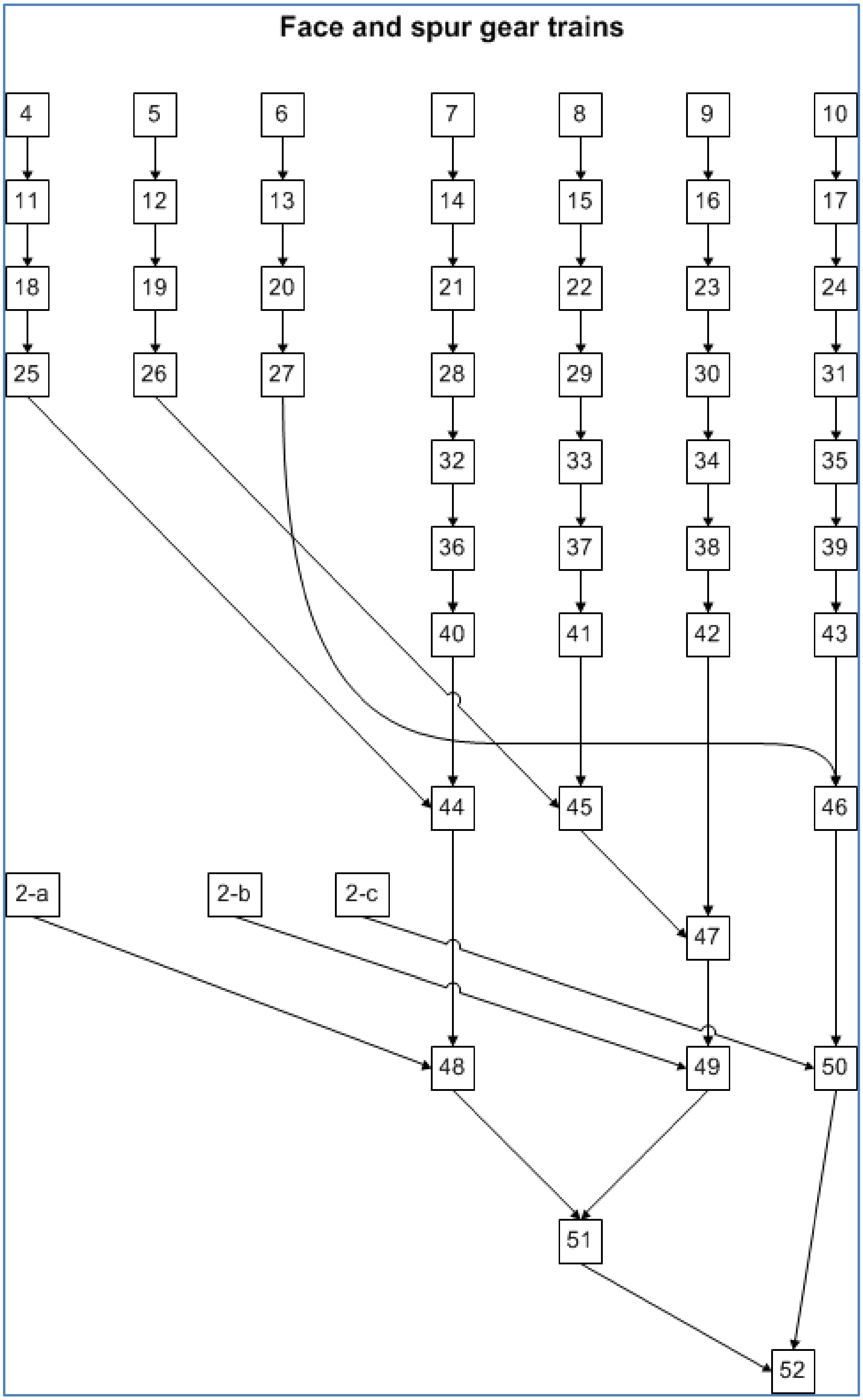


V. Variable list of Alternative-3 (Ref [13])

Number	Variable
2(a)	Select Bearings-1 for shafts – 1
2(b)	Select Bearings-2 for shafts – 2
2(c)	Select Bearings-3 for shafts – 3
4	Select metal blank appropriate for Shaft -1: cast or forged
5	Select metal blank appropriate for Shaft -2: cast or forged
6	Select metal blank appropriate for Shaft -3: cast or forged
7	Select metal blank appropriate for Spur Gear -1: cast or forged
8	Select metal blank appropriate for Spur Gear -2: cast or forged
9	Select metal blank appropriate for Spur Gear -3: cast or forged
10	Select metal blank appropriate for Face Gear: cast or forged
11	Turn: Dimension of Shaft-1
12	Turn: Dimension of Shaft-2
13	Turn: Dimension of Shaft-3
14	Drill to the dimension of Shaft -1 Hole
15	Drill to the dimension of Shaft -2 Hole
16	Drill to the dimension of Shaft -2 Hole
17	Drill to the dimension of Shaft -3 Hole
18	Gear Hob External Splines for shaft-1
19	Gear Hob External Splines for shaft-2
20	Gear Hob External Splines for shaft-3
21	Bore to the final dimension of Shaft-1 Hole
22	Bore to the final dimension of Shaft-2 Hole
23	Bore to the final dimension of Shaft-2 Hole
24	Bore to the final dimension of Shaft-3 Hole
25	Shot Peen shaft-1

26	Shot Peen shaft-2
27	Shot Peen shaft-3
28	Broach Mating Splines to the dimension of spline on shaft -1
29	Broach Mating Splines to the dimension of spline on shaft -2
30	Broach Mating Splines to the dimension of spline on shaft -2
31	Broach Mating Splines to the dimension of spline on shaft -2
32	Gear Hob: Spur Gear -1 Teeth
33	Gear Hob: Spur Gear -2 Teeth
34	Gear Hob: Spur Gear -3 Teeth
35	Gear Shaping: Use pinion cutters to shape face gear teeth
36	Gear Honing: Spur Gear -1
37	Gear Honing: Spur Gear -2
38	Gear Honing: Spur Gear -3
39	Gear Honing: Face Gear
40	Heat treat: Carburizing or Annealing
41	Heat treat: Carburizing or Annealing
42	Heat treat: Carburizing or Annealing
43	Heat treat: Carburizing or Annealing
44	Assemble Shaft-1, Spur Gear-1
45	Assemble Shaft-2, Spur Gear-2
46	Assemble Shaft-3, Face Gear
47	Assemble Shaft-2, Spur Gear-3
48	Fit Bearing-1 on Shaft-1
49	Fit Bearing-2 on Shaft-2
50	Fit Bearing-3 on Shaft-3
51	Mount Bearing -1 and Bearing -2 on housing to mesh Spur Gears – 1,2
52	Mount Bearing -3 on the housing to mesh Face Gear and Spur Gear -3

VI. Manufacturing structure of alternative-3 (Ref [13])

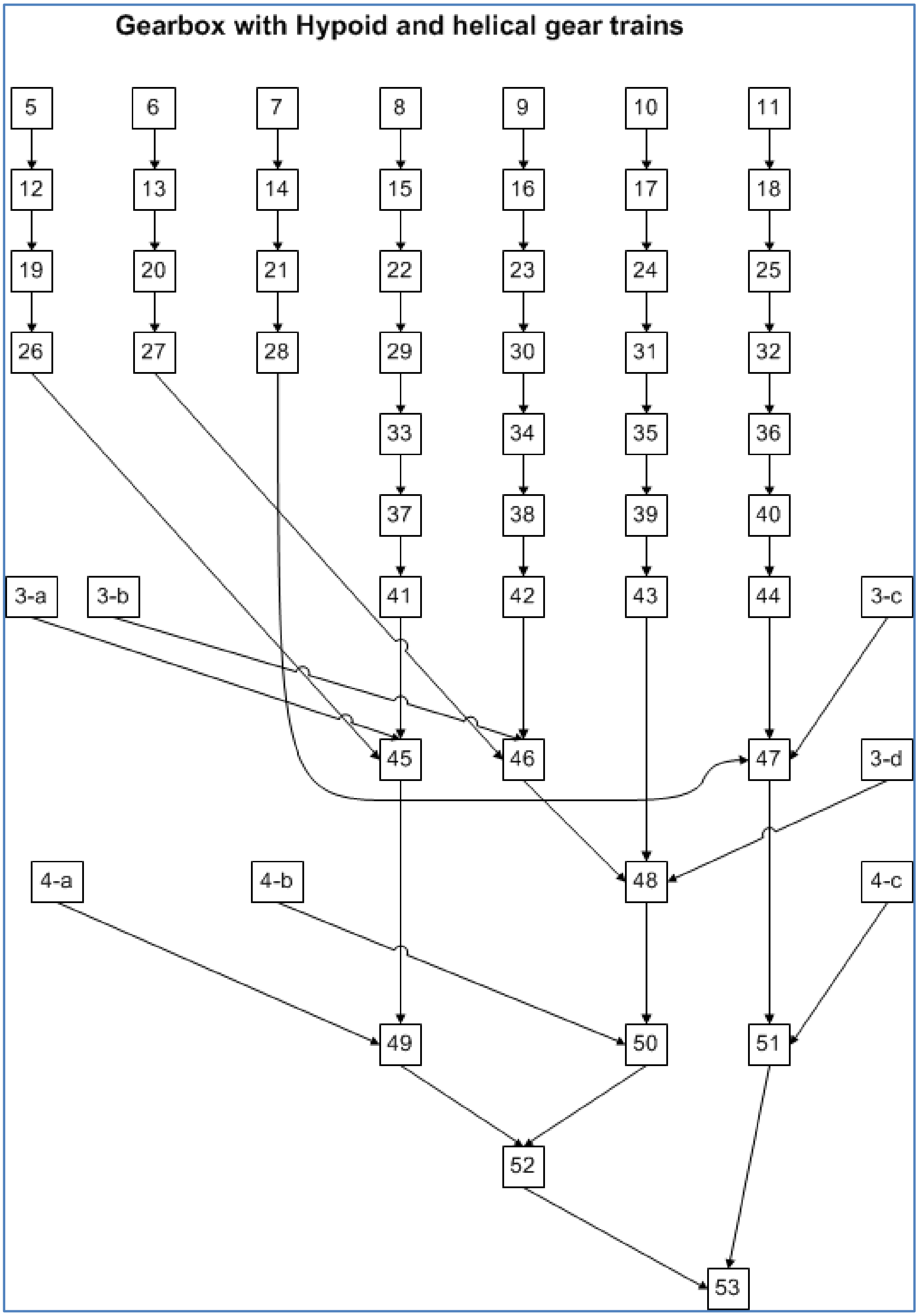


VII. Variable list of Alternative-4 (Ref [13])

Number	**Variable**
3(a)	Select Key-1 for Hypoid gear-1
3(b)	Select Key-2 for Hypoid gear-2
3(c)	Select Key-3 for Helical gear-1
3(d)	Select Key-4 for Helical gear- 2
4(a)	Select Bearings-1 for shafts – 1
4(b)	Select Bearings-2 for shafts – 2
4(c)	Select Bearings-3 for shafts – 3
5	Select metal blank appropriate for Shaft -1: cast or forged
6	Select metal blank appropriate for Shaft -2: cast or forged
7	Select metal blank appropriate for Shaft -3: cast or forged
8	Select metal blank appropriate for Hypoid Gear -1: cast or forged
9	Select metal blank appropriate for Hypoid Gear -2: cast or forged
10	Select metal blank appropriate for Helical Gear -1: cast or forged
11	Select metal blank appropriate for Helical Gear -2: cast or forged
12	Turn: Dimension of Shaft-1
13	Turn: Dimension of Shaft-2
14	Turn: Dimension of Shaft-3
15	Drill to the dimension of Shaft -1 Hole
16	Drill to the dimension of Shaft -2 Hole
17	Drill to the dimension of Shaft -2 Hole
18	Drill to the dimension of Shaft -3 Hole
19	Hob or Mill keyway for key-1
20	Hob or Mill keyway for key-2
21	Hob or Mill keyway for key-3
22	Bore to the final dimension of Shaft-1 Hole
23	Bore to the final dimension of Shaft-2 Hole
24	Bore to the final dimension of Shaft-2 Hole

25	Bore to the final dimension of Shaft-3 Hole
26	Shot Peen shaft-1
27	Shot Peen shaft-2
28	Shot Peen shaft-3
29	Broach keyway to the dimension of key -1
30	Broach keyway to the dimension of key -2
31	Broach keyway to the dimension of key -2
32	Broach keyway to the dimension of key -3
33	Gear Hobbing: Hypoid Gear -1 Teeth
34	Gear Hob: Hypoid Gear -2 Teeth
35	Face Mill or Face Hob: Straight Helical Gear-1
36	Face Mill or Face Hob: Straight Helical Gear- 2
37	Gear Honing: Hypoid Gear -1
38	Gear Honing: Hypoid Gear -2
39	Gear Honing: Helical Gear -1
40	Gear Honing: Helical Gear -2
41	Heat treat: Carburizing or Annealing
42	Heat treat: Carburizing or Annealing
43	Heat treat: Carburizing or Annealing
44	Heat treat: Carburizing or Annealing
45	Assemble Shaft-1, Hypoid Gear-1 and Key-1
46	Assemble Shaft-2, Hypoid Gear-2 and Key-2
47	Assemble Shaft-3, Helical Gear-2 and Key-3
48	Assemble Shaft-2, Helical Gear-1 and Key-2
49	Fit Bearing-1 on Shaft-1
50	Fit Bearing-2 on Shaft-2
51	Fit Bearing-3 on Shaft-3
52	Mount Bearing -1 and Bearing -2 on the housing to mesh Hypoid Gears – 1,2
53	Mount Bearing -3 on the housing to mesh Helical Gears – 1,2

VIII. Manufacturing structure of alternative-4 (Ref [13])



APPENDIX D

MATLAB PROGRAM FOR UNWEIGHTED COMPLEXITY METRICS

Matlab code have the following 5 functions:

1. adj_matrix_gui:

```
clear all;
close all;
clc;
adj_matrix_gui;

function [file_name]=adj_matrix_gui(action)
 % If there is no map, just initiate the plot
if nargin == 0
    action = 'init';
end

switch action
case 'motion'
    line_h = getappdata(gcf,'motionline');
    pt = get(gca,'CurrentPoint');
    pt = pt(1,:);
    xdata = get(line_h,'XData');
    ydata = get(line_h,'YData');
    xdata(2) = pt(1);
    ydata(2) = pt(2);
    set(line_h,'XData',xdata,'YData',ydata)
case 'down'
    button = get(gcf,'SelectionType');
    switch button
    case 'normal'
        %get the information of the clicked point and pass it
from gco to h
        h = gco;
        fig = gcf;
        % First click
        if ~isappdata(fig, 'motionline')
            %find the point by the text, h is the position of the
first point
            if isequal(get(h,'Type'),'text')
                pt = get(h,'Position');
                hold on
                line_h = plot(pt(1), pt(2),'b-' ...
                                          ,'EraseMode','normal');
                setappdata(line_h,'startobj',h)    % Save start
object in the line_h from h
                hold off
                stack_text_on_top
                setappdata(fig,'motionline',line_h)
                set(fig,'WindowButtonMotionFcn',
'adj_matrix_gui(''motion'')');
```

```
            end
        else
        % Second click
            line_h = getappdata(fig,'motionline');find the point
by the text
            if isequal(get(gco,'Type'),'text')
                startobj = getappdata(line_h,'startobj');
                endobj = gco;% Save start object in the line_h
from h
                startpt = get(startobj,'Position');
                endpt = get(endobj,'Position');
                set(line_h,'XData',[startpt(1) endpt(1)] ...
                          ,'YData',[startpt(2) endpt(2)]);
                I = round(str2double(get(startobj,'String')));
                J = round(str2double(get(endobj,'String')));
                Matrix = getappdata(gcf,'Matrix');
                Matrix(I,J) = Matrix(I,J)+1;%add a relation in
the matrix
                Matrix(J,I) = Matrix(J,I)+1;
                setappdata(gcf,'Matrix',Matrix)
                Matrix
            else
                delete(line_h)% If doesn't get a text--point,
stop plotting line
            end
            rmappdata(gcf,'motionline')
            set(fig,'WindowButtonMotionFcn', '');
            if isappdata(gcf,'message')
            delete(getappdata(gcf,'message'));
             end
        fig=gcf;
            [result,message] =
calc_metrics(getappdata(gcf,'Matrix'));
            setappdata(fig,'message',message)
            plot_charts(result,4) ;
        end

    case 'open'
        h = gco;
         if ~isequal(get(h,'Type'),'text')%If it does exist a
point at the
             %current click, start a new one
                pt = get(gca,'CurrentPoint');
                pt = pt(1,:);
                hold on
                n =
1+length(findobj(get(gca,'Children'),'Type','text'));
                h = text(pt(1),pt(2),num2str(n) ...

,'Color','r','FontWeight','bold','FontSize',16);
                        %define the size of the text and the
color
                hold off
                if ~isappdata(gcf,'Matrix')
```

```
                    setappdata(gcf,'Matrix',[])
                end
                if ~isappdata(gcf,'mypoints')
                    setappdata(gcf,'mypoints',[])
                end
                mypoints = getappdata(gcf,'mypoints');
                mypoints(n,:)=pt;
                setappdata(gcf,'mypoints',mypoints)
                mypoints
                Matrix = getappdata(gcf,'Matrix');
                Matrix(n,n) = 0;
                setappdata(gcf,'Matrix',Matrix)
                Matrix
                if isappdata(gcf,'message')
                  try  delete(getappdata(gcf,'message'));
                  end
                end
                fig=gcf;
                [result,message] =
calc_metrics(getappdata(gcf,'Matrix'));
                setappdata(fig,'message',message)
                plot_charts(result,4) ;
         end

    case 'alt'
     %Use the right click to delet components.
     fig=gcf;
     switch get(gco,'Type')
        case 'text'
            the_component_to_be_delete=gco;
            %If the mouse click on the text then delete the vetex
            n = round(str2double(get(gco,'String')));
            pt = get(gco,'Position');
            %at this time the handles include the text messages,
the

            %needed at that line.
            handles = get(gca,'Children');
            for I=1:length(handles)
                h = handles(I);
                if isequal(get(h,'Type'),'text')%Renumber the
text
                    n2 = round(str2double(get(h,'String')));
                    if n2 > n
                        set(h,'String',n2-1)
                    end
                else
                    xdata = get(h,'XData');
                    ydata = get(h,'YData');
                    if length(xdata)==2 %* to find the line that
link to the chosen point
                    if (xdata(1) == pt(1) & ydata(1) == pt(2))
...
                     |  (xdata(2) == pt(1) & ydata(2) == pt(2))
```

```
                delete(h)% to delete the line that link to the
chosen point
                    end
                    end
                end
            end
            if isappdata(gcf,'Matrix')
                Matrix = getappdata(gcf,'Matrix');
                Matrix(n,:) = [];
                Matrix(:,n) = [];
                setappdata(gcf,'Matrix',Matrix)
                 Matrix

            end
            delete(the_component_to_be_delete)%Delete the point
in the form of 'text'
            if isappdata(gcf,'message')
            delete(getappdata(gcf,'message'));
            end
        fig=gcf;
            [result,message] =
calc_metrics(getappdata(gcf,'Matrix'));
            setappdata(fig,'message',message)
        case 'line'
            the_component_to_be_delete=gco;
             %If the mouse click on the line then delete the line
            xdata = get(gco,'XData');
            ydata = get(gco,'YData');
            txt_h = findobj(get(gca,'Children'),'Type','text');
            for K=1:length(txt_h)
                h = txt_h(K);
                pt = get(h,'Position');
                if (xdata(1) == pt(1) & ydata(1) == pt(2))
                    I = round(str2double(get(h,'String')));
                elseif (xdata(2) == pt(1) & ydata(2) == pt(2))
                    J = round(str2double(get(h,'String')));
                end
            end
            if isappdata(gcf,'Matrix')
                Matrix = getappdata(gcf,'Matrix');
                Matrix(I,J) = Matrix(I,J)-1;
                Matrix(J,I) = Matrix(J,I)-1;

                setappdata(gcf,'Matrix',Matrix)
                my = getappdata(gcf,'mypoints');
                Ma = getappdata(gcf,'Matrix');
                Matrix
            end
            delete(the_component_to_be_delete)%Delete the line
            if isappdata(gcf,'message')
            delete(getappdata(gcf,'message'));
            end
```

```
                         fig=gcf;
            [result,message] =
calc_metrics(getappdata(gcf,'Matrix'));
            setappdata(fig,'message',message)
         otherwise return
        end % End object switch
         plot_charts(result,4);%Finish deleting the components.



    end % End button switch
case 'keypress'
    ESC = 27;%press ESC to stop drawing the line
    SPACE = 32;%press space to out put the vertex and matrix to
the txt file
    switch get(gcf,'CurrentCharacter')
    case ESC
        if isappdata(gcf,'motionline')
            line_h = getappdata(gcf,'motionline');
            delete(line_h)
            rmappdata(gcf,'motionline')
        end

     case SPACE
          my = getappdata(gcf,'mypoints'); % get the points data
from the handle gcf
          Ma = getappdata(gcf,'Matrix');   % get the matrix data
from the handle gcf
          new=inputdlg('Enter the  file name to restore the
output data :',...
                        'Require a name', 1,{'outputdata.txt'});
          options.Resize='on';
          options.WindowStyle='normal';
          options.Interpreter='tex';
            if strcmp(new,'')==1 %if there is no input
                errordlg('Invalid input','File Error')
                return
            elseif length(new)==0
                return
            else file_name=new{1};

          OutPutVertex_Matrix_ToFile( file_name,my,Ma); %Output
the vertex and matrix to the txt file
          end
     otherwise set(gcf,'WindowButtonMotionFcn', '');
    end

 %Initiate the plot
case 'init'
    scrsz = get(0,'ScreenSize');
    fig = figure('BackingStore', 'on', 'IntegerHandle', 'off',
'Name'...
                , 'Adjacency Matrix', 'NumberTitle', 'off',
'MenuBar',...
```

```
                'none', 'DoubleBuffer','on','Position'...
                ,[10 50 (scrsz(3)-400) (scrsz(4)-100)]);
        Operations=uimenu('Parent',fig,'Label','Operations') ;
        uimenu(Operations,'Label','Open
File','Callback',@refresh);
        uimenu(Operations,'Label','Save','Callback',...
            @save_current_plot);
%         uimenu(f,'Label','Save','Callback','save');

uimenu(Operations,'Label','Quit','Callback',@my_closefcn,...
           'Separator','on','Accelerator','Q');
        pictures=uimenu('Parent',fig,'Label','Pictures') ;
        uimenu(pictures,'Label','Original
Structure','Callback'...
            , @plot_the_image);
        uimenu(pictures,'Label','New Structure','Callback',...
            @plot_the_image2);% set the menu for plotting the
physical system
        HelpF=uimenu('Parent',fig,'Label','Help') ;
        uimenu(HelpF,'Label','Help file','Callback',@helpF);
%   ax  = axes;
%     axis off;
    title('Double click to create Node. Single click to create
link. Right click to delete. Space to save the file.')
    xlim([-1.1 1.1]);
    ylim([-1.1 1.1]);
    selection = questdlg('Use an old file or start a new one
',...
      'New file or not',...
      'New File','Old File','New File');
   switch selection,
      case 'New File',

      case 'Old File'
        %importing file
        status=0;
        new =inputdlg('Enter the  file name :',...
    'File name', 1,{'outputdata.txt'});
    if length(new)==0

        status=1;
    elseif (strcmp(new,'')==1) | (~(length(new{1})>4 &&
strcmp(new{1}(end-3:end),'.txt'))) %if there is no input
        errordlg('Invalid input','File Error')
        status=1;
    else inputfile_name=new{1};
    end
      if status==1
      else
          if ~isappdata(gcf,'Matrix')
            setappdata(gcf,'Matrix',[])
        end
        if ~isappdata(gcf,'mypoints')
            setappdata(gcf,'mypoints',[])
```

```
        end
        [Matrix,h,j]=openData(fig,inputfile_name);
        if j==-1
        else
        fig=h;
%         setappdata(fig,'mypoints',oldPoints)
        setappdata(fig,'Matrix',Matrix) ;
      end
      end
       otherwise return
   end
    set(fig,'CloseRequestFcn',@my_closefcn)
    set(fig,'WindowButtonDownFcn', 'adj_matrix_gui(''down'')');
    set(fig,'KeyPressFcn','adj_matrix_gui(''keypress'')')
    figure(fig)

otherwise
    error(['Unknown - ' action])

end % End action switch

function stack_text_on_top
    ax = gca;
    handles = get(gca,'Children');
    txt_h = findobj(handles,'Type','text');
    set(gca,'Children',[txt_h; setdiff(handles,txt_h)])
function my_closefcn(src,evnt)
% User-defined close request function
% to display a question dialog box
   selection = questdlg('Closing This Figure? The Data will be
Saved after closed ',...
      'Close Request Function',...
      'Save and Close','Close','Cancel','Save and Close');
   switch selection,
      case 'Save and Close',
          if 1
              my = getappdata(gcf,'mypoints'); % get the points
data from the handle gcf
              Ma = getappdata(gcf,'Matrix');   % get the matrix
data from the handle gcf
              new =inputdlg('Enter the  file name to restore the
output data :',...
                            'Require a name',
1,{'outputdata.txt'});% Give name of the output data
              if strcmp(new,'')==1 %if there is no input
                 errordlg('Invalid input','File Error')
                    return
              else file_name=new{1};% Store name of the output
data
              end

              delete(gcf)
              close all  %Close the current windows
```

```
               OutPutVertex_Matrix_ToFile( file_name,my,Ma)
%Generate the output data
          end
      case 'Close'
          delete(gcf)
          close all %Close the current windows
      case 'Cancel'
           return   %Do not close windows and continue plot
   end
function plot_the_image(src,evnt) %plot the original physical
relation
        status=0;
        new =inputdlg('Enter the  file name that restoring the
data :',...
                        'Require a name',
1,{'CircuitDiagram.jpg'});
    if strcmp(new,'')==1 %if there is no input
        errordlg('Invalid input','File Error')
        status=1;
    elseif length(new)==0
        status=1;
    else inputfile_name=new{1};
    end
    if status==1
    else    imdata=imread (inputfile_name);
            fig2=figure('BackingStore', 'on', 'IntegerHandle'...
                , 'off', 'Name', 'Original plot' ...
            ,'NumberTitle', 'off', 'MenuBar', 'none',
'DoubleBuffer','on'...
            ,'Position',[500 300 500 400]);
            image (imdata);
            axis off;
      end
function plot_the_image2(src,evnt) %plot the new physical
relation
        status=0;
        new =inputdlg('Enter the  file name that restoring the
data :',...
                        'Require a name',
1,{'ModifiedCircuit.jpg'});
    if strcmp(new,'')==1 %if there is no input
        errordlg('Invalid input','File Error')
        status=1;
    elseif length(new)==0
        status=1;
    else inputfile_name=new{1};
    end
    if status==1
    else    imdata=imread (inputfile_name);
            fig2=figure('BackingStore', 'on', 'IntegerHandle'...
                , 'off', 'Name', 'Original plot' ...
            ,'NumberTitle', 'off', 'MenuBar', 'none',
'DoubleBuffer','on'...
            );
```

```
            image (imdata);
            axis off;
    end
function [action]=refresh(src,evnt)
            delete(gcf)
            close all
            adj_matrix_gui('init');
function save_current_plot(src,evnt)
          my = getappdata(gcf,'mypoints'); % get the points data
from the handle gcf
          Ma = getappdata(gcf,'Matrix');   % get the matrix data
from the handle gcf
          new=inputdlg('Enter the  file name to restore the
output data :',...
                        'Require a name', 1,{'outputdata.txt'});
          options.Resize='on';
          options.WindowStyle='normal';
          options.Interpreter='tex';
            if strcmp(new,'')==1 %if there is no input
                errordlg('Invalid input','File Error')
                return
            elseif length(new)==0
                return
            else file_name=new{1};

          OutPutVertex_Matrix_ToFile( file_name,my,Ma); %Output
the vertex and matrix to the txt file
            end
function helpF(src,evnt)
open help.txt
```

2. *calc_metrics:*

```
   function [result,message] =  calc_metrics(Matrix)
nodes=length(Matrix);
links=sum(sum(Matrix))/2;
AvgNodeDegree=sum(sum(Matrix))/nodes;
MaxNodeDegree=max(sum(Matrix));
Connectedness=(2*links)/(nodes*(nodes-1));
Ivd=0;
e=0;
for i=1:nodes
    c=(sum(Matrix));
    b=c(i)*log2(c(i));
    Ivd=Ivd+b;
    ClusteringCoeff(i)=2/(c(i)+1);
    e=e+ClusteringCoeff(i);
end
NormalizedInfoContent=Ivd/(nodes*(nodes-1)*log2(nodes-1));
AvgClusteringCoeff=e/nodes;
Bnode=bridgeNode(Matrix);
```

```
LiMat(nodes,1) = 0; %for storing Li values for the each node for
unweighted graph
MatTemp(nodes,nodes) = 0;%temproray matrix
for i=1:nodes %clustering coefficient Li
    n=1;
    for j=1:nodes
        if Matrix(i,j)~= 0
            MatTemp(i,n) = j; %stores the values of nodes
connected with each node
            n=n+1;
        end
    end

    for k=1:(n-1)
        for l=(k+1):(n-1)
            b = MatTemp(i,k);
            c = MatTemp(i,l);
            if Matrix(b,c)~=0
                LiMat(i,1) = LiMat(i,1)+1;
            end
        end
    end
end
ki =sum(Matrix);%stores the values of number of nodes connected
with each node
ClusteringCoeff = 0;
ClusteringCoeff = 0;
ClusteringCoeffwt = 0;

for i=1:nodes
    if (ki(i)*(ki(i)-1)) ~= 0
    ClusteringCoeff= ClusteringCoeff
+((2*LiMat(i,1))/(ki(i)*(ki(i)-1)));%calculates the clustering
coefficient for unweighted graph
    end
end

AvgClusteringCoeff=ClusteringCoeff/nodes; %calculates average
clustering coeeficient for unweighted graph

% delete(message)
s={strcat('1-nodes=', num2str(nodes)); strcat('2-links=',
num2str(links)); strcat('3-AvgNodeDegree=',
num2str(AvgNodeDegree)); strcat('4-MaxNodeDegree=',
num2str(MaxNodeDegree)); strcat('5-Connectedness=',
num2str(Connectedness)); strcat('6-Ivd=', num2str(Ivd)) ;
strcat('7-NormalizedInfoContent=',
num2str(NormalizedInfoContent)) ; strcat('8-AvgClusteringCoeff=',
num2str(AvgClusteringCoeff));strcat('9-Bridge Nodes=',
num2str(Bnode))};
ppp=msgbox(s,'Results');
set(ppp,'Position',[750 500 150 150]);
% waitfor(ppp)
```

```
message=ppp;

result = [nodes links AvgNodeDegree MaxNodeDegree Connectedness
Ivd NormalizedInfoContent AvgClusteringCoeff Bnode];
```

3. OpenData:

```
function [Matrix,h,j]= openData(fig,a)
gcf=fig;
fid = fopen (a,'r');
if fid==-1
    j=-1;
    errordlg('Invalid input', 'Cannot find the file name')
    h=0;
    Matrix=[];
    oldPoints=[];
else
    j=1;

numberOfNodes=fscanf(fid,'%f',1);
for i=1:numberOfNodes
    Matrix(i,:)=fscanf(fid,'%f',numberOfNodes);
end

for i=1:numberOfNodes
    z(i)=1;
end

%Plotting nodes
fig;
hold on
t = 0:(2*pi)/numberOfNodes:2*pi;
x=sin(t);
y=cos(t);
for i=1:numberOfNodes
%       plot(x(i),y(i),'Color','r');  %  ,'o','MarkerSize',15
        n = 1+length(findobj(get(gca,'Children'),'Type','text'));
        h = text(x(i),y(i),num2str(n) ...

,'Color','r','FontWeight','bold','FontSize',16);

    hold on;
end

% Plotting Links
for i=1:numberOfNodes
    for j=1+i:(numberOfNodes)
        if i~=j && Matrix(i,j)~=0
            u=[x(i),x(j)];
```

```
            v=[y(i),y(j)];
            line(u,v);
            hold on;
        end
    end
end

% h=gcf;
if isappdata(gcf,'message')
            delete(getappdata(gcf,'message'));
end
      fig=gcf;
      [result,message] = calc_metrics(Matrix);
      setappdata(fig,'message',message)
      h=fig;
      plot_charts(result,4);
end

```

4. Plot charts:

```
function [h1] = plot_charts(metric_ar,handl)
[rows,cols] = size(metric_ar);

    if (isempty(handl))
        figure;
    else
        figure(handl);
        set(handl,'Name', 'Complexity Metrics','NumberTitle',
'off','MenuBar', 'none', 'DoubleBuffer','on','Position',[900 50
(480) (480)]);
    end;

% figure;
    label =
['nodes,links,AvgNodeDegree,MaxNodeDegree,Connectedness,Ivd,Norma
lizedInfoContent,AvgClusteringCoeff'];

for i = 1:rows
            subplot(1,rows,i);
            bar(metric_ar(i,:));
    end;
h1=gcf;
```

5. OutPutVertex_Matrix_ToFile:

```
function  OutPutVertex_Matrix_ToFile( file_name,my,Ma)

%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here

fid=fopen(file_name,'wt');
%write the Matrix
%a is the row
%b is the column
[a,b]=size(Ma);
fprintf(fid,  '%1.3e\t\t\n  ', a );

for i=1:a
      for j=1:b
      fprintf(fid,  ' %1.3e\t', Ma(i,j) );
      end
    fprintf(fid,  '\n  ' );
end

%write the vertex coordinates
[a,b]=size(my);

msgbox('File has been saved')
fclose(fid);
end
```

APPENDIX E

MATLAB CODE FOR WEIGHTED COMPLEXITY METRICS

The Matlab code has 5 functions:

1. adj_matrix_gui:

```
function [file_name]=adj_matrix_gui(action)

% If there is no map, just initiate the plot
if nargin == 0
    action = 'init';
end

switch action
case 'motion'
    line_h = getappdata(gcf,'motionline');
    pt = get(gca,'CurrentPoint');
    pt = pt(1,:);
    xdata = get(line_h,'XData');
    ydata = get(line_h,'YData');
    xdata(2) = pt(1);
    ydata(2) = pt(2);
    set(line_h,'XData',xdata,'YData',ydata)
case 'down'
    button = get(gcf,'SelectionType');
    switch button
    case 'normal'
        %get the information of the clicked point and pass it
from gco to h
        h = gco;
        fig = gcf;
        % First click
        if ~isappdata(fig, 'motionline')
            %find the point by the text, h is the position of the
first
            %point
            if isequal(get(h,'Type'),'text')
                pt = get(h,'Position');
                hold on
                line_h = plot(pt(1), pt(2),'b-' ...
                                          ,'EraseMode','normal');
                setappdata(line_h,'startobj',h)    % Save start
object in the line_h from h
                hold off
                stack_text_on_top
                setappdata(fig,'motionline',line_h)
                set(fig,'WindowButtonMotionFcn',
'adj_matrix_gui(''motion'')');
            end
        else
        % Second click
            line_h = getappdata(fig,'motionline');
            %find the point by the text
            if isequal(get(gco,'Type'),'text')
                startobj = getappdata(line_h,'startobj');
```

```
                endobj = gco;% Save start object in the line_h
from h
                startpt = get(startobj,'Position');
                endpt = get(endobj,'Position');
                set(line_h,'XData',[startpt(1) endpt(1)] ...
                          ,'YData',[startpt(2) endpt(2)]);
                I = round(str2double(get(startobj,'String')));
                J = round(str2double(get(endobj,'String')));
                Matrix = getappdata(gcf,'Matrix');

                llw=inputdlg('Enter the link weight:');
                lw=str2double(llw{1,1})
                Matrix(I,J) = Matrix(I,J)+lw;%add a relation in
the matrix
                Matrix(J,I) = Matrix(J,I)+lw;
                setappdata(gcf,'Matrix',Matrix)
                Matrix
            else
                delete(line_h)% If doesn't get a text--point,
stop plotting line
            end
            rmappdata(gcf,'motionline')
            set(fig,'WindowButtonMotionFcn', '');
            if isappdata(gcf,'message')
                try
            delete(getappdata(gcf,'message'));
                end
             end
        fig=gcf;
            [result,message] =
calc_metrics(getappdata(gcf,'Matrix'));
            setappdata(fig,'message',message)
            plot_charts(result,4) ;
        end

    case 'open'
        h = gco;
         if ~isequal(get(h,'Type'),'text')%If it does exist a
point at the
             %current click, start a new one
                pt = get(gca,'CurrentPoint');
                pt = pt(1,:);
                hold on
                n =
1+length(findobj(get(gca,'Children'),'Type','text'));
                h = text(pt(1),pt(2),num2str(n) ...

,'Color','r','FontWeight','bold','FontSize',16);
                        %define the size of the text and the
color
                hold off
                if ~isappdata(gcf,'Matrix')
                    setappdata(gcf,'Matrix',[])
                end
```

```
                if ~isappdata(gcf,'mypoints')
                    setappdata(gcf,'mypoints',[])
                end
                mypoints = getappdata(gcf,'mypoints');
                mypoints(n,:)=pt;
                setappdata(gcf,'mypoints',mypoints)
                mypoints;
                Matrix = getappdata(gcf,'Matrix');
                nnw=inputdlg('Enter the node weight:');
                nw=str2double(nnw{1,1})
                Matrix(n,n) = nw;
                setappdata(gcf,'Matrix',Matrix)
                Matrix
                if isappdata(gcf,'message')
                  try  delete(getappdata(gcf,'message'));
                  end
                end
                fig=gcf;
                [result,message] =
calc_metrics(getappdata(gcf,'Matrix'));
                setappdata(fig,'message',message)
                plot_charts(result,4) ;
         end

    case 'alt'
     %Use the right click to delet components.
     fig=gcf;
     switch get(gco,'Type')
        case 'text'
            the_component_to_be_delete=gco;
            %If the mouse click on the text then delete the vetex
            n = round(str2double(get(gco,'String')));
            pt = get(gco,'Position');
            %at this time the handles include the text messages,
the
            %position of the position of the points and the
lines, so in
            %the *****line, it's need to seperate the information
of the
            %points and information of the lines. The lines are
what we
            %needed at that line.
            handles = get(gca,'Children');
            for I=1:length(handles)
                h = handles(I);
                if isequal(get(h,'Type'),'text')%Renumber the
text
                    n2 = round(str2double(get(h,'String')));
                    if n2 > n
                        set(h,'String',n2-1)
                    end
                else
                    xdata = get(h,'XData');
                    ydata = get(h,'YData');
```

```
                    if length(xdata)==2 %***** to find the line
that link to the chosen point
                    if (xdata(1) == pt(1) & ydata(1) == pt(2))
...
                    |  (xdata(2) == pt(1) & ydata(2) == pt(2))

                delete(h)% to delete the line that link to the
chosen point
                    end
                    end
                end
            end
            if isappdata(gcf,'Matrix')
                Matrix = getappdata(gcf,'Matrix');
                Matrix(n,:) = [];
                Matrix(:,n) = [];
%                 mypoints = getappdata(gcf,'mypoints');
%                 mypoints(n,:) = [];
                setappdata(gcf,'Matrix',Matrix)
%                 setappdata(gcf,'mypoints',mypoints)
                %show the coordinate and Matrix on the screen
%                 mypoints
                Matrix



            end
            delete(the_component_to_be_delete)%Delete the point
in the form of 'text'
            if isappdata(gcf,'message')
            delete(getappdata(gcf,'message'));
            end
        fig=gcf;
            [result,message] =
calc_metrics(getappdata(gcf,'Matrix'));
            setappdata(fig,'message',message)
        case 'line'
            the_component_to_be_delete=gco;
             %If the mouse click on the line then delete the line
            xdata = get(gco,'XData');
            ydata = get(gco,'YData');
            txt_h = findobj(get(gca,'Children'),'Type','text');
            for K=1:length(txt_h)
                h = txt_h(K);
                pt = get(h,'Position');
                if (xdata(1) == pt(1) & ydata(1) == pt(2))
                    I = round(str2double(get(h,'String')));
                elseif (xdata(2) == pt(1) & ydata(2) == pt(2))
                    J = round(str2double(get(h,'String')));
                end
            end
            if isappdata(gcf,'Matrix')
                Matrix = getappdata(gcf,'Matrix');
                Matrix(I,J) = 0;
```

```
                Matrix(J,I) = 0;

                setappdata(gcf,'Matrix',Matrix)
                my = getappdata(gcf,'mypoints');
                Ma = getappdata(gcf,'Matrix');
                Matrix
            end
            delete(the_component_to_be_delete)%Delete the line
            if isappdata(gcf,'message')
            try
            delete(getappdata(gcf,'message'));
            end
            end
                         fig=gcf;
            [result,message] =
calc_metrics(getappdata(gcf,'Matrix'));
            setappdata(fig,'message',message)
         otherwise return
        end % End object switch
         plot_charts(result,4);%Finish deleting the components.


    end % End button switch
case 'keypress'
    ESC = 27;%press ESC to stop drawing the line
    SPACE = 32;%press space to out put the vertex and matrix to
the txt file
    switch get(gcf,'CurrentCharacter')
    case ESC
        if isappdata(gcf,'motionline')
            line_h = getappdata(gcf,'motionline');
            delete(line_h)
            rmappdata(gcf,'motionline')
        end

     case SPACE
          my = getappdata(gcf,'mypoints'); % get the points data
from the handle gcf
          Ma = getappdata(gcf,'Matrix');   % get the matrix data
from the handle gcf
          new=inputdlg('Enter the  file name to restore the
output data :',...
                        'Require a name', 1,{'outputdata.txt'});
          options.Resize='on';
          options.WindowStyle='normal';
          options.Interpreter='tex';
            if strcmp(new,'')==1 %if there is no input
                errordlg('Invalid input','File Error')
                return
            elseif length(new)==0
                return
            else file_name=new{1};
```

```
          OutPutVertex_Matrix_ToFile( file_name,my,Ma); %Output
the vertex and matrix to the txt file
          end
     otherwise set(gcf,'WindowButtonMotionFcn', '');

    end



 %Initiate the plot
case 'init'
    scrsz = get(0,'ScreenSize');
    fig = figure('BackingStore', 'on', 'IntegerHandle', 'off',
'Name'...
                , 'Adjacency Matrix', 'NumberTitle', 'off',
'MenuBar',...
                'none', 'DoubleBuffer','on','Position'...
                ,[10 50 (scrsz(3)-400) (scrsz(4)-100)]);
        Operations=uimenu('Parent',fig,'Label','Operations') ;
        uimenu(Operations,'Label','Open
File','Callback',@refresh);
        uimenu(Operations,'Label','Save','Callback',...
            @save_current_plot);
%         uimenu(f,'Label','Save','Callback','save');

uimenu(Operations,'Label','Quit','Callback',@my_closefcn,...
           'Separator','on','Accelerator','Q');
        pictures=uimenu('Parent',fig,'Label','Pictures') ;
        uimenu(pictures,'Label','Original
Structure','Callback'...
            , @plot_the_image);
        uimenu(pictures,'Label','New Structure','Callback',...
            @plot_the_image2);% set the menu for plotting the
physical system
        HelpF=uimenu('Parent',fig,'Label','Help') ;
        uimenu(HelpF,'Label','Help file','Callback',@helpF);
%   ax  = axes;
%     axis off;
    title('Double click to create vertex. Single click to
connect. Right click to delete. Space to store the file.')
    xlim([-1.1 1.1]);
    ylim([-1.1 1.1]);
    selection = questdlg('Use an old file or start a new one
',...
      'New file or not',...
      'New File','Old File','New File');
   switch selection,
      case 'New File',

      case 'Old File'
        %importing file
        status=0;
        new =inputdlg('Enter the  file name :',...
```

```
    'File name', 1,{'outputdata.txt'});
    if length(new)==0

        status=1;
    elseif (strcmp(new,'')==1) | (~(length(new{1})>4 &&
strcmp(new{1}(end-3:end),'.txt'))) %if there is no input
        errordlg('Invalid input','File Error')
        status=1;
    else inputfile_name=new{1};
    end
      if status==1
      else
          if ~isappdata(gcf,'Matrix')
            setappdata(gcf,'Matrix',[])
        end
        if ~isappdata(gcf,'mypoints')
            setappdata(gcf,'mypoints',[])
        end
        [Matrix,h,j]=openData(fig,inputfile_name);
        if j==-1
        else
        fig=h;
%         setappdata(fig,'mypoints',oldPoints)
        setappdata(fig,'Matrix',Matrix) ;
      end
      end
       otherwise return
   end
    set(fig,'CloseRequestFcn',@my_closefcn)
    set(fig,'WindowButtonDownFcn', 'adj_matrix_gui(''down'')');
    set(fig,'KeyPressFcn','adj_matrix_gui(''keypress'')')
    figure(fig)

otherwise
    error(['Unknown - ' action])

end % End action switch

function stack_text_on_top
    ax = gca;
    handles = get(gca,'Children');
    txt_h = findobj(handles,'Type','text');
    set(gca,'Children',[txt_h; setdiff(handles,txt_h)])
function my_closefcn(src,evnt)
% User-defined close request function
% to display a question dialog box
   selection = questdlg('Closing This Figure? The Data will be
Saved after closed ',...
      'Close Request Function',...
      'Save and Close','Close','Cancel','Save and Close');
   switch selection,
      case 'Save and Close',
          if 1
```

```
              my = getappdata(gcf,'mypoints'); % get the points
data from the handle gcf
              Ma = getappdata(gcf,'Matrix');    % get the matrix
data from the handle gcf
              new =inputdlg('Enter the  file name to restore the
output data :',...
                            'Require a name',
1,{'outputdata.txt'});% Give name of the output data
              if strcmp(new,'')==1 %if there is no input
                 errordlg('Invalid input','File Error')
                    return
              else file_name=new{1};% Store name of the output
data
              end

              delete(gcf)
              close all  %Close the current windows
              OutPutVertex_Matrix_ToFile( file_name,my,Ma)
%Generate the output data
          end
      case 'Close'
          delete(gcf)
          close all %Close the current windows
      case 'Cancel'
           return   %Do not close windows and continue plot
   end
function plot_the_image(src,evnt) %plot the original physical
relation
        status=0;
        new =inputdlg('Enter the  file name that restoring the
data :',...
                        'Require a name',
1,{'CircuitDiagram.jpg'});
    if strcmp(new,'')==1 %if there is no input
        errordlg('Invalid input','File Error')
        status=1;
    elseif length(new)==0
        status=1;
    else inputfile_name=new{1};
    end
    if status==1
    else    imdata=imread (inputfile_name);
            fig2=figure('BackingStore', 'on', 'IntegerHandle'...
                , 'off', 'Name', 'Original plot' ...
            ,'NumberTitle', 'off', 'MenuBar', 'none',
'DoubleBuffer','on'...
            ,'Position',[500 300 500 400]);
            image (imdata);
            axis off;
      end
function plot_the_image2(src,evnt) %plot the new physical
relation
        status=0;
```

```
        new =inputdlg('Enter the  file name that restoring the
data :',...
                        'Require a name',
1,{'ModifiedCircuit.jpg'});
    if strcmp(new,'')==1 %if there is no input
        errordlg('Invalid input','File Error')
        status=1;
    elseif length(new)==0
        status=1;
    else inputfile_name=new{1};
    end
    if status==1
    else    imdata=imread (inputfile_name);
            fig2=figure('BackingStore', 'on', 'IntegerHandle'...
                , 'off', 'Name', 'Original plot' ...
            ,'NumberTitle', 'off', 'MenuBar', 'none',
'DoubleBuffer','on'...
            );
            image (imdata);
            axis off;
    end
function [action]=refresh(src,evnt)
            delete(gcf)
            close all
            adj_matrix_gui('init');
function save_current_plot(src,evnt)
          my = getappdata(gcf,'mypoints'); % get the points data
from the handle gcf
          Ma = getappdata(gcf,'Matrix');   % get the matrix data
from the handle gcf
          new=inputdlg('Enter the  file name to restore the
output data :',...
                        'Require a name', 1,{'outputdata.txt'});
          options.Resize='on';
          options.WindowStyle='normal';
          options.Interpreter='tex';
            if strcmp(new,'')==1 %if there is no input
                errordlg('Invalid input','File Error')
                return
            elseif length(new)==0
                return
            else file_name=new{1};

          OutPutVertex_Matrix_ToFile( file_name,my,Ma); %Output
the vertex and matrix to the txt file
            end
function helpF(src,evnt)
open help.txt
```

2. calc_metrics:

```
function [result,message] =  calc_metrics(Matrix)
nodes=length(Matrix);% calculates the no. of nodes

%Separating node weight and link matrix
for i=1:nodes
    NodeWeight(i) = Matrix(i,i);
    Matrix(i,i) = 0;
end


Matrix
NodeWeight
nodes
%Building a uinweighted matrix
MatrixUnw(nodes,nodes)=0;
for i=1:nodes
    for j=1:(nodes)
        if Matrix(i,j)~=0 %changed from equal to 1 to not equal
to 0
           MatrixUnw(i,j) = 1;
        end
    end
end
%calculations for unweighted matrix
links=sum(sum(MatrixUnw))/2;
AvgNodeDegree=sum(sum(MatrixUnw))/nodes;
MaxNodeDegree=max(sum(MatrixUnw));
Connectedness=(2*links)/(nodes*(nodes-1));
OverallWeight=nodes+links;
Mobility = (3*(links-nodes-1)) + nodes;

%calculation for weighted matrix
OverallNodeWt=sum(NodeWeight);
NormNodeWt=OverallNodeWt/(nodes);
OverallLinkwt=sum(sum(Matrix))/2;
NormLinkWt=OverallLinkwt/(links);
WtAvgNodeDegree=(sum(sum(Matrix)))/nodes;
MaxNodeDegreeWt=max(sum(Matrix));
ConnectednessWt=OverallLinkwt/(nodes*(nodes-1));%calculate
connectedness for weighted graph
OverallWeightWt=OverallNodeWt+OverallLinkwt;
Mobilitywt = (3*(links-nodes-1)) + OverallNodeWt;

%calculation of Ivd for weighted and unweighted
Ivd=0;
Ivdw=0;
e=0;

for i=1:nodes
    c=(sum(MatrixUnw));%clalculates the information content of
node degree for unweighted graph
    b=c(i)*log2(c(i));
```

```
    Ivd=Ivd+b;

    m=(sum(Matrix));%calculates the information content of the
node degree for weighted graph
    n=(m(i)*log2(m(i)))*NodeWeight(1,i);
    Ivdw=Ivdw+n;
end

NormalizedInfoContentwt=Ivdw/(3*nodes*(10*(nodes-
1))*log2(10*(nodes-1)));% Norm. Info. content for weighted graph
NormalizedInfoContent=Ivd/(nodes*(nodes-1)*log2(nodes-
1));%calculates Normalized information content for unweighted
graph

LiMat(nodes,1) = 0; %for storing Li values for the each node for
unweighted graph
MatTemp(nodes,nodes) = 0;%temproray matrix
for i=1:nodes %clustering coefficient Li
    n=1;
    for j=1:nodes
        if MatrixUnw(i,j)~= 0
            MatTemp(i,n) = j; %stores the values of nodes
connected with each node
            n=n+1;
        end
    end

    for k=1:(n-1)
        for l=(k+1):(n-1)
            b = MatTemp(i,k);
            c = MatTemp(i,l);
            if MatrixUnw(b,c)~=0
                LiMat(i,1) = LiMat(i,1)+1;
            end
        end
    end
end



ki =sum(MatrixUnw);%stores the values of number of nodes
connected with each node
ClusteringCoeff = 0;
ClusteringCoeffwt = 0;



for i=1:nodes
    if (ki(i)*(ki(i)-1)) ~= 0
    ClusteringCoeff= ClusteringCoeff
+((2*LiMat(i,1))/(ki(i)*(ki(i)-1)));%calculates the clustering
coefficient for unweighted graph
```

```
    ClusteringCoeffwt= ClusteringCoeffwt
+((LiMat(i,1))/(5*ki(i)*(ki(i)-1)));%calculates the clustering
coefficient for weighted graph
    end
end


AvgClusteringCoeff=ClusteringCoeff/nodes; %calculates average
clustering coeeficient for unweighted graph
AvgClusteringCoeffwt=ClusteringCoeffwt/nodes; %calculates average
clustering coeeficient for unweighted graph


% delete(message)
s={strcat('1-nodes=', num2str(nodes)); strcat('2-links=',
num2str(links)); strcat('3-AvgNodeDegree=',
num2str(AvgNodeDegree)); strcat('4-MaxNodeDegree=',
num2str(MaxNodeDegree)); strcat('5-Connectedness=',
num2str(Connectedness)); strcat('6-Ivd=', num2str(Ivd)) ;
strcat('7-NormalizedInfoContent=',
num2str(NormalizedInfoContent)) ; strcat('8-AvgClusteringCoeff=',
num2str(AvgClusteringCoeff)); strcat('9-OverallNodeWeight=',
num2str(OverallWeight)); strcat('10-Mobility=',
num2str(Mobility))};
ppp=msgbox(s,'Results');
set(ppp,'Position',[830 450 140 130]);
% waitfor(ppp)
message=ppp;
% Bnode=bridgeNode(Matrix);

%weighted matrix display
% delete(message)
t={strcat('1-nodes=', num2str(nodes)); strcat('2-
OverallNodeWeight=', num2str(OverallNodeWt)); strcat('3-
NormalizedNodeWeight=', num2str(NormNodeWt)); strcat('4-
OverallLinkWeight=', num2str(OverallLinkwt)); strcat('5-
NormalizedLinkWeight=', num2str(NormLinkWt)); strcat('6-
WeightedAvergaeNodeDegree=', num2str(WtAvgNodeDegree));
strcat('7-WeightedMaxNodeDegree=', num2str(MaxNodeDegreeWt));
strcat('8-WeightedConnectedness=', num2str(ConnectednessWt));
strcat('9-WegihtedInformationContent=', num2str(Ivdw));
strcat('10-WeightedNormalizedInfoContent=',
num2str(NormalizedInfoContentwt)); strcat('11-
WeightedClusteringCoefficient=', num2str(AvgClusteringCoeffwt));
strcat('12-NodeWt+LinkWt=', num2str(OverallWeightWt));strcat('13-
Mobility=', num2str(Mobilitywt))};
pppw=msgbox(t,'Results weighted');
set(pppw,'Position',[665 440 165 160]);
% waitfor(ppp)
message=ppp;
% Bnode=bridgeNode(Matrix);
IvdDisp = log(Ivd);
IvdwDisp = log(Ivdw);
```

```
result = [nodes links AvgNodeDegree MaxNodeDegree Connectedness
IvdDisp NormalizedInfoContent AvgClusteringCoeff OverallWeight
Mobility];
resultwt = [nodes links WtAvgNodeDegree MaxNodeDegreeWt
ConnectednessWt IvdwDisp NormalizedInfoContentwt
AvgClusteringCoeffwt OverallWeightWt Mobilitywt];

Plot_weighted(resultwt,5);
```

3. OpenData:

```
function [Matrix,h,j]= openData(fig,a)
gcf=fig;
fid = fopen (a,'r');
if fid==-1
    j=-1;
    errordlg('Invalid input', 'Cannot find the file name')
    h=0;
    Matrix=[];
    oldPoints=[];
else
    j=1;

numberOfNodes=fscanf(fid,'%f',1);
for i=1:numberOfNodes
    Matrix(i,:)=fscanf(fid,'%f',numberOfNodes);
end

for i=1:numberOfNodes
    z(i)=1;
end

%Plotting nodes
fig;
hold on
t = 0:(2*pi)/numberOfNodes:2*pi;
x=sin(t);
y=cos(t);
for i=1:numberOfNodes
%      plot(x(i),y(i),'Color','r');  %  ,'o','MarkerSize',15
        n = 1+length(findobj(get(gca,'Children'),'Type','text'));
        h = text(x(i),y(i),num2str(n) ...

,'Color','r','FontWeight','bold','FontSize',16);

    hold on;
end

% Plotting Links
for i=1:numberOfNodes
    for j=1+i:(numberOfNodes)
```

```
        if i~=j && Matrix(i,j)~=0
            u=[x(i),x(j)];
            v=[y(i),y(j)];
            line(u,v);
            hold on;
        end
    end
end

% h=gcf;
if isappdata(gcf,'message')
            delete(getappdata(gcf,'message'));
end
fig=gcf;
[result,message] = calc_metrics(Matrix);
setappdata(fig,'message',message)
h=fig;
plot_charts(result,4);
end
```

4. Plot charts:

```
function [h1] = plot_charts(metric_ar,handl)
[rows,cols] = size(metric_ar);

    if (isempty(handl))
        figure;
    else
        figure(handl);
        set(handl,'Name', 'Complexity Metrics','NumberTitle',
'off','MenuBar', 'none', 'DoubleBuffer','on','Position',[900 50
(480) (480)]);
    end;
% figure;
    label =
['nodes,links,AvgNodeDegree,MaxNodeDegree,Connectedness,Ivd,Norma
lizedInfoContent,AvgClusteringCoeff'];
    for i = 1:rows
%         for j = 1:cols

            subplot(1,rows,i);
            bar(metric_ar(i,:));
%                 xlim([0 10]);
%                 ylim([0 10]);

%         end
    end;
h1=gcf;

function [h2] = Plot_weighted(metric_ar,handl)
[rows,cols] = size(metric_ar);
```

```
    if (isempty(handl))
        figure;
    else
        figure(handl);
        set(handl,'Name', 'Weighted Complexity
Metrics','NumberTitle', 'off','MenuBar', 'none',
'DoubleBuffer','on','Position',[900 50 (480) (480)]);
    end;
% figure;
    label =
['nodes,links,AvgNodeDegree,MaxNodeDegree,Connectedness,Ivd,Norma
lizedInfoContent,AvgClusteringCoeff'];
    for i = 1:rows
%         for j = 1:cols

            subplot(1,rows,i);
            bar(metric_ar(i,:));
%                 xlim([0 10]);
%                 ylim([0 10]);

%         end
    end;
h2=gcf;
```

5. OutPutVertex_Matrix_ToFile:

```
function  OutPutVertex_Matrix_ToFile( file_name,my,Ma)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
fid=fopen(file_name,'wt');
%write the Matrix
%a is the row
%b is the column
[a,b]=size(Ma);
fprintf(fid,  '%1.3e\t\t\n  ', a );

for i=1:a
 for j=1:b
fprintf(fid,  ' %1.3e\t', Ma(i,j) );
  end
    fprintf(fid,  '\n  ' );
end

%write the vertex coordinates
[a,b]=size(my);
msgbox('File has been saved')
fclose(fid);

end
```

MATLAB Program for Cost Computation:

```
close all;
clear all;
clc;
a='IFV_cfg';
d='.txt';

for i=1:522
    b= num2str(i);
    e=strcat(a,b);
    c=strcat(e,d);
    Total_cost(i)=cost_analysis(c);
end
close all;

fid = fopen('cost_of_alternatives.txt', 'wt');
for i=1:522
    b= num2str(i);
    e=strcat(a,b);
    c=strcat(e,d);
fprintf(fid,'%10s = $%-d\n',e,Total_cost(i));
end
fclose(fid);
disp('Program execution complexted');


function [Total_cost]=cost_analysis(c)
close all;

%% cost of the components

Components ={'BK' 'BK-DUA Front Mid' 'BK-DUA Rear Mid'
'ECE_Hybrid_Controller' '3616_Diesel' 'C7_Diesel' 'C9_Diesel'
'C13_Diesel' 'C15_Diesel'...
    'C18_Diesel' 'D6V8 International' 'VU_ISG_V1' 'VU_ISG_V2'
'VU_ISG_V3' 'VU_ISG_V4' 'Saft_HEMV_5_One' 'Saft_HEMV_5_Three'
'SAFT HEMV-5 Rack'...
    '455' '485' 'BDC2' 'BDC4' 'DC02' 'DC07' 'dc12' 'IPU160'
'PCM' 'PCM-hr' 'CX28' 'CX31' };
%'HMPT_1000HP ' 'HMPT_1500HP ' 'HMPT800HP ' 'X300_12' 'LSG_2000'
% Component_cost=zeros(1,36);
Component_cost(:,:)=[4800 5000 5000 1000 40000 11075 13000 15500
18000 22000 12000 11000 11500 12000  12500 2500 4000  5500 4500
5000 3000 3500 ...
                        4000 4500 6000 7000 7500 8000 13900
15000];


% a = input('Enter the  file name : ', 's');
fid = fopen (c,'r');
k = 0;
```

```
Total_cost=0;
while ~feof(fid)
    curr = fgetl(fid);
    if ~isempty(curr)
       k = k+1;
       file_components{1,k}=curr;
    end
end

for i=1:k
    for j=1:30
        flag=strcmp(file_components{1,i},Components{1,j});
        if flag==1
            Total_cost=Total_cost+Component_cost(1,j);
            break;
        end
    end
end
fclose ('all');
```

APPENDIX F

WEIGHTS COMPLEXITY METRICS RESULTS FOR HYBRID POWERTRAIN ALTERNATIVES (REF [37])

Weights complexity metrics results

Design alternative #	nodes (N)	links (L)	N+L	Avg Node weight	Avg link weight	Avg Node degree	Connectedness	Ivd	Nivd	Clustering Coeff	Mobility	Weighted Connectedness	wt Clustering Coeff	nodes weight(N)	links weight (L)	Total Weighted size	wt Ivd	wt Nivd	wt avg node degree	wt Mobility
1	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
2	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
3	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
4	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
5	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
6	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
7	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
8	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
9	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
10	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
11	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
12	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
13	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
14	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
15	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
16	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
17	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
18	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
19	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
20	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
21	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
22	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
23	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
24	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
25	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
26	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
27	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
28	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
29	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
30	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
31	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
32	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
33	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
34	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20

35	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
36	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
37	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
38	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
39	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
40	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
41	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
42	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
43	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
44	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
45	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
46	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
47	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
48	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
49	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
50	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
51	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
52	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
53	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
54	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
55	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
56	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
57	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
58	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
59	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
60	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
61	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
62	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
63	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
64	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
65	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
66	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
67	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
68	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
69	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
70	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
71	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
72	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
73	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
74	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
75	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
76	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
77	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
78	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
79	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22

80	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
81	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
82	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
83	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
84	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
85	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
86	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
87	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
88	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
89	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
90	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
91	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
92	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
93	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
94	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
95	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
96	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
97	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
98	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
99	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
100	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
101	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
102	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
103	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
104	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
105	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
106	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
107	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
108	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
109	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
110	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
111	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
112	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
113	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
114	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
115	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
116	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
117	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
118	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
119	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
120	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
121	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
122	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
123	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
124	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19

125	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
126	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
127	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
128	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
129	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
130	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
131	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
132	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
133	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
134	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
135	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
136	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
137	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
138	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
139	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
140	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
141	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
142	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
143	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
144	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
145	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
146	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
147	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
148	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
149	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
150	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
151	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
152	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
153	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
154	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
155	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
156	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
157	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
158	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
159	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
160	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
161	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
162	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
163	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
164	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
165	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
166	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
167	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
168	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
169	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20

170	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
171	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
172	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
173	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
174	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
175	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
176	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
177	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
178	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
179	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
180	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
181	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
182	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
183	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
184	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
185	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
186	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
187	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
188	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
189	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
190	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
191	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
192	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
193	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
194	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
195	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
196	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
197	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
198	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
199	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
200	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
201	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
202	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
203	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
204	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
205	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
206	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
207	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
208	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
209	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
210	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
211	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
212	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
213	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
214	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21

215	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
216	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
217	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
218	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
219	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
220	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
221	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
222	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
223	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
224	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
225	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
226	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
227	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
228	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
229	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
230	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
231	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
232	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
233	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
234	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
235	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
236	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
237	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
238	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
239	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
240	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
241	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
242	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
243	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
244	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
245	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
246	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
247	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
248	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
249	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
250	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
251	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
252	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
253	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
254	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
255	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
256	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
257	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
258	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
259	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20

260	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
261	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
262	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
263	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
264	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
265	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
266	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
267	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
268	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
269	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
270	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
271	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
272	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
273	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
274	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
275	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
276	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
277	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
278	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
279	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
280	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
281	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
282	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
283	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
284	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
285	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
286	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
287	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
288	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
289	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
290	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
291	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
292	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
293	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
294	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
295	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
296	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
297	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
298	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
299	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
300	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
301	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
302	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
303	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
304	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20

305	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
306	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
307	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
308	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
309	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
310	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
311	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
312	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
313	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
314	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
315	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
316	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
317	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
318	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
319	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
320	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
321	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
322	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
323	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
324	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
325	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
326	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
327	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
328	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
329	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
330	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
331	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
332	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
333	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
334	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
335	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
336	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	66.63	0.002	2.67	21
337	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
338	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
339	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
340	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
341	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
342	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
343	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
344	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
345	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
346	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
347	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
348	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
349	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20

350	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
351	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
352	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	51.77	0.002	2.50	20
353	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
354	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
355	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
356	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
357	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
358	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
359	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
360	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
361	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
362	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
363	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
364	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
365	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
366	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
367	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
368	12	15	27	1.33	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	16	16	27	78.24	0.003	2.67	22
369	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
370	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
371	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
372	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
373	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
374	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
375	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
376	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
377	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
378	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
379	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
380	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
381	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
382	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
383	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
384	12	15	27	1.25	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	15	15	27	59.77	0.002	2.50	21
385	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
386	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
387	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
388	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
389	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
390	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
391	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
392	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
393	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
394	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21

395	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
396	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
397	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
398	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
399	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
400	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
401	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
402	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
403	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
404	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
405	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
406	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
407	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
408	12	15	27	1.17	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	14	16	27	61.87	0.002	2.67	20
409	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
410	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
411	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
412	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
413	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
414	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
415	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
416	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	47.02	0.002	2.50	19
417	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
418	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
419	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
420	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
421	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
422	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
423	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
424	12	15	27	1.25	1.07	2.50	0.23	42.26	0.09	0.13	18	0.12	0.01	15	16	27	73.48	0.003	2.67	21
425	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
426	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
427	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
428	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
429	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
430	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
431	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
432	12	15	27	1.17	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	14	15	27	55.02	0.002	2.50	20
433	12	14	26	0.92	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	11	14	26	35.51	0.001	2.33	14
434	12	14	26	1.00	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	12	14	26	43.51	0.002	2.33	15
435	12	14	26	0.92	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	11	14	26	35.51	0.001	2.33	14
436	12	14	26	0.92	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	11	14	26	35.51	0.001	2.33	14
437	12	14	26	1.00	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	12	14	26	43.51	0.002	2.33	15
438	12	14	26	1.00	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	12	14	26	43.51	0.002	2.33	15
439	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18

440	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
441	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
442	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
443	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
444	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
445	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
446	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
447	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
448	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
449	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
450	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
451	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
452	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
453	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
454	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
455	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
456	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
457	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
458	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
459	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
460	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
461	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
462	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
463	12	14	26	0.92	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	11	14	26	35.51	0.001	2.33	14
464	12	14	26	1.00	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	12	14	26	43.51	0.002	2.33	15
465	12	14	26	0.92	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	11	14	26	35.51	0.001	2.33	14
466	12	14	26	0.92	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	11	14	26	35.51	0.001	2.33	14
467	12	14	26	1.00	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	12	14	26	43.51	0.002	2.33	15
468	12	14	26	1.00	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	12	14	26	43.51	0.002	2.33	15
469	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
470	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
471	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
472	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
473	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
474	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
475	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
476	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
477	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
478	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
479	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
480	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
481	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
482	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
483	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
484	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18

485	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
486	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
487	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
488	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
489	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
490	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
491	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
492	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
493	12	14	26	0.92	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	11	14	26	35.51	0.001	2.33	14
494	12	14	26	1.00	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	12	14	26	43.51	0.002	2.33	15
495	12	14	26	0.92	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	11	14	26	35.51	0.001	2.33	14
496	12	14	26	0.92	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	11	14	26	35.51	0.001	2.33	14
497	12	14	26	1.00	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	12	14	26	43.51	0.002	2.33	15
498	12	14	26	1.00	1.00	2.33	0.21	37.51	0.08	0.13	15	0.11	0.01	12	14	26	43.51	0.002	2.33	15
499	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
500	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
501	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
502	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
503	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
504	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
505	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
506	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
507	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
508	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
509	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
510	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
511	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
512	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
513	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
514	12	15	27	1.00	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	12	15	27	42.26	0.002	2.50	18
515	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
516	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
517	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
518	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
519	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
520	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
521	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19
522	12	15	27	1.08	1.00	2.50	0.23	42.26	0.09	0.13	18	0.11	0.01	13	15	27	50.26	0.002	2.50	19

APPENDIX G

DATA FOR COST VERSUS COMPLEXITY CHART (REF [37])

Data for cost versus complexity chart (Ref [37])

Design #	Cost	Complexity
1	78600	261.12
2	78600	261.12
3	74600	261.12
4	74600	261.12
5	79600	267.96
6	79600	267.96
7	75600	267.96
8	75600	267.96
9	76100	242.94
10	76100	242.94
11	72100	242.94
12	72100	242.94
13	77100	249.78
14	77100	249.78
15	73100	249.78
16	73100	249.78
17	74100	274.81
18	74100	274.81
19	70100	274.81
20	70100	274.81
21	75100	281.65
22	75100	281.65
23	71100	281.65
24	71100	281.65
25	71600	253.02
26	71600	253.02
27	67600	253.02
28	67600	253.02
29	72600	259.86
30	72600	259.86
31	68600	259.86
32	68600	259.86
33	78600	261.12
34	78600	261.12
175	68600	259.86
176	68600	259.86
177	78600	261.12
178	78600	261.12
179	78600	261.12
180	78600	261.12
181	74600	261.12
182	74600	261.12
183	74600	261.12
184	74600	261.12
185	79600	267.96
186	79600	267.96
187	79600	267.96
188	79600	267.96
189	75600	267.96
190	75600	267.96
191	75600	267.96
192	75600	267.96
193	76100	242.94
194	76100	242.94
195	76100	242.94
196	76100	242.94
197	72100	242.94
198	72100	242.94
199	72100	242.94
200	72100	242.94
201	77100	249.78
202	77100	249.78
203	77100	249.78
204	77100	249.78
205	73100	249.78
206	73100	249.78
207	73100	249.78
208	73100	249.78
349	73100	249.78
350	73100	249.78
351	73100	249.78
352	73100	249.78
353	74100	274.81
354	74100	274.81
355	74100	274.81
356	74100	274.81
357	70100	274.81
358	70100	274.81
359	70100	274.81
360	70100	274.81
361	75100	281.65
362	75100	281.65
363	75100	281.65
364	75100	281.65
365	71100	281.65
366	71100	281.65
367	71100	281.65
368	71100	281.65
369	71600	253.02
370	71600	253.02
371	71600	253.02
372	71600	253.02
373	67600	253.02
374	67600	253.02
375	67600	253.02
376	67600	253.02
377	72600	259.86
378	72600	259.86
379	72600	259.86
380	72600	259.86
381	68600	259.86
382	68600	259.86

35	78600	261.12
36	78600	261.12
37	74600	261.12
38	74600	261.12
39	74600	261.12
40	74600	261.12
41	79600	267.96
42	79600	267.96
43	79600	267.96
44	79600	267.96
45	75600	267.96
46	75600	267.96
47	75600	267.96
48	75600	267.96
49	76100	242.94
50	76100	242.94
51	76100	242.94
52	76100	242.94
53	72100	242.94
54	72100	242.94
55	72100	242.94
56	72100	242.94
57	77100	249.78
58	77100	249.78
59	77100	249.78
60	77100	249.78
61	73100	249.78
62	73100	249.78
63	73100	249.78
64	73100	249.78
65	74100	274.81
66	74100	274.81
67	74100	274.81
68	74100	274.81
69	70100	274.81
70	70100	274.81
71	70100	274.81
72	70100	274.81
73	75100	281.65
209	74100	274.81
210	74100	274.81
211	74100	274.81
212	74100	274.81
213	70100	274.81
214	70100	274.81
215	70100	274.81
216	70100	274.81
217	75100	281.65
218	75100	281.65
219	75100	281.65
220	75100	281.65
221	71100	281.65
222	71100	281.65
223	71100	281.65
224	71100	281.65
225	71600	253.02
226	71600	253.02
227	71600	253.02
228	71600	253.02
229	67600	253.02
230	67600	253.02
231	67600	253.02
232	67600	253.02
233	72600	259.86
234	72600	259.86
235	72600	259.86
236	72600	259.86
237	68600	259.86
238	68600	259.86
239	68600	259.86
240	68600	259.86
241	79100	261.12
242	79100	261.12
243	75100	261.12
244	75100	261.12
245	76600	242.94
246	76600	242.94
247	72600	242.94
383	68600	259.86
384	68600	259.86
385	79100	261.12
386	79100	261.12
387	75100	261.12
388	75100	261.12
389	76600	242.94
390	76600	242.94
391	72600	242.94
392	72600	242.94
393	74600	274.81
394	74600	274.81
395	70600	274.81
396	70600	274.81
397	72100	253.02
398	72100	253.02
399	68100	253.02
400	68100	253.02
401	79100	261.12
402	79100	261.12
403	79100	261.12
404	79100	261.12
405	75100	261.12
406	75100	261.12
407	75100	261.12
408	75100	261.12
409	76600	242.94
410	76600	242.94
411	76600	242.94
412	76600	242.94
413	72600	242.94
414	72600	242.94
415	72600	242.94
416	72600	242.94
417	74600	274.81
418	74600	274.81
419	74600	274.81
420	74600	274.81
421	70600	274.81

74	75100	281.65
75	75100	281.65
76	75100	281.65
77	71100	281.65
78	71100	281.65
79	71100	281.65
80	71100	281.65
81	71600	253.02
82	71600	253.02
83	71600	253.02
84	71600	253.02
85	67600	253.02
86	67600	253.02
87	67600	253.02
88	67600	253.02
89	72600	259.86
90	72600	259.86
91	72600	259.86
92	72600	259.86
93	68600	259.86
94	68600	259.86
95	68600	259.86
96	68600	259.86
97	79100	261.12
98	79100	261.12
99	75100	261.12
100	75100	261.12
101	76600	242.94
102	76600	242.94
103	72600	242.94
104	72600	242.94
105	74600	274.81
106	74600	274.81
107	70600	274.81
108	70600	274.81
109	72100	253.02
110	72100	253.02
111	68100	253.02
112	68100	253.02

248	72600	242.94
249	74600	274.81
250	74600	274.81
251	70600	274.81
252	70600	274.81
253	72100	253.02
254	72100	253.02
255	68100	253.02
256	68100	253.02
257	79100	261.12
258	79100	261.12
259	79100	261.12
260	79100	261.12
261	75100	261.12
262	75100	261.12
263	75100	261.12
264	75100	261.12
265	76600	242.94
266	76600	242.94
267	76600	242.94
268	76600	242.94
269	72600	242.94
270	72600	242.94
271	72600	242.94
272	72600	242.94
273	74600	274.81
274	74600	274.81
275	74600	274.81
276	74600	274.81
277	70600	274.81
278	70600	274.81
279	70600	274.81
280	70600	274.81
281	72100	253.02
282	72100	253.02
283	72100	253.02
284	72100	253.02
285	68100	253.02
286	68100	253.02

422	70600	274.81
423	70600	274.81
424	70600	274.81
425	72100	253.02
426	72100	253.02
427	72100	253.02
428	72100	253.02
429	68100	253.02
430	68100	253.02
431	68100	253.02
432	68100	253.02
433	49075	212.14
434	44575	222.23
435	49075	212.14
436	49075	212.14
437	44575	222.23
438	44575	222.23
439	74575	236.10
440	74575	236.10
441	70575	236.10
442	70575	236.10
443	70075	246.19
444	70075	246.19
445	66075	246.19
446	66075	246.19
447	74575	236.10
448	74575	236.10
449	74575	236.10
450	74575	236.10
451	70575	236.10
452	70575	236.10
453	70575	236.10
454	70575	236.10
455	70075	246.19
456	70075	246.19
457	70075	246.19
458	70075	246.19
459	66075	246.19
460	66075	246.19

113	79100	261.12
114	79100	261.12
115	79100	261.12
116	79100	261.12
117	75100	261.12
118	75100	261.12
119	75100	261.12
120	75100	261.12
121	76600	242.94
122	76600	242.94
123	76600	242.94
124	76600	242.94
125	72600	242.94
126	72600	242.94
127	72600	242.94
128	72600	242.94
129	74600	274.81
130	74600	274.81
131	74600	274.81
132	74600	274.81
133	70600	274.81
134	70600	274.81
135	70600	274.81
136	70600	274.81
137	72100	253.02
138	72100	253.02
139	72100	253.02
140	72100	253.02
141	68100	253.02
142	68100	253.02
143	68100	253.02
144	68100	253.02
145	78600	261.12
146	78600	261.12
147	74600	261.12
148	74600	261.12
149	79600	267.96
150	79600	267.96
151	75600	267.96

287	68100	253.02
288	68100	253.02
289	78600	261.12
290	78600	261.12
291	74600	261.12
292	74600	261.12
293	79600	267.96
294	79600	267.96
295	75600	267.96
296	75600	267.96
297	76100	242.94
298	76100	242.94
299	72100	242.94
300	72100	242.94
301	77100	249.78
302	77100	249.78
303	73100	249.78
304	73100	249.78
305	74100	274.81
306	74100	274.81
307	70100	274.81
308	70100	274.81
309	75100	281.65
310	75100	281.65
311	71100	281.65
312	71100	281.65
313	71600	253.02
314	71600	253.02
315	67600	253.02
316	67600	253.02
317	72600	259.86
318	72600	259.86
319	68600	259.86
320	68600	259.86
321	78600	261.12
322	78600	261.12
323	78600	261.12
324	78600	261.12
325	74600	261.12

461	66075	246.19
462	66075	246.19
463	49075	212.14
464	44575	222.23
465	49075	212.14
466	49075	212.14
467	44575	222.23
468	44575	222.23
469	74575	236.10
470	74575	236.10
471	70575	236.10
472	70575	236.10
473	70075	246.19
474	70075	246.19
475	66075	246.19
476	66075	246.19
477	74575	236.10
478	74575	236.10
479	74575	236.10
480	74575	236.10
481	70575	236.10
482	70575	236.10
483	70575	236.10
484	70575	236.10
485	70075	246.19
486	70075	246.19
487	70075	246.19
488	70075	246.19
489	66075	246.19
490	66075	246.19
491	66075	246.19
492	66075	246.19
493	49075	212.14
494	44575	222.23
495	49075	212.14
496	49075	212.14
497	44575	222.23
498	44575	222.23
499	74575	236.10

152	75600	267.96
153	76100	242.94
154	76100	242.94
155	72100	242.94
156	72100	242.94
157	77100	249.78
158	77100	249.78
159	73100	249.78
160	73100	249.78
161	74100	274.81
162	74100	274.81
163	70100	274.81
164	70100	274.81
165	75100	281.65
166	75100	281.65
167	71100	281.65
168	71100	281.65
169	71600	253.02
170	71600	253.02
171	67600	253.02
172	67600	253.02
173	72600	259.86
174	72600	259.86

326	74600	261.12
327	74600	261.12
328	74600	261.12
329	79600	267.96
330	79600	267.96
331	79600	267.96
332	79600	267.96
333	75600	267.96
334	75600	267.96
335	75600	267.96
336	75600	267.96
337	76100	242.94
338	76100	242.94
339	76100	242.94
340	76100	242.94
341	72100	242.94
342	72100	242.94
343	72100	242.94
344	72100	242.94
345	77100	249.78
346	77100	249.78
347	77100	249.78
348	77100	249.78

500	74575	236.10
501	70575	236.10
502	70575	236.10
503	70075	246.19
504	70075	246.19
505	66075	246.19
506	66075	246.19
507	74575	236.10
508	74575	236.10
509	74575	236.10
510	74575	236.10
511	70575	236.10
512	70575	236.10
513	70575	236.10
514	70575	236.10
515	70075	246.19
516	70075	246.19
517	70075	246.19
518	70075	246.19
519	66075	246.19
520	66075	246.19
521	66075	246.19
522	66075	246.19