UNIVERSITY OF CALGARY

Optimal Design of Reconfigurable Products Considering Product Configurations and Reconfiguration Processes

by

Moustafa Gadalla

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN MECHANICAL AND MANUFACTURING ENGINEERING

CALGARY, ALBERTA

August, 2018

Abstract

A reconfigurable product is used as several products for delivering different functions through reconfiguration processes to change between product configurations during its utilization stage. Despite the research efforts devoted to design and manufacture of reconfigurable products, influences of reconfiguration processes have not been considered in the design of reconfigurable products. Since different designs can lead to different reconfiguration processes, and both the product configurations and reconfiguration processes influence the evaluation measures of the product, an optimal reconfigurable product design method is developed in this research considering both product configurations and reconfiguration processes.

First, a generic design AND-OR tree is used to model design candidates and design parameters based on the same requirements, and generic process AND-OR graphs are employed to model reconfiguration process candidates and process parameters. Specific design and reconfiguration process candidates are created from the generic design AND-OR tree and the generic process AND-OR graphs through tree/graph based search. Based on evaluations to different design candidates and reconfiguration process candidates, the optimal design solution is identified using a multi-level and multi-objective optimization method. The optimization is conducted at two different levels: the candidate optimization to achieve the optimal design/process candidates, and parameter optimization to identify the optimal design/process parameter values.

Second, the efficiency of the developed optimization method is improved when the numbers of design/process candidates are large. A heuristic is developed to evaluate nodes of the design/process tree/graphs considering the importance of these nodes to the optimal design, and some nodes that are unlikely to achieve the optimal solution are pruned from the tree/graphs. To further improve the optimization efficiency, a heuristic function is developed to rank the created design solutions to identify the top-ranked solutions for detailed optimization.

Third, a new extended analytic network process (EANP) method is developed to determine weights of the design/process nodes for improving the quality of the developed optimization method considering dependency relations among descriptions of design/process candidates and evaluation criteria. In addition, by selecting the evaluation criteria in the EANP method with the similar evaluation measures in the optimization process, better top-ranked solutions can be identified for detailed optimization.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Deyi Xue. Dr. Xue is the person with passion and giving me valuable guidance not only in the academic research but also in the attitude of life. In the past years, Dr. Xue has spent considerable time with me on research. I would like to thank him for his endless patience and insightful research comments.

I also wish to thank my supervisory committee members, Dr. Yiliu (Paul) Tu and Dr. Alejandro Ramirez-Serrano, for providing valuable insights, discussions, and encouragements to my research.

I gratefully acknowledge the support provided by the Egyptian Armament Authority (EAA) through its Ph.D. Study Scholarship, and the Natural Sciences and Engineering Research Council (NSERC) of Canada through its Discovery Grant. Without their financial help, this research would not have been achievable.

At last, a special thanks to my wife for her support and encouragement during my Ph.D. study.

Dedication

To My Sweet Wife *Mai Hamed*, My Beloved Sons

"To All of You, I Shall Be Thankful Forever"

Table of Contents

List of Tables

List of Figures and Illustrations

Chapter 1 Introduction

1.1 Background

Global competition in marketplace demands new products to serve for multiple tasks. Among various types of new products, reconfigurable products are developed to deliver multiple functions through reconfiguration processes to change between different product configurations during their utilization stages. For example, instead of using multiple large machine tools to achieve different functions that may rarely be required, a reconfigurable machine tool can be developed and used to achieve these functions through reconfiguration processes between different machine configurations to reduce the cost of the machine tool. In the past decade, many methodologies have been introduced for the design and manufacturing of reconfigurable products and systems.

Siddiqi, Weck, and Iagnemma (2006) identified three significant requirements on reconfigurability that should be addressed in the development of reconfigurable products and systems: multi-ability, evolution, and survivability. The multi-ability allows the reconfigurable product or system to achieve efficient utilization of modules or resources through non-concurrent multiple configurations for the planned functions. The evolution through modifications over time lets the reconfigurable product or system work for the unplanned functions. The survivability leads the reconfigurable product or system to be more insensitive (robust) to the deviations from expected design functions and functional performances during product utilization stages.

Various principles and behaviors on product reconfiguration processes have been identified in the researches on reconfigurable products. Weaver, Wood, and Jensen (2008) and Singh et al. (2009) developed the main transformation principles and facilitators through studies of various reconfigurable products. The transformation principles described the methods of configuration changes, while the transformation facilitators specified how the transformations were accomplished. Typical transformation principles included expand/collapse, expose/cover, and fuse/divide. Weaver, Wood, and Jensen (2008) developed a reconfigurable shower to demonstrate the transformation/reconfiguration principles. Singh et al. (2009) designed a reconfigurable motorcycle that could be changed into an all-terrain vehicle. Haldaman and Parkinson (2010) discovered the orientation principle for reconfiguration in which a new configuration was created through reorientations of some product modules. A new product was developed in this work and used as both on-ear headphone and over-ear headphone by using the four transformation principles.

Developments of reconfigurable products were primarily in the following areas: modular self-reconfigurable robots, reconfigurable manufacturing systems and tools, and morphing aircraft (Haldaman and Parkinson 2010). A modular self-reconfigurable robot (MSR) is a robotic system composed of various small robots/modules that can be connected in different ways to change its configuration to complete multiple tasks. It is called self-reconfigurable because it can be autonomously reconfigured without any external help. Feczko et al. (2015) identified three main advantages of the MSRs: flexibility, resistance to damage, and low cost. A reconfigurable machine tool (RMT) is a kind of reconfigurable manufacturing systems (RMSs) (Bi et al. 2008; Andersen et al. 2017) to provide the required functions to satisfy the changeable customer requirements. The first patent on "RMTs" was developed by Koren and Kota (1999) based on the advantages of dedicated machines and computer numerical controlled machines. Since then, considerable effort has been devoted to the development of RMTs (Gadalla and Xue 2017a). The development of morphing aircraft at NASA aimed at improving the reconfigurability in flight and control. A morphing aircraft was designed to be able to change its shape for different requirements. For example, when the external weather condition changed, the geometries of the adaptive wings of the aircraft were then changed to optimize the airflow during the mission (Barbarino et al. 2011).

Modeling, analysis, and computer tools have also been developed for the design of reconfigurable products. In this area, Siddiqi, Weck, and Iagnemma (2006) developed a modeling methodology for design and analysis of the reconfigurable planetary surface vehicles (PSVs) for future manned space exploration missions. These PSVs aimed at reducing the mission costs through different configurations for different requirements. The non-homogeneous Markov model (NHMM) and control theory were employed for the design and evaluation of the PSVs.

1.2 Problem Statements

Despite the research efforts devoted to the design and manufacturing of reconfigurable products, influences of reconfiguration processes to change between product configurations on the performance of reconfigurable products have not been considered in the design of reconfigurable products. Since efforts of reconfiguration processes, such as time and cost, also influence customer satisfaction of the reconfigurable product, a systematic approach for optimal design of reconfigurable products considering both the product configurations and reconfiguration processes need to be developed.

Through extensive literature review and investigation, the following problems and requirements have been identified for optimal design of reconfigurable products considering both product configurations and reconfiguration processes.

(1) Modelling, evaluation, and optimization schemes are required for optimal design of reconfigurable products considering product configurations and reconfiguration processes.

- *Modelling of different design candidates, product configurations, design parameters, reconfiguration processes, and process parameters*

Despite the progress in research on modeling of reconfigurable products, the developed methods only focused on modelling of product configurations. Modelling of reconfiguration processes to change between product configurations has never been studied. Since different design candidates can lead to different reconfiguration processes, and both product configurations and reconfiguration processes influence the evaluation measures of the reconfigurable products, a systematic approach to model both different product configurations and reconfiguration processes is needed. In addition, modeling of product configuration parameters and reconfiguration process parameters should also be considered.

- *Evaluations of the reconfigurable product considering both the product configurations and reconfiguration processes*

In the past, only the evaluations to product configurations were considered for optimal design of reconfigurable products. Since customer satisfaction is also influenced by the evaluation measures to the reconfiguration processes, a new evaluation method considering both the product configurations and reconfiguration processes is required. In addition, since different evaluation measures to product configurations and reconfiguration processes are usually in different units such as performance, time and cost, a systematic evaluation method is needed to integrate all these different evaluation measures with different units into the same environment.

- *Optimization considering both the product configurations and reconfiguration processes*

In the previous studies, only product configurations were considered for optimal design of a reconfigurable product. Since reconfiguration processes also influence customer satisfaction, a new optimization method to identify both the optimal product configurations and reconfiguration processes is required. In addition, identification of the optimal product configuration parameters and reconfiguration process parameters should also be considered.

(2) Efficiency of optimization method should be improved for the optimal design of reconfigurable product when large numbers of design candidates and reconfiguration process candidates are considered.

- *Improvement of optimization efficiency by reducing numbers of design candidates and reconfiguration process candidates*

When large number of design candidates can be created from the same design requirements, and large number of reconfiguration process candidates can be obtained to change between two product configurations, significant computation effort is required to identify the optimal design candidate and the optimal reconfiguration processes to change between any two configurations of this design candidate. Since the specific design candidates and reconfiguration processes are usually identified from generic design and process descriptions, contributions of these partial design and process descriptions to the optimal design of reconfigurable product should be evaluated, such that the design and process descriptions that are unlikely to lead to the optimal design are not considered for creation of the design and process candidates.

- *Further improvement of optimization efficiency by initial ranking of the design and reconfiguration process candidates to select the top-ranked candidates for detailed optimization*

When numbers of created design candidates and reconfiguration process candidates are still large, further improvement of optimization efficiency has to be considered. Since heuristics are often used to provide initial evaluations in optimization, simple heuristic functions need to be introduced to provide initial evaluations to the design and the reconfiguration process candidates, and to select the top-ranked candidates for detailed evaluations and optimization.

(3) Quality of optimization method should be improved for the optimal design of reconfigurable products when dependency relations among design/process descriptions and evaluation criteria are considered for rating of contributions of these partial design/process descriptions to the optimal design.

- *Modelling and evaluation considering dependency relations among design/process descriptions and evaluation criteria*

Since contributions of the partial design and process descriptions to the optimal design of reconfigurable product are first ranked such that the less-important design and process descriptions are not considered for creation of the design and process candidates, accurate ranking of the design and process descriptions plays an important role to the improvement of the optimization quality. When decisions to select partial design/process descriptions are influenced by decisions to select other design/process descriptions due to the dependency relations among design/process descriptions and evaluation criteria, modeling of these dependency relations among design/process descriptions and evaluation criteria is required for accurate ranking of these partial design and process descriptions to the optimal design.

- *Initial evaluations to design/process descriptions considering different evaluation measures*

For ranking of the contributions of the partial design and process descriptions to the optimal design of reconfigurable product, heuristics are often used since the detailed evaluation measures cannot be achieved at this stage. Simple heuristics, such as the importance weights, are often employed to rank the partial solutions in the traditional optimization methods. Since the optimal reconfigurable product design is achieved considering multiple evaluation measures, a systematic method to rank the partial design and process descriptions based on heuristics similar to the evaluation measures is expected.

1.3 Research Objectives

The overall objective of this research is to develop a systematic approach for optimal design of reconfigurable products considering product configurations and reconfiguration processes. To achieve the overall objective of this research, first an integrated scheme is required for modeling, evaluation, and optimization of the reconfigurable product design. Second, efficiency of optimization should be improved to identify the optimal design solution of the reconfigurable product within the required computation time when large numbers of design and process candidates are considered. Third, quality of optimization should be improved when the dependency relations among design/process descriptions and evaluation criteria are considered.

Based on the three problems stated in Sections 1.2, the three objectives of this research are described as follows:

Objective 1: Development of modelling, evaluation, and optimization schemes for design of reconfigurable products considering both the product configurations and reconfiguration processes

(1.1) Development of a new modelling scheme to describe different design candidates, product configurations, reconfiguration processes, and parameters of product configurations and reconfiguration processes.

(1.2) Development of a new evaluation method to evaluate the reconfigurable product considering both the product configurations and reconfiguration processes.

(1.3) Development of a new optimization method to identify the optimal design solution that is modelled by product configurations, reconfiguration processes, and parameters of product configurations and reconfiguration processes.

Objective 2: Improvement of optimization efficiency for design of reconfigurable products considering large numbers of design candidates and reconfiguration process candidates

(2.1) Development of a new method to rank partial design and process descriptions and to delete the partial design and process descriptions that are unlikely to lead to the optimal reconfigurable product design for reducing the numbers of design candidates and reconfiguration process candidates.

(2.2) Development of a heuristic function to rank the created design solutions to identify the top-ranked solutions for detailed optimization.

Objective 3: Improvement of optimization quality for design of reconfigurable products when dependency relations among design/process descriptions and evaluation criteria are considered for rating of contributions of the partial design/process descriptions to the optimal design

(3.1) Development of a new modeling and initial evaluation method to rank the partial design/process descriptions considering dependency relations among design/process descriptions and evaluation criteria.

(3.2) Improvement of the initial evaluation quality by using heuristics that are similar to the multiple evaluation measures in the detailed optimization.

1.4 Research Summary

The results and major characteristics of this research are summarized as follows.

(1) Development of new modelling, evaluation, and optimization schemes for optimal design of reconfigurable products considering product configurations and reconfiguration processes.

From the design requirements, a generic AND-OR tree with design nodes and parameters of design nodes is used to model all design candidates, their product configurations and parameters of these configurations. Each node in the AND-OR tree is used to model a partial design solution (e.g., a sub-assembly or a component). The relations among sub-nodes of a super-node are classified into three categories: AND relation (i.e., composition relation) in which all sub-nodes should be used together to obtain/enable the parent node, OR relation in design (OR-D) in which only one sub-node should be selected to obtain/enable the parent node to create a specific design candidate, and OR relation in configuration (OR-C) when different sub-nodes are used to model different product configurations for delivering different functions. A specific design solution is created from the generic tree through a tree-based search (West 2001; Edelkamp and Schroedl 2011) and modeled by different product configurations.

For a reconfigurable product with multiple operation configurations, a reconfiguration process is defined to change between two product configurations. The generic reconfiguration process is modeled by an AND-OR graph with operation nodes, parameters of operation nodes, and sequential constraints among operation nodes. The relations among sub-nodes of a super-node are classified into three categories: AND relation (i.e., composition relation) in which all sub-nodes should be used together to obtain/enable the parent node, OR relation in operation (OR-O) when alternative operations are considered to achieve the same operation function, and OR relation in design condition (OR-DC) through IF-THEN descriptions when alternative operations are considered for different design candidates to achieve the same operation function. A graph-based search is used to generate all feasible reconfiguration process candidates from the generic reconfiguration process graph.

To identify the optimal design solution and its reconfiguration processes, first a solution with specific design candidate and specific reconfiguration processes are evaluated by multiple evaluation measures. Then the optimal design solution is identified by a multi-level and multi-objective optimization method. The optimization is conducted at two different levels: the candidate optimization level to identify the optimal design candidate that is modeled by its product configurations and reconfiguration processes, and the parameter optimization level to identify the optimal parameter values of product configurations and reconfiguration processes.

The effectiveness of the newly developed method is demonstrated by the design of a reconfigurable machine tool with three configurations, a vertical milling configuration, a horizontal milling configuration, and a turning configuration, for machining large-size components.

Characteristics in this research aspect are summarized as follows:

- The hybrid AND-OR tree and AND-OR graph are effective data structures to model different design candidates, product configurations, reconfiguration processes, and parameters of product configurations and reconfiguration processes. Different design solutions of the reconfigurable product can be created through tree-based search and graph-based search.
- Compared with the traditional methods for reconfigurable product design where only evaluations to product configurations are considered, evaluations to both the product configurations and reconfiguration processes are considered in this new design approach. In addition, by converting different evaluation measures in different units into comparable evaluation indices, the evaluations to both product configurations and reconfiguration processes from different perspectives are integrated into the same environment.
- The new multi-level and multi-objective optimization method is effective to identify the optimal design that is modeled by product configurations, reconfiguration processes, and parameters of product configurations and reconfiguration processes.

(2) *Improvement of optimization efficiency for optimal design of reconfigurable products considering large numbers of design candidates and reconfiguration process candidates.*

A new optimization method is developed to improve the computation efficiency when numbers of design candidates and reconfiguration process candidates are large. First, a heuristic is developed to evaluate nodes of the design tree and reconfiguration process graphs considering the importance of these nodes to the optimal design. Then the efficiency of tree/graph-based search is improved by pruning of some nodes in the AND-OR tree/graph that are unlikely to achieve the optimal solution. The created solutions modeled by different design candidates and their reconfiguration processes are ranked using a simple heuristic function to identify the top-ranked solutions for detailed optimization. The effectiveness of the newly developed method is demonstrated by the design of a reconfigurable bicycle with three configurations, mountain bicycle, snow bicycle, and electrical bicycle, to reduce the cost to buy three different types of bicycles.

Characteristics in this research aspect are summarized as follows:

- The new pruning method to cut off some nodes in the design tree and reconfiguration process graphs based on heuristic evaluation is effective to reduce the numbers of design candidates and reconfiguration process candidates.
- The new method for initial ranking of the design candidates and reconfiguration process candidates based on heuristic function is effective to further reduce the numbers of design candidates and process candidates by selecting the top-ranked candidates for detailed optimization to improve the computation efficiency.

(3) *Improvement of optimization quality for optimal design of reconfigurable products considering dependency relations among descriptions of design/process candidates and evaluation criteria.*

A new extended analytic network process (EANP) method is developed to determine importance weights of the design/process nodes for improving the quality of the developed optimization method when dependency relations among descriptions of design/process candidates and evaluation criteria are considered. The importance weights of design/process nodes represent their contributions to the optimal solutions. In addition, by selecting the evaluation criteria in the EANP method using similar evaluation measures in the final optimization process, better top-ranked solutions can be identified for detailed optimization. The effectiveness of the EANP method is demonstrated by design of a reconfigurable truck with three configurations, standard truck, recreational vehicle, and canopy truck, considering dependency relationships among descriptions of design/process candidates and evaluation criteria.

Characteristics in this research aspect are summarized as follows:

- The newly developed EANP method can be used to achieve better initial evaluations for design/process nodes which truly reflect the contributions of these nodes to the optimal solutions when the dependency relations among these nodes have to be considered. Since the truly poor nodes are cut off in this new method, better optimal designs of reconfigurable products can be achieved.
- Multiple evaluation criteria are used in the EANP method for the initial evaluations of design/process nodes. By selecting the evaluation criteria in EANP which are similar to the evaluation measures in the final optimization process, the better initial evaluations for the design/process nodes can be achieved. By deleting the nodes with poor measures considering the selected multi-criteria evaluation measures, better optimal designs of reconfigurable products can be obtained.

1.5 Thesis Structure

The rest of this thesis is organized as follows.

In Chapter 2, a literature review for this thesis work is provided. First the researches on architecture design, the configuration design and optimization in development of reconfigurable products are discussed. Then, the heuristic search methods to explore the tree/graph for modeling of product configurations and reconfiguration processes of the reconfigurable products are reviewed for improvement of search efficiency. Finally, the multi-criteria decision-making approaches to evaluate different design/process candidates are explained.

In Chapter 3, an overview of the optimal reconfigurable product design approach considering both product configurations and reconfiguration processes is introduced.

In Chapter 4, a systematic design approach is developed for modeling, evaluation, and optimization of reconfigurable products considering both product configurations and reconfiguration processes.

In Chapter 5, methods to improve optimization efficiency are provided for optimal design of reconfigurable products considering large numbers of design candidates and reconfiguration process candidates.

In Chapter 6, methods to improve optimization quality are explained for optimal design of reconfigurable products considering dependency relations among descriptions of design/process candidates and evaluation criteria.

In Chapter 7, the major contributions of this research are summarized. Potential improvements and challenges to be further investigated are also discussed as future work.

Chapter 2 Literature Review

2.1 Overview

Since this research focuses on three aspects in the optimal design of reconfigurable products considering product configurations and reconfiguration processes, including (1) development of modelling, evaluation, and optimization schemes for optimal design of reconfigurable products considering product configurations and reconfiguration processes, (2) improvement of optimization efficiency for optimal design of reconfigurable products considering large numbers of design candidates and reconfiguration process candidates, and (3) improvement of optimization quality for optimal design method of reconfigurable products considering dependency relations among descriptions of design/process candidates and evaluation criteria, the literature review focuses on the three relevant research areas, including (1) architecture design, configuration design, and optimization of reconfigurable products, (2) search methods to explore the tree/graph which will be used for identification of the optimal product configurations and reconfiguration processes of the reconfigurable product, and (3) multi-criteria decision-making approaches which will be used to evaluate different design/process candidates of reconfigurable products.

In Section 2.2, a brief introduction is first provided to explain the concepts for architecture and configuration design of the reconfigurable product with a structure to change from one configuration to other configurations with the minimum effort. Several research areas related to architecture and configuration design, including modular architecture, integral architecture, configuration design, and configuration optimization, are discussed. In Section 2.3, several methods related to tree and graph based search, including uninformed search and informed search, are summarized. In Section 2.4, several techniques related to multi-criteria decision-making approaches to evaluate alternative solutions, including the analytic hierarchy process (AHP) and the analytic network process (ANP), are introduced.

2.2 Architecture and Configuration Design of Reconfigurable Products

A reconfigurable product has multiple configurations to provide different functions through reconfiguration processes to change between product configurations. The challenges in research on the design of reconfigurable products are summarized in the following aspects (Figure 2.1).

(1) Architecture design:

Architecture is the overall structure of a product. For reconfigurable products, modular architecture is often employed such that different modules can be added to and removed from a product to change its configuration for different functions.

(2) Configuration design and optimization:

For a specific reconfigurable product, different configurations are required for different functions. Various configurations should be designed in a way such that change from one configuration to others requires the minimum time and effort. Optimization is often employed to identify the optimal design based on evaluations to different configurations and reconfiguration processes.

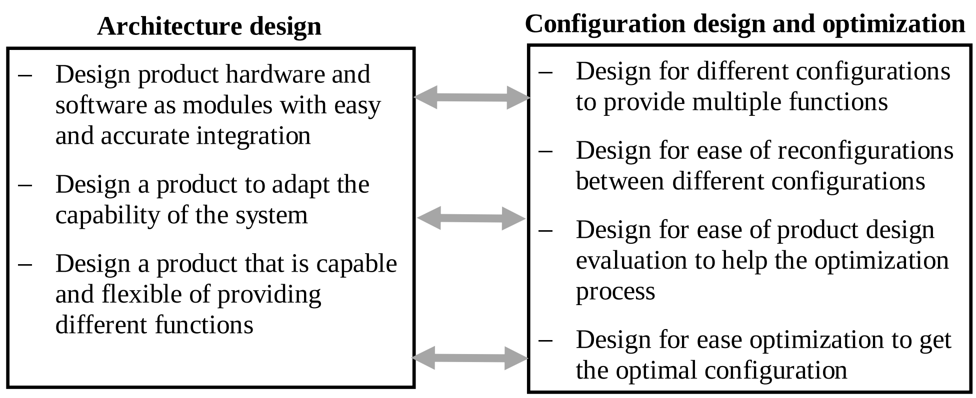


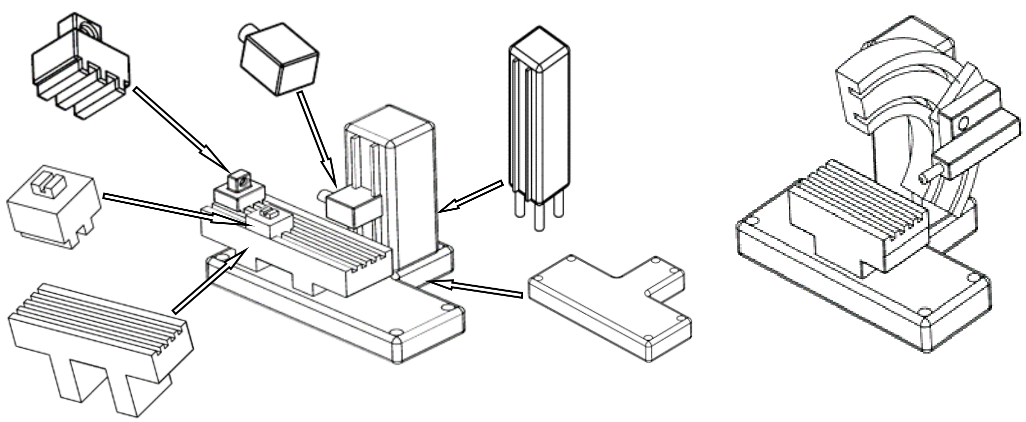
Figure 2.1: Challenges for development of reconfigurable products.

2.2.1 Architecture design

Like a typical mechanical product or system, a reconfigurable product is composed of components and modules that are connected. Different from the conventional product or system, the components and modules of a reconfigurable product need to be assembled and disassembled in the operation stage to satisfy different customer requirements. The architecture of the reconfigurable product is the overall structure about how the components and modules are designed, manufactured, connected and interacted to satisfy the required functions. For reconfigurable products, modular architecture is often employed such that different modules with different functions can be added to and removed from a product platform to change its configuration for different requirements. In addition, open architecture reconfigurable products are also considered to allow the modules developed by third-party vendors to be used to extend functions of the reconfigurable products. Mehrabi, Ulsoy, and Koren (2000) developed a set of the essential characteristics of the reconfigurable products such as:

- *Modularity*. Hardware and software should be designed as modules that can be added to and removed from the product platform.
- *Scalability*. The reconfigurable product should be able to adapt to satisfy the new requirements.
- *Integrability*. Interfaces of modules are designed and used to connect various modules into the same environment through transmission of motion, power, and data.
- *Convertibility*. Change from on configuration to others can be conducted with the minimum effort.

Architectures of reconfigurable products are classified into two categories: modular architecture and integral architecture as shown in Figure 2.2.



(a): A reconfigurable machine tool (RMT) with modular architecture.

(b): A reconfigurable machine tool (RMT) with integral architecture.

Figure 2.2: Reconfigurable products with modular architecture and integral architecture.

2.2.1.1 Modular architecture

In modular architecture design, components are grouped into modules according to the similarities among their design functions and manufacturing processes (Gershenson, Prasad, and Zhang 2003; Chatzopoulos 2014; Bonvoisin et al. 2016). Since the same module can be shared by a number of products, manufacturing effort can be reduced by designing and manufacturing these relatively independent modules separately and creating different products by combining these modules. For a reconfigurable product, different modules are used to form different configurations of the product to deliver different functions.

Zheng and Zhong (2012) summarized the general guideline for design of reconfigurable products with modular architecture.

- *Separation*. To separate components of the product into modules and develop standardized modules that can be shared by different configurations.
- *Unification*. To formulate a group of modules to limit the number of modules to reduce the costs.
- *Connection*. To integrate the modules to the same system through interfaces, especially mechanical interfaces such as joints.
- *Adaptation*. To change from one configuration to the others considering ease of reconfiguration, compatibility among modules, and upgradability with future modules developed using new technologies.
- *Reusability*. To reuse the platform for different configurations and/or up to several cycles within its life to other products after necessary modifications have been made.

According to the differences in interfaces, the reconfigurable products with modular architecture can be classified into two categories: the reconfigurable products with semi-open architecture interfaces and the reconfigurable products with open architecture interfaces. When the specifications of an interface are determined by a group of companies such that the modules developed by the third-party vendors can be used through the standardized interface, this interface is called open architecture interface. When the specifications of an interface are determined by a single company such that only limited modules developed by this company can be used, this interface is called semi-open architecture interface. For example, USB and fire hydrant valve are typical open architecture interfaces, while the connections between a Kenmore vacuum cleaner and its various accessories are typical semi-open architecture interfaces.

(1) Reconfigurable products with semi-open architecture.

A reconfigurable product with semi-open architecture is usually designed with a platform that is connected with different modules for different functions. These modules are usually designed by the same company and can be connected to or disconnected from the platform with the minimum effort.

In this research area, Padayachee and Bright (2012) developed a modular reconfigurable machine (MRM) to work as milling and turning centers. This MRM was composed of 12 hardware modules to form 9 configurations. Aguilar, Roman-Flores, and Huegel (2013) developed a lathe-mill reconfigurable machine tool (RMT) to provide turning and milling functions with flexibility and ease of reconfiguration. This lathe-mill RMT was composed of four main modules: a spindle, a vertical bed, an angle table and a linear module. With these modules, various configurations including a lathe configuration, a mill configuration, a rotated mill configuration, and a heavy mill configuration could be achieved. Majija, Mpofu, and Modungwa (2013) presented the conceptual development of modular machine tools, by selecting and compiling a list of standard modules available commercially off the shelf. The use of these standard modules could reduce costs and time to develop RMT.

As the design of reconfigurable products requires significant effort, computer support systems have been developed to model the knowledge of reconfigurable products and to help engineers for analysis and design of reconfigurable products. Bi and Wang (2009) developed an integrated toolbox with functions of modeling, analysis, simulation, optimization, and monitoring for the design of tripod machines. Graphic-Display and Text-Display Menus were used to display graphical and textual results. Yin and Xie (2011) developed a reconfigurable manufacturing execution system (RMES) for various pipe cutting production operations, milling of intersecting surfaces and automatic welding of pipes. The RMES was composed of a reconfiguration software platform and a reconfiguration controller.

(2) Reconfigurable products with open architecture.

The IEEE Technical Committee of Open Systems defined an open system as (Pritschow et al. 2001):

"An open system provides capabilities that enable properly implemented applications to run on a variety of platforms from multiple vendors, interoperate with other system applications and present a consistent style of interaction with the user."

In this work, an open architecture product (OAP) is considered as a product with open interfaces to allow the third-party vendors to develop new add-on modules and connect these add-on modules through the open interfaces (Zhang, Xue, and Gu 2015). Zhao, Peng, and Gu (2015) stated that a typical OAP consists of three types of functional modules: common platform modules, customized modules and personalized modules as shown in Figure 2.3.

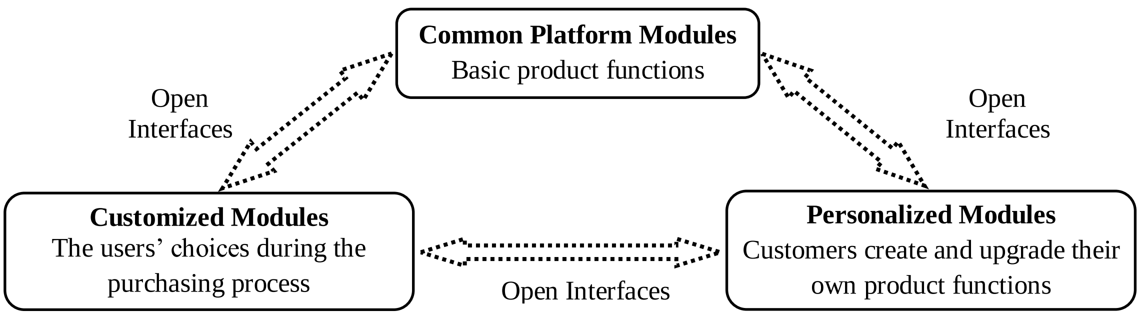


Figure 2.3: Modules of an open architecture product.

Various open-architecture reconfigurable products have been developed in the past decades. Du et al. (2009) developed a reconfigurable CNC (RCNC) machine using modular design approach with a layered system architecture to improve the reusability and upgradability of system software. The modular design approach was used to facilitate the software reuse and satisfy the requirements for the reconfigurable CNC system. The layered system architecture was used to divide the specific requirements and functions of CNC into many layers. Identification of modules plays an important role in the design of OAPs. Many methods have been developed for clustering components of OAPs into modules based on the results achieved in product platform designs and product family designs. Zhao, Peng, and Gu (2015) developed a module-planning method for OAP to identify appropriate product architectures, functional modules, and manufacture solutions to achieve personalized products. This method integrated the extended quality function deployment (QFD) method to decide OAP module types, and the axiomatic design (AD) method to satisfy customer requirements.

Due to uncertainties especially uncertainties of operating conditions, the designed OAP should be less sensitive to these uncertainties including the uncertainties of the specific add-on modules designed at the product development stage and the uncertainties of the add-on modules that will be added in the future. Zhang, Xue, and Gu (2015) developed a robust design approach to identify the optimal design of an open architecture adaptable product whose functional performance measures were the least sensitive to the variations of operating parameters. A statistical method and a worst-case method were developed to calculate the robustness of the OAP.

Koren, Gu, and Shpitalni (2013) introduced a new design paradigm called mass individualization. In this paradigm, individual users were involved in the design process to develop various individualized modules to connect with the OAPs. Innovations in three areas were required for the development of such OAPs: product design, manufacturing and assembly of modules, and cyber-physical systems to facilitate interactions between customers and manufacturers of platforms and modules.

2.2.1.2 Integral architecture

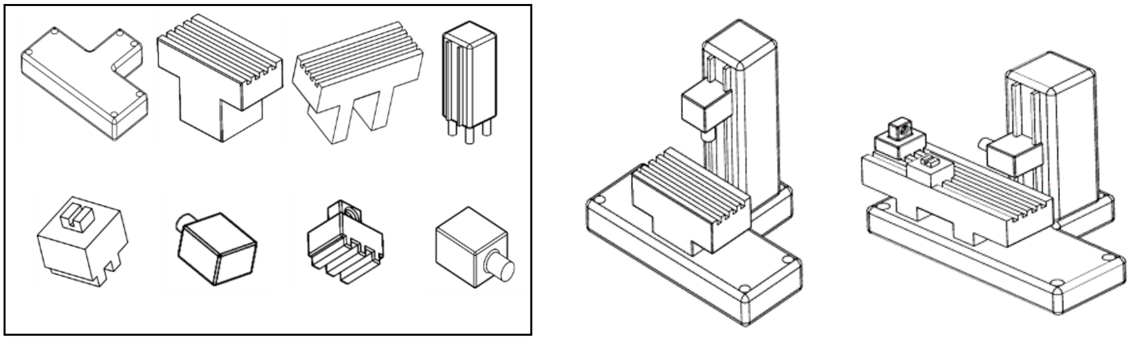
As reconfiguration through adding/removing modules takes time and calibration is required after the product is reconfigured, researches to develop reconfigurable products with integral architecture have been conducted to avoid changes of the modules in the reconfiguration process. In this area, Walczyk, Lakshmikanthan, and Kirk (1998) developed a reconfigurable discrete die (i.e., a matrix of individually actuated pins) for aircraft body parts manufactured by forming of sheet metals or molding of composite materials. Yoon et al. (2014) developed a sheet metal forming process called the flexibly reconfigurable roll forming (FRRF) process. The developed process used reconfigurable rollers whose curvatures were slightly different from each other and rotated by the forming roller motors on both of their ends to attain the shape of the multi-curved sheets.

The arch-type RMTs were developed to achieve customized flexibility to manufacture a family of parts with inclined surfaces. The machine tool has three controlled degrees of freedom (DOF) along the column, the spindle axis, and the table axis directions. The additional passive motion (i.e., the angular reconfiguration motion of the spindle) allows reconfiguration of the spindle’s angular position in five pre-designed locations to machine different inclined surfaces. Dhupia et al. (2007) developed an arch-type RMT to provide customized flexibility in milling and drilling operations for V6 and V8 engine cylinder heads. The frequency response functions (FRFs) based on experiments were employed to study the dynamic behaviors of this arch-type RMT at different reconfiguration positions. Son, Choi, and Park (2010) developed a multi-DOF reconfigurable machine capable of machining complex 3D objects. The machine was composed of two groups of arched rails that were perpendicularly mounted into the main frame using bearings. The rails had a common rotational center to provide 3-DOF rotational motion to change the tool orientation using two actuators at the base.

Moreover, morphing aircrafts were developed to have a wide range of wing configurations to be used for various flight tasks. The ability of the wing surface to change its geometry during flight has been investigated by researchers and designers over the years. An adaptive wing was designed to ensure the operations of the aircraft in multiple flight conditions (Stanewsky 2001). The wing morphing concepts have been classified into three major types (Barbarino et al. 2011): planform alteration (i.e., a wing capable of telescoping), out-of-plane transformation (i.e., twist, dihedral/gull, and spanwise bending), and airfoil adjustment (i.e., changes of camber and thickness). Many studies have targeted the variations of the wing shapes (Weiss 2003; Sofla et al. 2010) to control shock waves during transonic flight conditions, or to control turbulence, or to control flutter (Stanewsky 2001).

2.2.2 Configuration design and optimization

Reconfigurable products are designed to satisfy different requirements of customers using basic process modules and changeable hardware/software modules to form different configurations quickly and reliably. Configuration synthesis of a product is a design process to convert functional requirements of a system into a solution with a set of technologically feasible modules considering interrelations and arrangements of these modules for different operations as shown in Figure 2.4. Many issues need to be considered in configuration design such as to find the best configuration from all feasible configurations, and to check whether the design satisfies design requirements. To achieve the optimal configuration, evaluation to the created design solutions has to be conducted first. Various evaluation measures have been developed to obtain the optimal design with the maximum number of configurations, the minimum cost of product modules, and/or the minimum reconfiguration effort to change from one configuration to the others.



(a): Machine modules.

(b): Milling/drilling configuration.

(c): Turning configuration

Figure 2.4: Two configurations of a reconfigurable product.

Optimization is a process of search for the best solution when operating within a set of constraints. There are two types of optimization methods, exhaustive and heuristic methods. In exhaustive methods, since all solutions must be tested to identify the optimal solution, the exhaustive methods are guaranteeing to find the optimal solution. However, only a percentage of solutions are examined in the heuristic methods to determine the optimal solution. For example, if we tested 30% from all solutions using the exhaustive method, the probability to find the optimal solution is 30%. However, if we tested 30% from all solutions using the heuristic method, the possibility to find the optimal solution could be 90%.

Based on the literature review, the requirements for design and optimization of reconfigurable product configurations are summarized as follows:

- In module design approach, the number of modules should be minimized while the number of configurations should be maximized.
- Interrelations among modules and arrangements of these modules should be investigated for different functions through different configurations of the product.
- Software kits are expected to design, analyze, and optimize product configurations.
- A metrics needs to be developed to evaluate the design of the reconfigurable product.
- Multi-objective optimization tools are usually required to identify the optimal configurations of the reconfigurable products considering different evaluation aspects.

2.2.2.1 Configuration design

Many methods have been developed for the systematic design of reconfigurable product configurations. Spicer et al. (2002) introduced principles for configuration design of machining systems with changeable configurations. They also developed an upgradable multi-spindle reconfigurable machine as a cost-effective alternative considering system scalability. Ersal, Stein, and Louca (2004) developed a modular modelling approach for the design of RMTs. Modelling tasks in this method were conducted in two steps: development of component models and assembly of the components to build the configuration model. Chen, Xi, and Macwan (2005) developed a systematic method, called feature-based selection method, for identifying a minimum yet sufficient set of modules to machine a given set of machining features. Moon (2006) developed a method to design RMT configurations by using graph theory to model different configurations. Mori and Fujishima (2009) developed a CNC machining center with different configurations for horizontal machining, vertical machining and turning for the automotive industry. Zhang et al. (2014) developed a robust method for design of adaptable products considering changes of product configurations using an AND/OR tree to model the relations among functional requirements, product configurations, and configurations parameters.

Computer systems have been developed for the support of configuration design of reconfigurable products. In this area, Lorenzer et al. (2007) developed a software tool for obtaining reliable physical information on the behaviors of machine tool structure variants. In the software tool called Axis Construction Kit, both the physical properties and geometric properties of the machine components were considered for machine evaluation to get the optimal RMT configuration. Marin et al. (2015) developed a method for machine configuration analysis based on virtual interactive prototyping using Visual C and Lite C programming. Since reconfigurability of the machine depends on both the machine and its operator, human-machine interaction also plays a significant role in configuration design of a RMT. Yin et al. (2012) developed an imaginal thinking based human machine interactive design methodology to achieve highly efficient processing of information for enterprises. By imitating the imaginal thinking style of human, the computer obtained human knowledge to get all possible configurations of RMT. This methodology shortened the lead-time for configuration design of RMT, and enabled the RMT to respond more quickly and accurately to both current and future market changes.

2.2.2.2 Configuration optimization

When various solutions are available for configuration design, optimization is then conducted to select the best one based on the evaluation to these feasible solutions. Spicer, Yip-Hop, and Koren (2005) developed a mathematical approach to determine the optimal number of modules to be included in a scalable machine. Various evaluation measures, including machine investment cost, production rate, and equipment availability, were used for identification of the optimal number of modules for the machine. Riba et al. (2006) developed a metrics to evaluate the RMTs using reconfigurability rates of machines and cutting tools. The machine's reconfigurability rate was used to evaluate the degree of reconfigurability of a specific machine tool. Goyal, Jain, and Jain (2013) developed metrics to evaluate the responsiveness of RMTs. A RMT with higher evaluation measures in operational capability and machine reconfigurability requires shorter response time to handle the fluctuating demands and lower associated reconfiguration costs.

Configuration optimization for RMT is often modeled as multi-objective optimization (MOO) problems and/or multi-level optimization (MLO) problems due to the presence of many and/or conflicting objectives. Various optimization methods and tools have been developed for identifying the optimal reconfigurable products as summarized in Table 2.1.

Table 2.1: Optimization objective functions and methodologies.

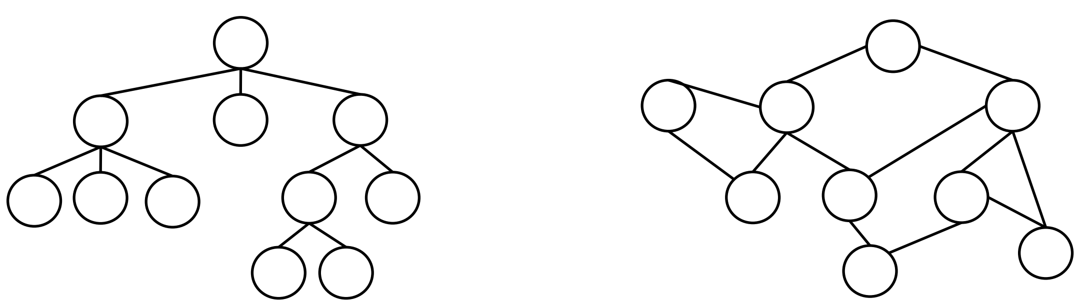
Reference	**Optimization objective functions**	**Methodology**
Liu and Liang (2008)	Machine configurability, total cost (purchase, module exchange, and storage), and process accuracy by estimating the tool position errors.	A Modified Fuzzy Chebyshev Programming (MFCP) method to achieve the preferred compromise among design objectives considering the relative importance of objective functions. Particles swarm optimization algorithm (PSOA) to improve optimization efficiency.
Abdi (2009)	Manufacturing configurability (capacity and functionality), total cost (operating cost, overhead, and capital), quality (convenience of use, reliability, accuracy, and compatibility), and performance (efficiency, risk, and safety).	Fuzzy analytical hierarchical process (FAHP) model with five hierarchical levels to set strategic goals, set objectives and criteria in a hierarchical order, identify available machine/configuration options, trade-off the objectives and criteria with fuzzy/crisp values, and select the most preferred configuration.
Mpofu and Tlale (2012)	Minimum number of degrees of axes, minimum reconfigurations, and minimum dissimilarity between old and new configurations.	A method to create dynamic optimal configurations of machine structures through multi-level fuzzy decisions considering part geometry.
Zhang and Bi (2011)	Maximum rotational motion of the parallel kinematic machine (PKM) end-effector.	Analysis of direct and inverse kinematics. Optimization of structural parameters based on a genetic algorithm (GA) to get the optimal configuration.

Table 2.1: Optimization objective functions and methodologies (continued).

Reference	Optimization objective functions	Methodology
Xie, Li, and Xue (2012)	Product quality, and production cost.	A cooperative optimization model for identifying the optimal RMT configuration based on production process plans using genetic algorithm (GA).
Bensmaine, Dahane, and Benyoucef (2013)	Total cost (production, tool changing and operation cost), and total completion time (processing time of a particular operation, and tool change time).	A method to provide a set of non-dominated solutions by using Non-dominating Sorting Genetic Algorithm (NSGA-II) based on the process plan that gave more flexibility to the decision maker.
Goyal, Jain, and Jain (2012)	Machine reconfigurability, operational capability (number of operations that can be performed by a machine), and cost.	The Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) approach to rank the non-dominated solutions of NSGA-II and to obtain the attribute weights through the Shanon entropy weight theory.
Azulay, Mills, and Benhabib (2014)	Various optimization objective functions in the five tiers.	A multi-tier optimization approach with five tiers to achieve reconfigurability via redundancy without the need to assemble/disassemble modules.

2.3 Tree and Graph Based Search

A tree is modeled by nodes in a hierarchical data structure as shown in Figure 2.5(a). Each node may have one or multiple descendant nodes (i.e., child-nodes). Except for the root node, a node in the tree has only one parent node. The root node is linked with other nodes by paths. A graph is also composed of nodes as shown in Figure 2.5(b). Different from the tree, any two nodes in the graph can be connected by a path and the arcs may have directions (West 2001).



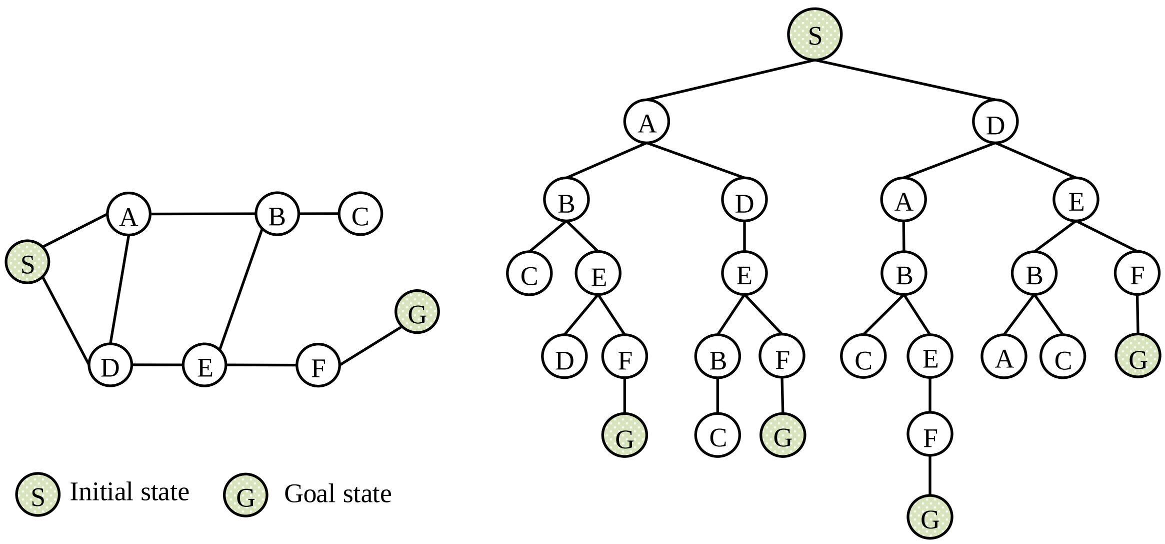
(a): A tree data structure. **(b):** A graph data structure.

Figure 2.5: Tree and graph data structures.

A search tree can be created with all available solutions to solve a problem. For example, to find a path between the initial state node A and the goal state node G from the graph shown in Figure 2.6(a), a complete search tree as shown in Figure 2.6(b) is generated by traversing the graph in Figure 2.6(a) to create all solutions from the initial state to the goal states. Identification of a path depends on the traversal strategy (i.e., search method). The initial state is selected as the root node of the tree, and the goal states are selected as the leaf nodes of the tree. In this section, two types of search methods, uninformed search methods and informed search methods, are reviewed.

2.3.1 Uninformed search

In uniformed search, the heuristic value (also called cost) from the current state to the goal state is not available in the search process. Uninformed search methods, including breadth-first search (BFS) and depth-first search (DFS), will be reviewed in this section.



(a): A graph search problem. **(b):** Traversing a graph as a search tree.

Figure 2.6: Tree-based search.

2.3.1.1 Breadth-first search (BFS).

Breadth-first search (BFS) is an uninformed search method in which the root node is explored first, and then all successor nodes are explored as shown in Figure 2.7. The BFS method explores all nodes at a given level before exploring any nodes at the next level. For example, when the breadth-first search is used for the search tree shown in Figure 2.6(b), the search starts at node S and explores A and D nodes in the next level. Then, BFS explores other nodes in the following levels until reaching the goal node (i.e., node G) as shown in Figure 2.7.

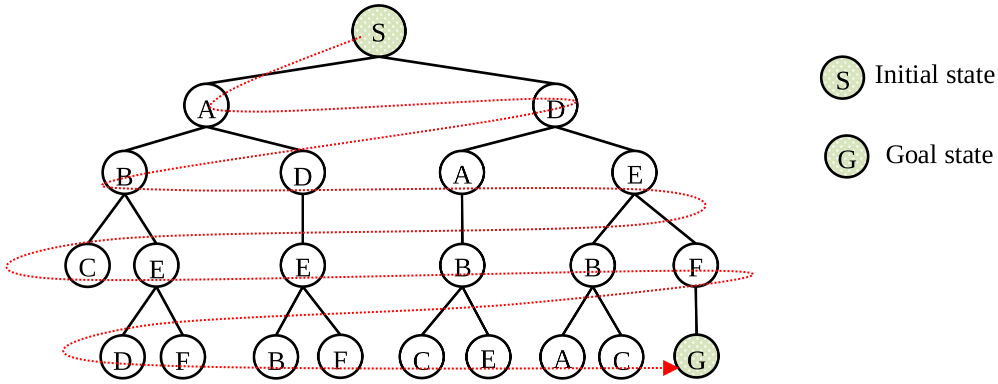


Figure 2.7: Order in which nodes are explored using BFS.

2.3.1.2 Depth-first search (DFS).

Depth-first search (DFS) is an uninformed search method which starts at the root node and continues down a particular path (branch) by selecting a child node at the deepest level of the search tree. The order of exploring nodes using the DFS method is down path and from left to right. For the search tree shown in Figure 2.6(b), the DFS method starts by selecting the root node and then explores nodes in the following levels until reaching the goal node as shown in Figure 2.8.

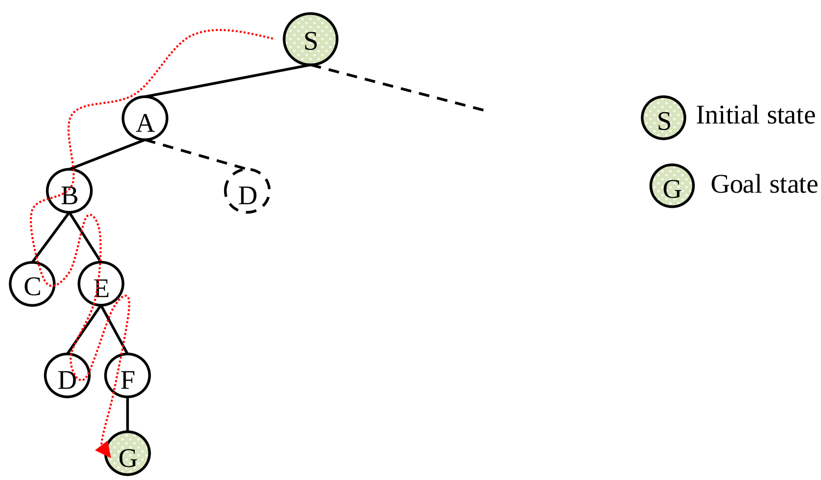


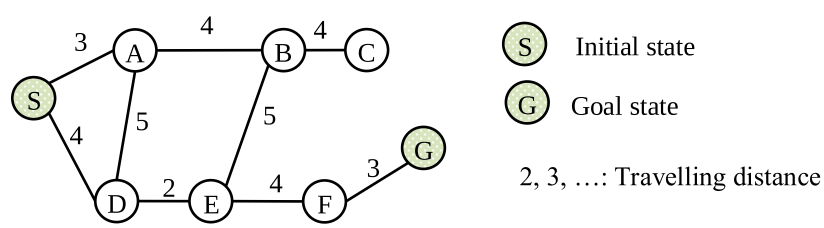
Figure 2.8: Order in which nodes are explored using DFS.

2.3.2 Informed search

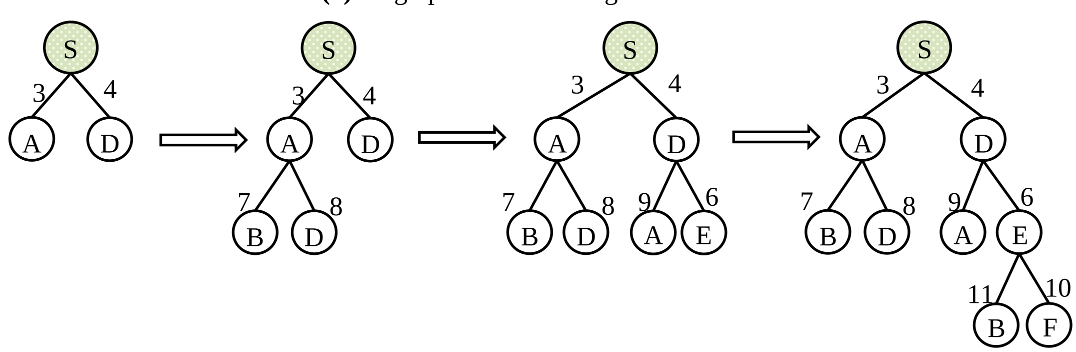
The informed search, also called heuristic search, solves the search problem based on the evaluation to the nodes using a heuristic function. The informed search makes an intelligent choice of nodes for exploration (Edelkamp and Schroedl 2011). Heuristic search evaluates the likelihood that a given node is on the solution path by using a heuristic function. The heuristic function uses the available information to rank different nodes at each branching step to decide which branch to follow. The partial solution achieved so far may not be the best of all solutions to this problem. Some of the informed search methods to be reviewed in this section include the greedy best-first search, A* search, and beam search.

2.3.2.1 Greedy best-first search

Greedy best-first-search is an informed search method which explores the node that appears to be closest to the goal. So, the greedy best-first-search ranks nodes of the search tree using a heuristic function which estimates how a node is close to the goal state. For example, when a greedy best-first search is used to expand nodes of the search tree shown in Figure 2.9(a), the search starts by node S, then expands the node with the shortest travelling distance until reaching the goal node as shown in Figure 2.9(b). The created solution path using the greedy best-first search is S-D-E-F-G with a total travelling distance of 13. The greedy best-first search is useful and may yield a locally optimal solution. A variety of greedy best-first-search applications can be found in (Lawler 1976).



(a): A graph with travelling distances.



(b): Node expansion using greedy best-first search.

Figure 2.9: Greedy best-first search.

2.3.2.2 A search.*

A* (A star) search is an efficient informed search method used for finding a path between two nodes in the tree. A* search is commonly used for solving pathfinding problems in applications such as games (Hart, Nilsson, and Raphael 1968). Like the greedy best-first search, a heuristic function is used to guide the order in which nodes are explored. Unlike greedy best-first search, the heuristic function used in A* search considers both the cost from the start state to the current state and the cost from the current state to the goal state.

2.3.2.3 Beam search.

Beam search method is a heuristic method in which the promising nodes of the search tree are explored in a limited set (Edelkamp and Schroedl 2011). Beam search is an optimization of best-first search to reduce the search time through exploring only some good partial solutions as candidates (Wilt, Thayer, and Ruml 2010). At each level of the search tree, beam search ranks all successor nodes using a heuristic function. Then, the beam search explores a predetermined number of top-ranked nodes at each level, and this predetermined number is called a beam width (i.e., threshold value, α). The remaining nodes are cut off from the search tree. When the beam width is large, the number of selected nodes is large, and the number of the pruned nodes is small.

As shown in Figure 2.10, beam search explores only the top-ranked nodes and prunes the remaining nodes with poor ranking. In Figure 2.10, the beam width is five (i.e., the maximum number of explored nodes at each level is 5). For the first level, there exist only 3 nodes, so the pruning is not conducted. For the second level, there exist 6 nodes, so, only five nodes from the top-ranked nodes are selected and one node with its sub-nodes are pruned.

The beam search is used to solve various problems such as to solve the classic job shop scheduling problem to achieve the minimum make-span and mean tardiness (Sabuncuoglu and Bayiz (1999). Blum (2005) integrated the solution construction mechanism of ant colony optimization algorithms and the beam search for application to open shop scheduling.

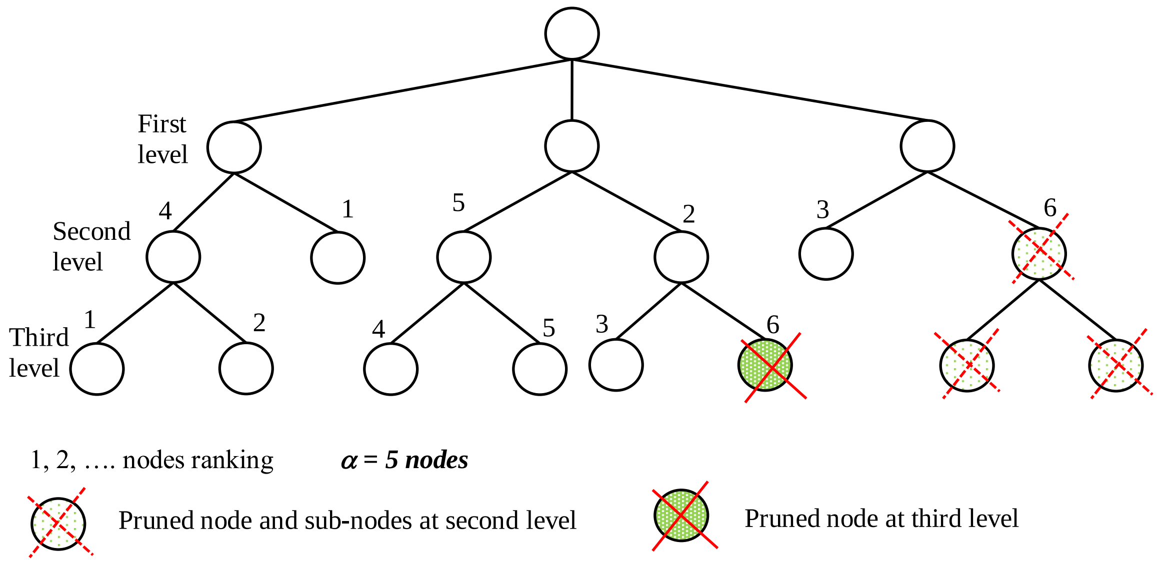


Figure 2.10: Beam search for the tree.

2.4 Multi-Criteria Decision-Making Approaches to Rank Design/Process Candidates

Multi-criteria decision-making (MCDM) is a process to structure and solve decision and planning problems involving multiple criteria. In recent years, several MCDM techniques have been developed for ranking various alternatives. Typical MCDM methods including analytic hierarchy process (AHP) (Saaty 1990), technique for order of preference by similarity to ideal solution (TOPSIS) (Hwang and Yoon 1981), and analytic network process (ANP) (Saaty 1999) are used to evaluate different alternatives. Since ANP has been used in this research and ANP was developed based on AHP, both AHP and ANP are reviewed in this section.

2.4.1 Analytic hierarchy process (AHP)

The analytic hierarchy process (AHP) is a multi-criteria decision-making tool used to develop importance weights for different alternatives considering both qualitative and quantitative criteria. The AHP method uses a ratio scale developed by Saaty (1977) to rate the decision maker preferences. AHP method assumes independent relationships between all alternatives and criteria. In AHP, different evaluation criteria are modeled in a hierarchical structure as shown in Figure 2.11. The relative importance factors of these sub-criteria are defined by weights.

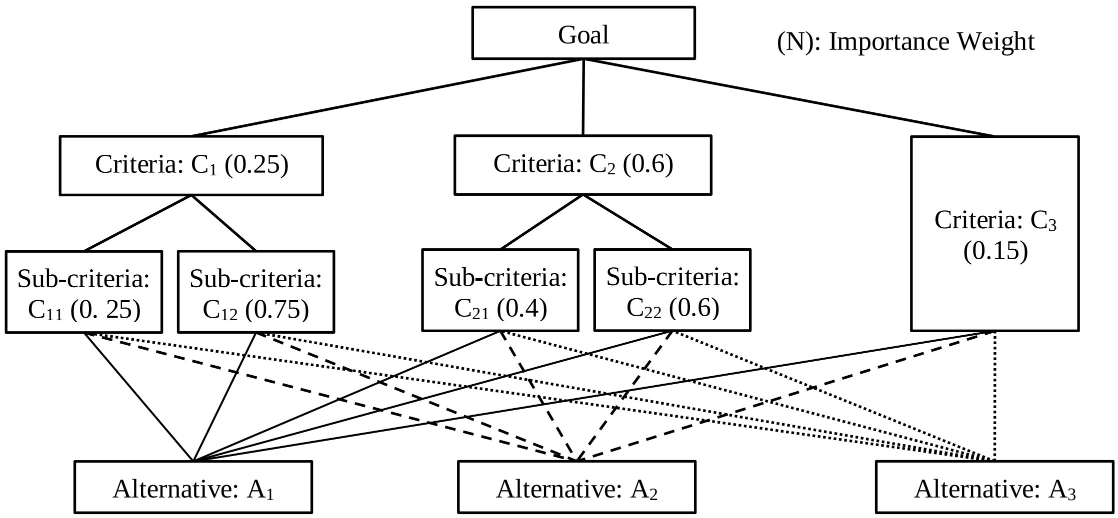


Figure 2.11: Hierarchical structure of analytic hierarchy process (AHP).

The first step in AHP is to find weights of evaluation criteria in different groups through pairwise comparison using the judgment scales shown in Table 2.2. The sum of weights of all sub-criteria in each cluster is 1. The relative weights of criteria are obtained through pairwise comparison (Saaty 1977). The second step in AHP is to identify weights of all alternatives considering one evaluation criterion through pairwise comparison, and the sum of these weights is 1. Finally, the preference of an alternative, called overall priority, is calculated based on the weights of this alternative considering all the evaluation criteria and weights of these evaluation criteria in the hierarchical data structure. All the alternatives are ranked based on their overall priorities as shown in Table 2.3. The sum of overall priorities for all alternatives is 1.

Table 2.2: Judgement scale used for AHP and ANP.

Scale	Importance
1	equal important
3	moderate important
5	strong or essential important
7	very strong or demonstrated important
9	extreme important
2,4,6,8	intermediate values

Table 2.3: Priorities using AHP.

Criteria / Alternatives	C_1 (0.25)		C_2 (0.60)		C_3 (0.15)	AHP value	Ranking
	C_{11} (0.25)	C_{12} (0.75)	C_{21} (0.40)	C_{22} (0.60)			
A_1	0.395	0.320	0.250	0.290	0.320	0.297	**3**
A_2	0.250	0.325	0.295	0.320	0.320	0.311	**2**
A_3	0.355	0.355	0.455	0.390	0.360	0.392	**1**

After each pairwise comparison, a consistency check for each comparison is conducted. The consistency ratio (*CR*) is calculated using the consistency index (*CI*) and the random index (*RI*) for n×n decision matrix. It is recommended for the consistency ratio to be less than or equal to 0.10 which indicates sufficient consistency for a decision. The *CR* is calculated by:

$$CR = CI/RI \tag{2.1}$$

$$CI=(\lambda_{max}-n)/(n-1) \tag{2.2}$$

where λ_{max} is the maximum eigenvalue of the n×n decision matrix, *RI* is the random index *(*Saaty *1981)* as shown in Table 2.4.

Table 2.4: Values of random indices used for AHP and ANP.

n	**3**	**4**	**5**	**6**	**7**	**8**	**9**
RI	0.58	0.90	1.12	1.24	1.32	1.41	1.45

AHP has been widely used for solving various multi-criteria decision-making problems (Ishizaka and Labib 2011; Ic, Yurdakul, and Eraslan 2012). AHP has also been integrated with other methods, such as meta-heuristics, strengths, weaknesses opportunities and threats (SWOT) analysis, mathematical programming, quality function deployment (QFD), and data envelopment analysis (DEA), to solve various problems (Ho 2008; Ho, Xu, and Dey 2010). Advantages and limitations of AHP were summarized by Ishizaka and Labib (2009).

2.4.2 Analytic network process (ANP)

When dependency relations among alternatives and criteria have to be considered, the AHP would provide inaccurate weights. So, Saaty and Takizawa (1986) developed an analytic network process (ANP) method to overcome this problem by considering dependency relations among alternatives and criteria. Thus, ANP is a generalization of the AHP and both AHP and ANP methods are useful tools for multi-criteria decision-making. The ANP was developed to model the dependency relations among evaluation criteria within and between different clusters (Saaty 1999). In ANP, several clusters of evaluation criteria and one cluster of alternatives are modeled by nodes of the network as shown in Figure 2.12. Inner dependency relations (i.e., one element in the cluster influences other elements in the same cluster) and outer dependency relations (i.e., one element in the cluster influences other elements in other clusters) are defined as arcs of the network. Various applications of ANP were summarized by Saaty and Vargas (2006).

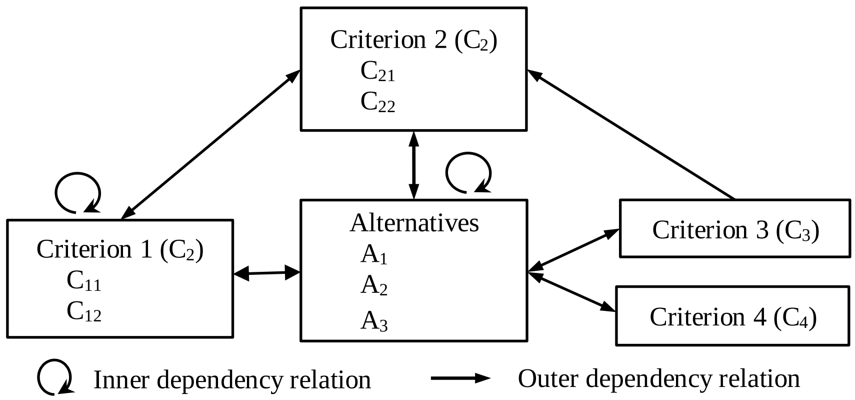


Figure 2.12: Analytic network process (ANP).

The dependency relations among clusters are defined by weights in a cluster matrix $\mathbf{M_c}$ as shown in Table 2.5. For each influencing cluster, its total influencing weight to other clusters is 1. Then different alternatives are compared considering each of the evaluation criteria through pairwise compassion. The evaluation results are described by priorities between 0 and 1, and the sum of the priorities for all alternatives considering one evaluation criterion is 1. In addition, the influences among alternatives and evaluation criteria in the clusters given in Figure 2.12, representing the inner and outer dependency relations, are also defined by weights between 0 and 1. The sum of the dependency weights for alternatives/criteria in a cluster against one alternative/criterion is 1.

Table 2.5: Dependency relations among clusters ($\mathbf{M_c}$) given in Figure 2.12.

Influenced cluster	**Influencing clusters**				
	A	C_1	C_2	C_3	C_4
A	0.188	0.327	0.500	0.500	1
C_1	0.373	0.413	0.500	0	0
C_2	0.213	0.260	0	0.500	0
C_3	0.133	0	0	0	0
C_4	0.093	0	0	0	0

Then different alternatives in different clusters are compared considering each of the evaluation criteria through pairwise comparison as shown in the unweighted matrix $\mathbf{M_u}$ in Table 2.6(a). The weights in the unweighted matrix $\mathbf{M_u}$ are usually assigned by users for specific problems. The evaluation results are described by priorities between 0 and 1, and the sum of the priorities for all alternative in a cluster considering one evaluation criterion is 1. In addition, the influences among alternatives and evaluation criteria in the clusters given in Figure 2.12, representing the inner and outer dependency relations, are also defined by weights between 0 and 1 in the unweighted matrix $\mathbf{M_u}$ in Table 2.6(a). The sum of the dependency weights for alternatives/criteria in a cluster considering one alternative/criterion is 1. The unweighted supermatrix $\mathbf{M_u}$ is then converted into a weighted supermatrix $\mathbf{M_w}$ shown in Table 2.6(b) by multiplying each weight in Table 2.6(a) by the corresponding weight for the cluster defined in Table 2.5. Weights in each column of $\mathbf{M_w}$ are normalized such that the sum of weights is 1. The weighted supermatrix $\mathbf{M_w}$ is further converted into the limited supermatrix $\mathbf{M_l}$ as shown in Table 2.6(c) by raising its limiting power k through $(\mathbf{M_w})^k$ using the method developed by Saaty (1999). In the limited supermatrix, the weights in each row are converged into global priorities with the same value. Then, final normalized weights for the alternative cluster in Figure 2.12 are calculated as shown in Table 2.7.

Table 2.6: Evaluation and ranking of different alternatives in different clusters by ANP.

(a): Unweighted supermatrix $\mathbf{M_u}$.

		A			C_1		C_2		C_3	C_4
		A_1	A_2	A_3	C_{12}	C_{12}	C_{21}	C_{22}		
A	A_1	0.571	0	0	0.540	0.600	0.500	0.625	0.594	0.550
	A_2	0.286	0	0	0.297	0.200	0.250	0.238	0.249	0.240
	A_3	0.143	0	0	0.163	0.200	0.250	0.137	0.157	0.210
C_1	C_{11}	0.670	0	0.500	0.670	0.500	0.750	0	0	0
	C_{12}	0.330	0	0.500	0.330	0.500	0.250	0	0	0
C_2	C_{21}	0.500	0	1	1	1	0	0	0.750	0
	C_{22}	0.500	0	0	0	0	0	0	0.250	0
C_3		0.028	1	0	0	0	0	0	0	0
C_4		0.019	1	0	0	0	0	0	0	0

(b): Weighted supermatrix $\mathbf{M_w}$.

		A			C_1		C_2		C_3	C_4
		A_1	A_2	A_3	C_{12}	C_{12}	C_{21}	C_{22}		
A	A_1	0.108	0	0	0.177	0.196	0.250	0.625	0.297	0.550
	A_2	0.054	0	0	0.097	0.066	0.125	0.238	0.125	0.240
	A_3	0.027	0	0	0.054	0.066	0.125	0.137	0.079	0.210
C_1	C_{11}	0.249	0	0.318	0.275	0.206	0.375	0	0	0
	C_{12}	0.124	0	0.318	0.138	0.206	0.125	0	0	0
C_2	C_{21}	0.107	0	0.363	0.260	0.260	0	0	0.375	0
	C_{22}	0.107	0	0	0	0	0	0	0.125	0
C_3		0.028	0.028	0.133	0	0	0	0	0	0
C_4		0.019	0.019	0.093	0	0	0	0	0	0

Table 2.6: Evaluation and ranking of different alternatives in different clusters by ANP (continued).

(c): Limited supermatrix $\mathbf{M_l}$.

		A			C1		C2		C3	C4
		A_1	A_2	A_3	C_{12}	C_{12}	C_{21}	C_{22}	C_3	C_4
A	A_1	0.191	0.191	0.191	0.191	0.191	0.191	0.191	0.191	0.191
	A_2	0.089	0.089	0.089	0.089	0.089	0.089	0.089	0.089	0.089
	A_3	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066
C_1	C_{11}	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259
	C_{12}	0.145	0.145	0.145	0.145	0.145	0.145	0.145	0.145	0.145
C_2	C_{21}	0.176	0.176	0.176	0.176	0.176	0.176	0.176	0.176	0.176
	C_{22}	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.026
C_3		0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028
C_4		0.019	0.019	0.019	0.019	0.019	0.019	0.019	0.019	0.019

Table 2.7: Identified weights for all alternatives.

Parent node	Sub-nodes	ANP weights	Normalized weights
A	A_1	0.191	0.552
	A_2	0.089	0.257
	A_3	0.066	0.191

Chapter 3 Framework of the Systematic Approach for Optimal Design of Reconfigurable Products Considering Product Configurations and Reconfiguration Processes

3.1 Introduction

A systematic approach for optimal design of reconfigurable products considering product configurations and reconfiguration processes has been developed in this research. In this approach, first modeling, evaluation, and optimization schemes have been developed to identify the optimal design solution of reconfigurable product considering product configurations and reconfiguration processes. A modelling scheme is developed to describe different product configurations, reconfiguration processes, and design/process parameters. An evaluation and optimization scheme is introduced to identify the optimal design solution of reconfigurable product based on the evaluations to product configurations and reconfiguration processes. Second, a new method has been developed to improve the efficiency of the optimization method for optimal design of reconfigurable products when large numbers of design candidates and reconfiguration process candidates are considered. Finally, a new method has been developed to improve the quality of the optimization method for optimal design of reconfigurable products when dependency relations among design/process descriptions and evaluation criteria are considered. In the rest of this chapter, the framework of this approach will be introduced, and the three aspects of this approach will be explained

3.2 An Approach to Identify the Optimal Configurations and Reconfiguration Processes for Design of Reconfigurable Products

In this approach, a generic design AND-OR tree is used to model different design solution candidates, their product configurations and parameters of these configurations. A specific design solution is created from the generic design AND-OR tree through tree-based search (West 2001; Edelkamp and Schroedl 2011) and modeled by different product configurations. For a reconfiguration process to change between two product configurations, a generic process AND-OR graph is used to model reconfiguration operation candidates, sequential constraints among operations and operation parameters. A graph-based search is used to generate all feasible reconfiguration process candidates from the generic process AND-OR graph. The optimal design is identified by multi-level and multi-objective optimization. The new optimal reconfigurable product design approach with three steps is shown in Figure 3.1.

3.2.1 Modeling of design candidates and their configurations

In this step, different design solution candidates, S_i (i = 1, 2, …, p), are first modeled based on design requirements. Each of these design solution candidates is modeled by product configurations C_{ij} (i = 1, 2, …, p; j = 1, 2, …, q), and each product configuration C_{ij} is described by design parameters, $d_{ij,1}, d_{ij,2}, \dots, d_{ij,u_{ij}}$, where p is the number of design solutions, q is the number of product configurations, and u_{ij} is the number of design parameters for the C_{ij}.

In this work, a generic design AND-OR tree with design nodes and their parameters (Xue, Cheing, and Gu 2006; Hong et al. 2008) is used to model all possible design solution candidates, product configurations of these candidates, and configuration parameters. The relations among design nodes are classified into three categories as shown in Table 3.1(a): AND relation (AND), OR relation in design (OR-D), and OR relation in configuration (OR-C). Tree-based search (Edelkamp and Schroedl 2011) is conducted to create specific design candidates from the generic design AND-OR tree. For each specific design candidate, a tree-based search is further conducted to create multiple product configurations.

Table 3.1: Various types of relations in the AND-OR tree/graph.

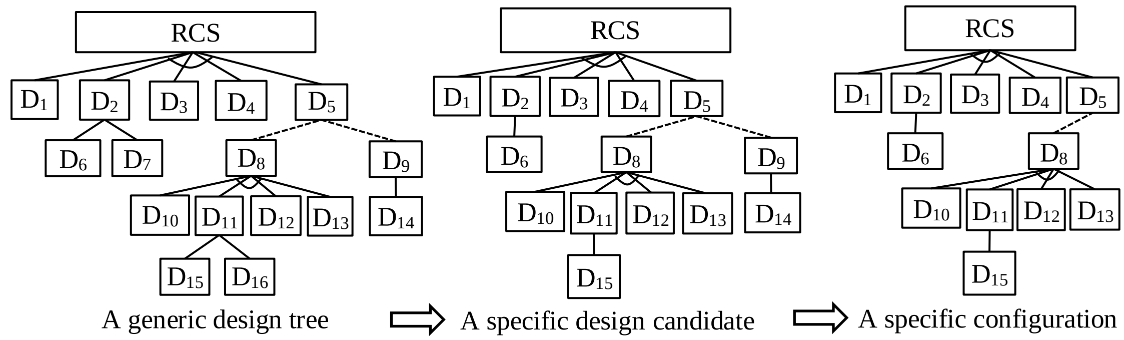
(a): Various types of relations in the design AND-OR tree.

Relation type	Modeling of generic design tree	Generation of a specific design tree
AND relation (AND)	A parent design node is supported by all the sub-nodes.	All sub-nodes should be selected.
OR relation in design (OR-D)	A parent design node is supported by alternative sub-nodes.	Only one sub-node is selected.
OR relation in configuration (OR-C)	A parent design node is supported by sub-nodes for different product configurations.	All sub-nodes should be selected.

(b): Various types of relations in the reconfiguration process AND-OR graph.

Relation type	Modeling of generic process graph	Generation of a specific process graph
AND relation (AND)	A parent operation node is supported by all the sub-nodes.	All sub-nodes should be selected.
OR relation in operation (OR-O)	A parent operation node is supported by alternative sub-nodes.	Only one sub-node is selected.
OR relation in design condition (OR-DC)	A parent operation node is supported by alternative sub-nodes for different design candidates.	Only one sub-node is selected according to the selected design candidate.

Modeling of design candidates and their configurations



Node	Description	Node	Description
D_1	Pipe and base	D_9	Fan
D_2	Blade	D_{10}	Grill with copper tubing
D_3	Motor	D_{11}	Cooler box
D_4	Control panel	D_{12}	Water pump
D_5	Functional system	D_{13}	Clear tubing
D_6	Metal blade	D_{14}	Standard grill
D_7	Plastic blade	D_{15}	Thermal cooler bag
D_8	Air conditioner	D_{16}	Foam cooler

RCS: Reconfigurable Cooling System

D_i: *i-th* design node

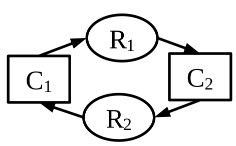
Design node

AND relation (AND)

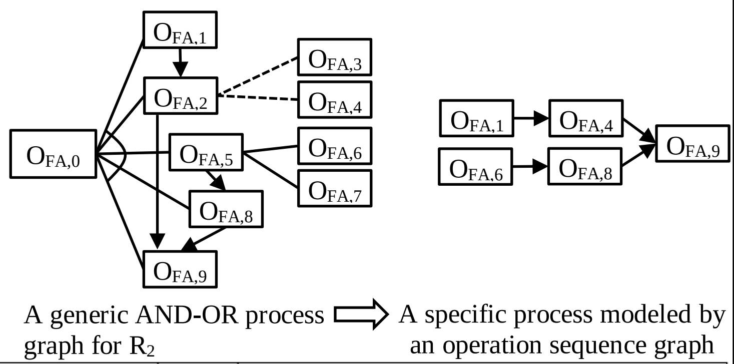
OR relation in design (OR-D)

OR relation in configuration (OR-C)

Modeling of reconfiguration processes



C_1: Air conditioner configuration
C_2: Fan configuration
R_1: Reconfiguration process from C_1 to C_2
R_2: Reconfiguration process from C_2 to C_1



Node	Description	Node	Description
$O_{FA,0}$	Reconfiguration from fan to air conditioner	$O_{FA,5}$	Set the 2 clear tubes through the cooler box
$O_{FA,1}$	Set the front grill that has a copper tube	$O_{FA,6}$	IF it is foam, THEN drill two holes
$O_{FA,2}$	Connect the 2 clear tubes to the cooper tube	$O_{FA,7}$	IF it is bag, THEN put them at the bag top
$O_{FA,3}$	Connect using nylon tie	$O_{FA,8}$	Connect one clear tube to the water pump
$O_{FA,4}$	Connect using metal ring	$O_{FA,9}$	Put water/ice in the cooler box, then close it

$O_{FA,i}$: i-th reconfiguration operation to change from fan configuration (F) to air conditioner configuration (A)

Operation node AND Relation OR relation in design condition (OR-DC)

OR relation in operation (OR-O) Sequential constraint

Identification of the optimal design

Figure 3.1: Optimal reconfigurable product design approach.

3.2.2 Modeling of reconfiguration processes

For any two product configurations C_{ij} and C_{ik} (i = 1, 2, …, p; j= 1, 2, …, q; k = 1, 2, …, q; $j \neq k$) of the i-th design solution candidate, the reconfiguration process to change from C_{ij} to C_{ik} is modeled by R_{ijk}. Each reconfiguration process R_{ijk} is described by process parameters, $p_{ijk,1}, p_{ijk,2}, \ldots, p_{ijk,u_{ijk}}$ where u_{ijk} is the number of process parameters for the R_{ijk}.

In this work, a generic process AND-OR graph with operation nodes is used to model all possible reconfiguration process candidates, sequential constraints between operation nodes, and parameters of operation nodes. The relations among sub-nodes of a super-node are classified into three categories as shown in Table 3.1(b): AND relation (i.e., composition relation), OR relation in operation (OR-O), and OR relation in design condition (OR-DC) through IF-THEN descriptions. Creation of specific reconfiguration processes from the generic process AND-OR graph is conducted through graph-based search.

3.2.3 Identification of the optimal design

Since each specific solution candidate can be described by multiple product configurations, reconfiguration processes, and parameters of product configurations and reconfiguration processes, the optimal design, S_i^*, is identified from all the feasible design candidates based on evaluations to the product configurations and reconfiguration processes of these design candidates. If a product configuration is evaluated by an evaluation index $I_C(C_{ij})$ and a reconfiguration process to change between two product configurations is evaluated by another evaluation index $I_R(R_{ijk})$, the optimal design, S_i^*, can be identified through optimization.

$$\begin{aligned}
&\max_{w.r.t. S_i} \left(\sum_{j=1}^{q} I_C(C_{ij}) + \sum_{j=1, j\neq k}^{q} \sum_{k=1}^{q} I_R(R_{ijk}) \right),\ i = 1,2,\ldots,p \\
&d_{ij,u}^{(L)} \leq d_{ij,u} \leq d_{ij,u}^{(U)} \\
&p_{ijk,u}^{(L)} \leq p_{ijk,u} \leq p_{ijk,u}^{(U)}
\end{aligned} \tag{3.1}$$

where:

w.r.t. is an abbreviation of "with respect to",

$\sum_{j=1}^{q} I_C(C_{ij})$ is the sum of evaluation indices considering all q configurations in the i-th design solution,

$\sum_{j=1,j\neq k}^{q} \sum_{k=1}^{q} I_R(R_{ijk})$ is the sum of evaluation indices considering all reconfiguration processes in the i-th design solution,

$d_{ij,u}$ is the u-th design parameter of the C_{ij} configuration,

$d_{ij,u}^{(L)}$ and $d_{ij,u}^{(U)}$ are the lower and upper limits for $d_{ij,u}$,

$p_{ijk,u}$ is the u-th process parameter of the R_{ijk} reconfiguration process, and

$p_{ij,u}^{(L)}$ and $p_{ij,u}^{(U)}$ are the lower and upper limits for $p_{ijk,u}$.

The optimal reconfigurable product design is identified using a multi-level and multi-objective optimization method. For each design candidate, S_i, and its reconfiguration processes, parameter optimization is conducted to obtain the optimal design and process parameters, S_i^*, considering various evaluation measures. From all the feasible design candidates and reconfiguration processes, the optimal design and its reconfiguration processes are subsequently obtained. The new multi-level and multi-objective optimization method is more effective than other optimization methods used in Table 2.1.

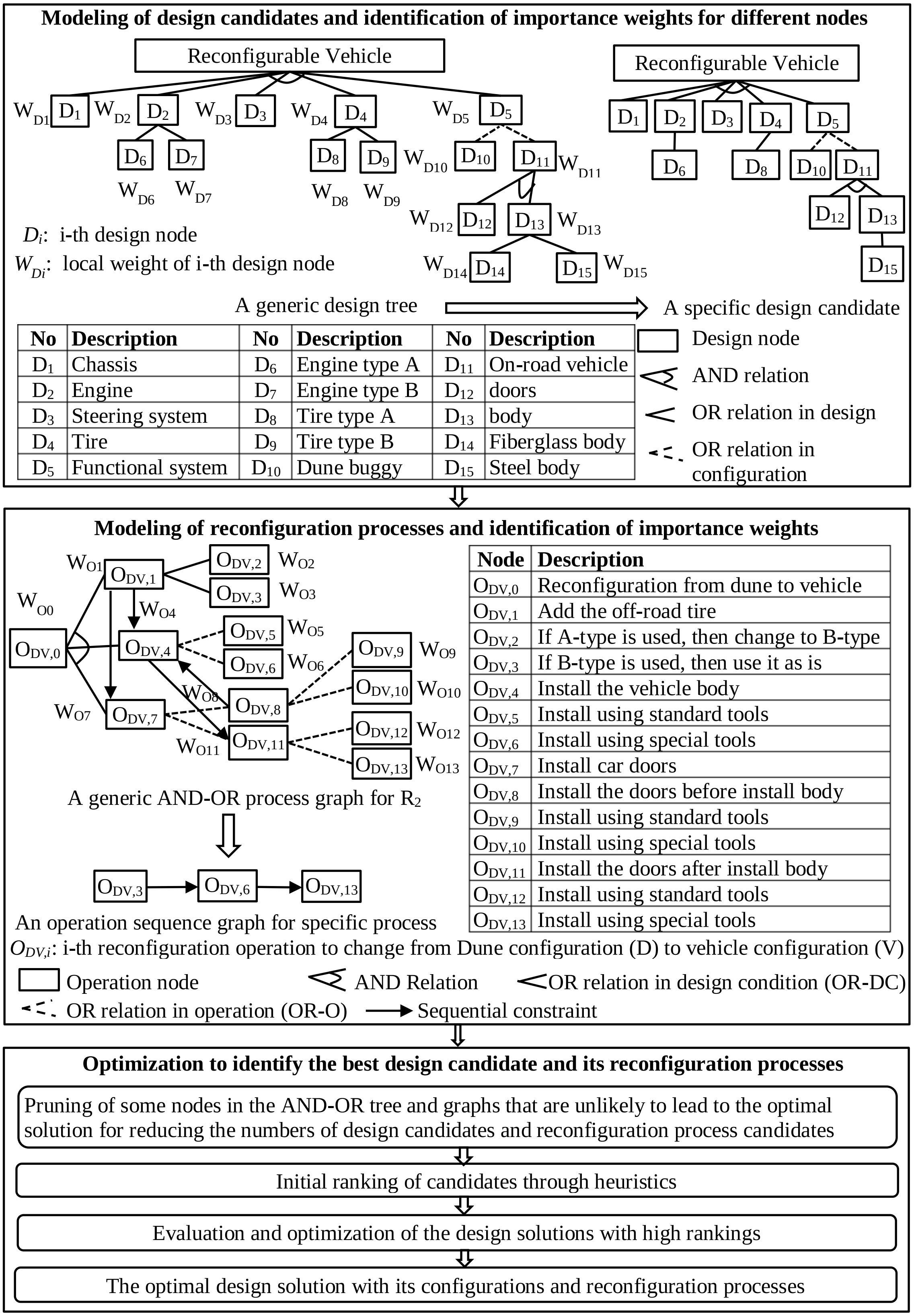
3.3 Improvement of Optimization Efficiency for Optimal Design of Reconfigurable Products Considering Large Numbers of Design/Process Candidates

For the design of reconfigurable products, multiple design candidates are usually considered based on design requirements, where each of these design candidates can have multiple product configurations. For a reconfiguration process to change between two product configurations, multiple reconfiguration process candidates are also considered. When numbers of design candidates and reconfiguration process candidates are large, computation efficiency has to be considered such that the optimal solution can be achieved within the required computation time period. In this research, a new optimization method is developed to improve the computation efficiency of the previously developed optimal reconfigurable product design method introduced in Section 3.2 when the numbers of design candidates and reconfiguration process candidates are large. First, importance weights of nodes in an AND-OR tree or AND-OR graph are obtained by evaluating these nodes using heuristics. Then, the nodes in the AND-OR tree/graph with poor weights are deleted through a pruning method to reduce the optimization search space. The initial ranking of the created design solutions that are modeled by design candidates and their reconfiguration processes is subsequently conducted through heuristic-based evaluation. The top-ranked design solutions are further evaluated with sophisticated evaluation methods to identify the optimal design solution. The new efficient optimization approach for the design of reconfigurable products considering large numbers of design candidates and reconfiguration process candidates with three steps is shown in Figure 3.2.

3.3.1 Modeling of design candidates and identification of importance weights for different design nodes

Since different design candidates are modeled by a generic design AND-OR tree with design nodes, evaluations to these nodes considering their contributions to the optimal design are first conducted. In this research, heuristics are employed to evaluate these nodes efficiently, and the evaluation results are described by weights of the nodes in the AND-OR tree. Weight is defined by a value in the range of [0, 1], corresponding to the least important measure and the most important measure respectively.

For any node in the design tree, its relative importance related to other nodes within the same group of sub-nodes in an AND or OR relation is modeled by a local weight W_L. Meaning of a local weight in different types of relations in the design AND-OR tree is summarized in Table 3.2. The global weight of a node in a design AND-OR tree is used to model the importance of this node in the whole design. In this research, the global weight, W_G, of a node is calculated based on the hierarchical relations in the AND-OR tree considering different types of relations in the AND-OR tree.



No	Description	No	Description	No	Description
D_1	Chassis	D_6	Engine type A	D_{11}	On-road vehicle
D_2	Engine	D_7	Engine type B	D_{12}	doors
D_3	Steering system	D_8	Tire type A	D_{13}	body
D_4	Tire	D_9	Tire type B	D_{14}	Fiberglass body
D_5	Functional system	D_{10}	Dune buggy	D_{15}	Steel body

Node	Description
$O_{DV,0}$	Reconfiguration from dune to vehicle
$O_{DV,1}$	Add the off-road tire
$O_{DV,2}$	If A-type is used, then change to B-type
$O_{DV,3}$	If B-type is used, then use it as is
$O_{DV,4}$	Install the vehicle body
$O_{DV,5}$	Install using standard tools
$O_{DV,6}$	Install using special tools
$O_{DV,7}$	Install car doors
$O_{DV,8}$	Install the doors before install body
$O_{DV,9}$	Install using standard tools
$O_{DV,10}$	Install using special tools
$O_{DV,11}$	Install the doors after install body
$O_{DV,12}$	Install using standard tools
$O_{DV,13}$	Install using special tools

Figure 3.2: The approach to improve optimization efficiency for optimal design of reconfigurable products.

Table 3.2: Meaning of local weights in different types of relations in the design AND-OR tree.

Relation type	Meaning of local weight	Example
AND relation (AND)	Partial contribution of the sub-node to the parent node	A car is composed of many modules, and local weight of the body is 0.4.
OR relation in design (OR-D)	How good the sub-node is compared with other alternative sub-nodes of the parent node	Material of the car body can be alloy or aluminum, and the local weight of alloy is 0.6.
OR relation in configuration (OR-C)	Partial contribution of the sub-node to the parent node in a product configuration	Both solar power and electrical power are used to recharge the battery of a system, and local weight of the solar power charging system is 0.7.

3.3.2 Modelling of reconfiguration processes and identification of importance weights for different operation nodes

For a reconfigurable product with multiple configurations, the generic reconfiguration process to change between two configurations is modeled by a generic process AND-OR graph with operation nodes and their parameters. Since sequential constraints among operations are also considered, the generic reconfiguration process is defined by a graph instead of a tree. Same as the design tree, local weight W_L and global weight W_G of an operation node are used to describe its importance related to other nodes with an AND or OR relation and its importance related to other nodes in the whole reconfiguration process graph. Meaning of a local weight in different types of relations in the reconfiguration process AND-OR graph is summarized in Table 3.3.

Table 3.3: Meaning of local weights in different types of relations in the reconfiguration process AND-OR graph.

Relation type	Meaning of local weight	Example
AND relation (AND)	Partial contribution of the sub-node to the parent node	Change of the fixture is conducted by removing the old fixture and installing the new fixture, and local weight for removing the old fixture is 0.4.
OR relation in operation (OR-O)	How good the sub-node is compared with other alternative sub-nodes of the parent node	Change of the power charging system can be conducted either manually or automatically, and the local weight for manual change is 0.4.
OR relation in design condition (OR-DC)	Same contribution as the parent node since only the sub-node for the selected design candidate is used	Mechanical operation or electrical operation is conducted based on whether mechanical power or electrical power is selected in design, and the local weight of mechanical operation or electrical operation is not required.

3.3.3 Optimization to identify the best design candidate and its reconfiguration processes

When numbers of design candidates and reconfiguration process candidates are large, pruning is conducted based on a modified beam search to delete some nodes that are unlikely to lead to the optimal solution. From the sub-nodes with OR relation in design (OR-D) in a generic design AND-

OR tree or OR relation in operation (OR-O) in a generic reconfiguration process AND-OR graph, at least one node should be selected. For the remaining nodes, some are pruned based on the importance weights of these nodes through a new modified beam search method to reduce the search space. The beam width parameters at different levels in the tree/graph are selected based on the numbers of nodes at these levels in the tree/graph and the computation efficiency requirement.

When the numbers of the design candidates and reconfiguration process candidates are still large, an initial ranking of the created candidates is conducted to evaluate these candidates using a simple heuristic function. Only the top-ranked candidates are selected for detailed evaluation to further improve the computation efficiency. The multi-level and multi-objective optimization method is then used to obtain the final optimal solution modeled by the design candidate, its reconfiguration processes, and parameters of the design candidate and reconfiguration processes.

3.4 Improvement of Optimization Quality for Optimal Design of Reconfigurable Products Considering Dependency Relations among Descriptions of Design/Process Candidates and Evaluation Criteria

In this research, a new method, called extended analytic network process (EANP) method, has been developed. The proposed method improves the quality of the optimization method for optimal design of reconfigurable products introduced in Section 3.3. The proposed method considers dependency relations among design/process descriptions and evaluation criteria for rating of contributions of these partial design/process descriptions to the optimal design. The new method is used to achieve better initial evaluations for design/process nodes which truly reflect the contributions of these nodes to the optimal solutions when the dependency relations among these nodes have to be considered. Since the truly poor nodes are cut off in this new method, better optimal designs of reconfigurable products can be achieved. In addition, by selecting the evaluation criteria in the EANP method using similar evaluation measures in the final optimization process, better top-ranked solutions can be identified for detailed optimization. The new approach has three steps as shown in Figure 3.3.

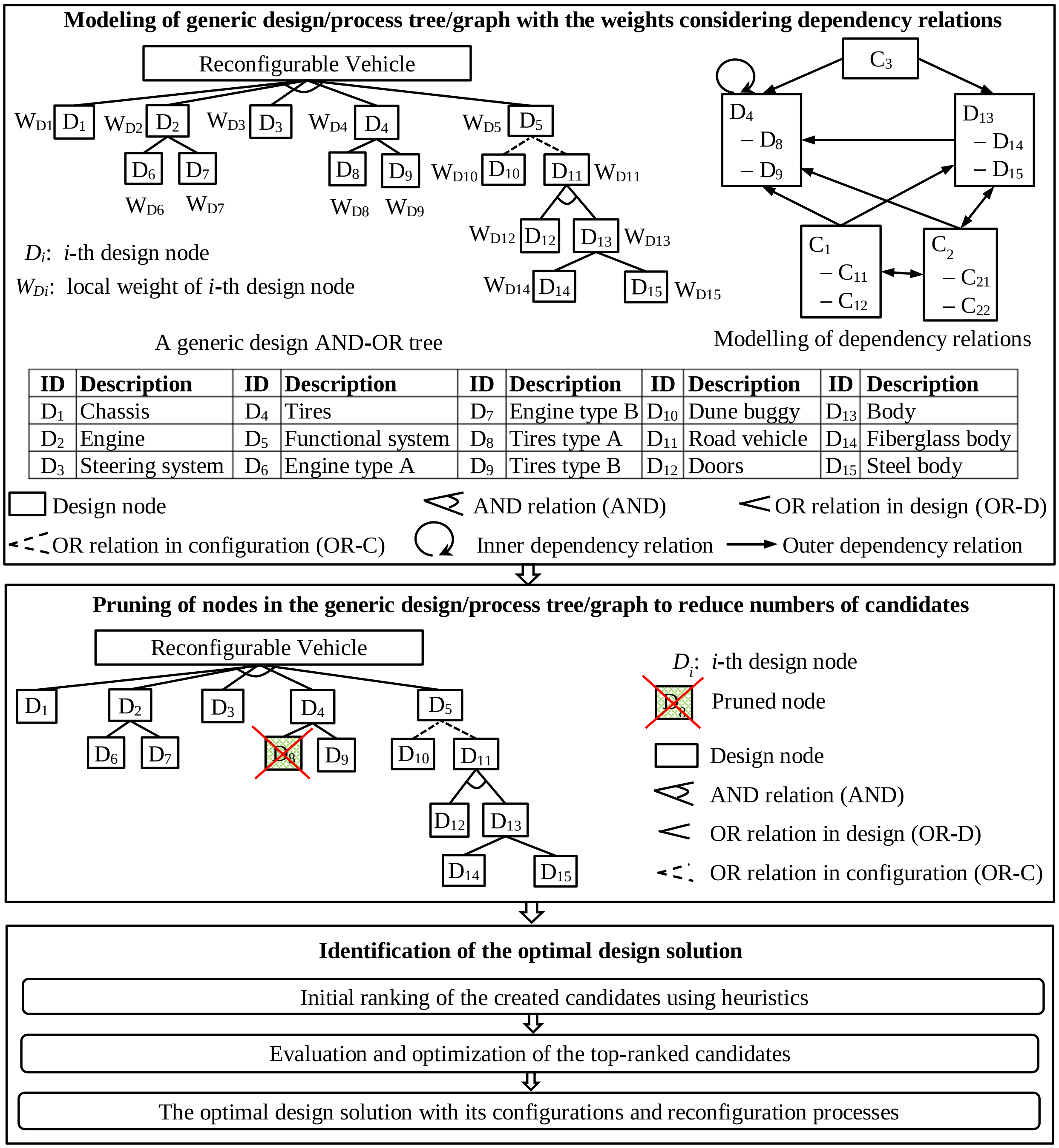


Figure 3.3: The approach to improve optimization quality for optimal design of reconfigurable products.

3.4.1 Modeling of generic AND-OR tree/graphs with the weights considering dependency relations among design/process nodes and evaluation criteria

Since different design candidates are modeled by a generic design AND-OR tree with design nodes, and different reconfiguration process candidates to change between two product configurations are modeled by a generic reconfiguration process AND-OR graph with operation nodes, evaluations to these nodes considering dependency relations among design/process nodes and evaluation criteria are first conducted using a new extended analytic network process (EANP) method.

A new extended analytic network process (EANP) method is developed to identify the relative importance weights for sub-nodes with an AND or OR relation in the tree/graph when decisions to select partial design/process nodes are influenced by decisions to select other design/process nodes due to the dependency relations among design/process nodes and evaluation criteria. For example, Figure 3.4(a) shows a generic design AND-OR tree for modeling a car which is composed of a body, tires, mechanical systems, and electrical systems with the assigned local weights W_L. For this car, when an armored body is preferred, armored tires should also be preferred. In other words, the decision to select the design solution for the tires is influenced by the decision to select the design solution for the body. Due to the dependency relation, the weight of the armored tires is increased from the initial value of 0.5 to the adjusted one of 0.65, and weight of the standard tires is decreased from the initial value of 0.5 to the adjusted one of 0.35 as shown in Figure 3.4(b). Then based on the calculated local weights W_L, the global weights of these nodes W_G are calculated.

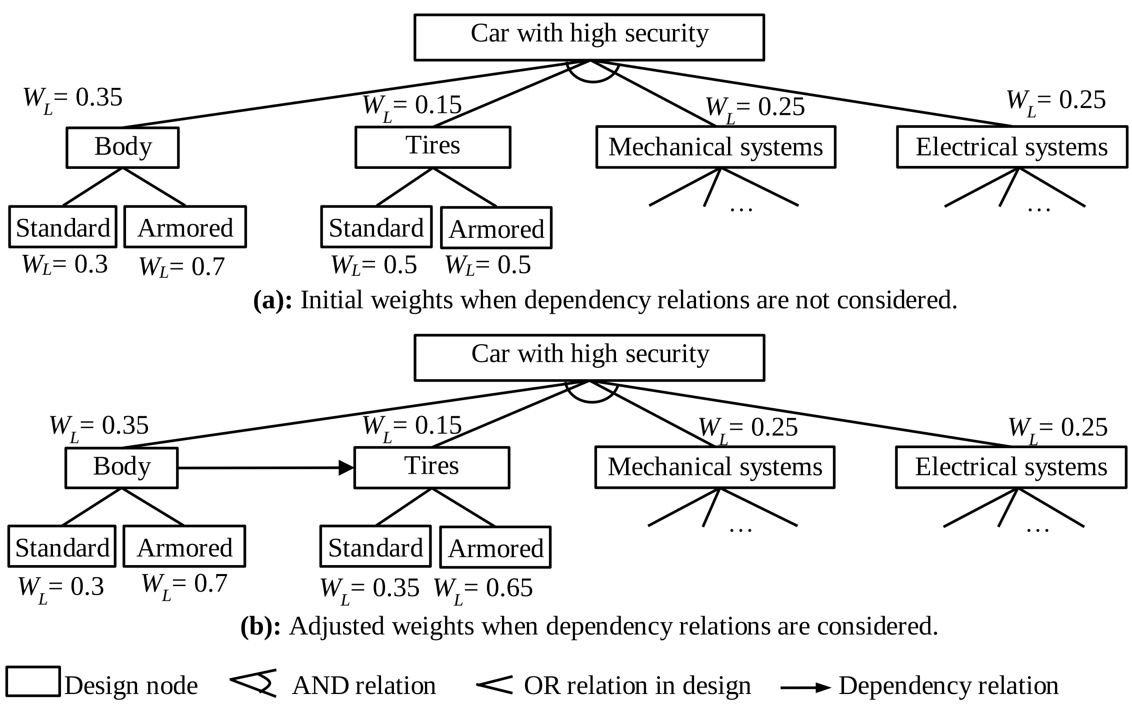


Figure 3.4: Influences of dependency relations on weights of nodes in a design AND-OR tree.

3.4.2 Pruning of nodes in the generic AND-OR tree/graphs to reduce the numbers of design candidates and reconfiguration process candidates

When large numbers of design and process candidates can be created from the generic design AND-OR tree and reconfiguration process AND-OR graphs, some nodes with OR relations in design for the design tree and some nodes with OR relation in operations for the process graphs, which are unlikely to achieve the optimal solutions, are pruned to reduce the numbers of candidates. Pruning method introduced in Section 3.3.3 is used to cut off the design/process nodes using a modified beam-search method.

3.4.3 Identification of the optimal design solution

Specific design/process candidates are created from the pruned generic tree/graph through tree/graph-based search. When the numbers of created design candidates and reconfiguration process candidates are still large, further improvement of optimization efficiency is conducted by an initial ranking of the created candidates using a simple heuristic function to evaluate these candidates. Then, to identify the optimal design solution and its reconfiguration processes, first, the top-ranked design solutions created from the top-ranked design candidates and top-ranked process candidates are evaluated by multiple evaluation measures. Then, the optimal design solution that is modelled by a design candidate with different configurations, its reconfiguration processes, and design/process parameter values is identified by a multi-level and multi-objective optimization method. The methods introduced in Section 3.3.3 are used to identify the optimal design solution.

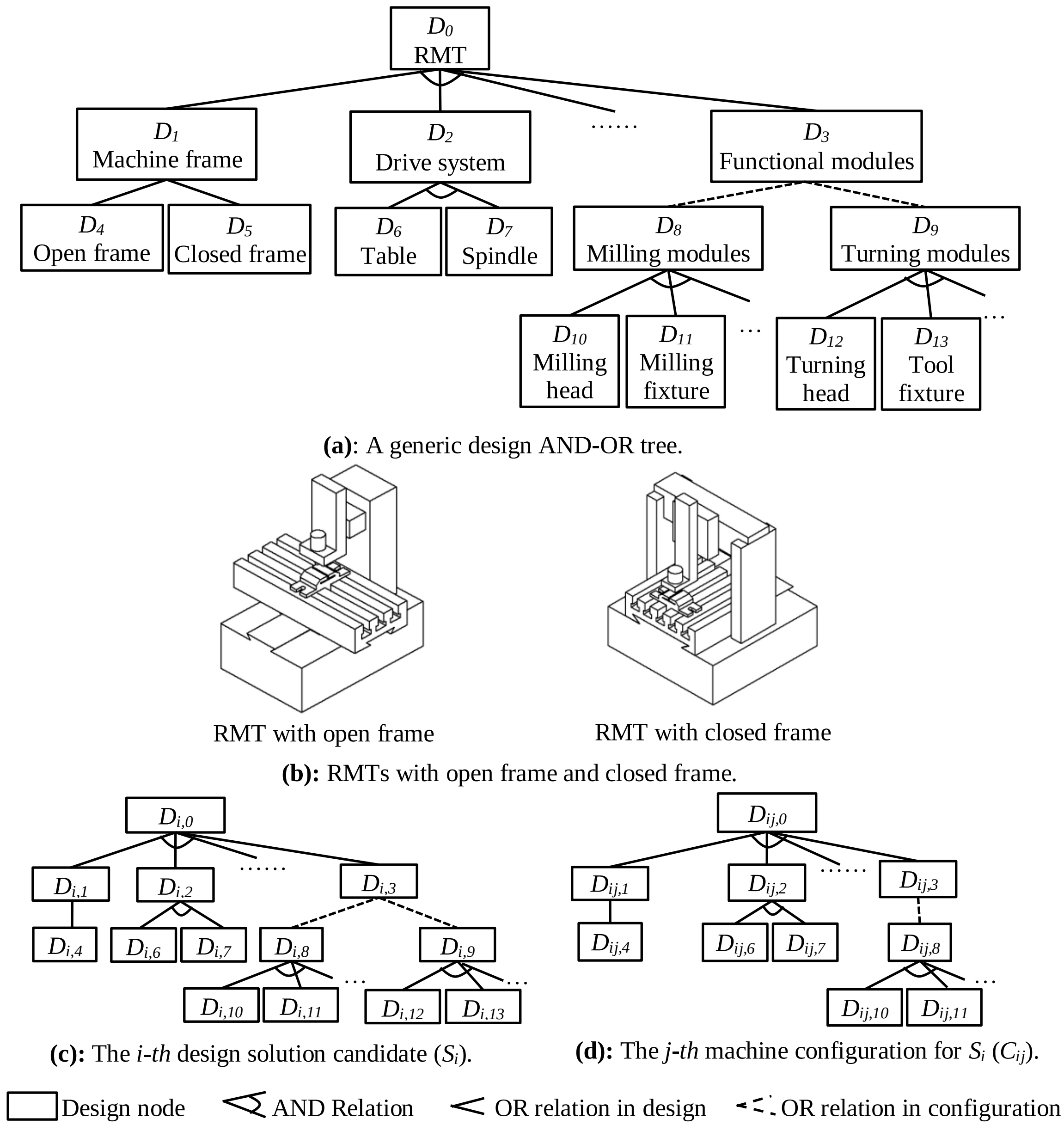
Chapter 4 An Approach to Identify the Optimal Configurations and Reconfiguration Processes for Design of Reconfigurable Products

As discussed in Chapter 3, the optimal reconfigurable product design is conducted through three steps. First, different design solution candidates are modeled using a generic design AND-OR tree and each of these design solution candidates is further modeled by its product configurations and parameters of these configurations. A reconfiguration process to change between two configurations in a design solution and parameters of this reconfiguration process are then modeled using a generic reconfiguration process AND-OR graph. Based on evaluations to design solutions considering different product configurations and reconfiguration processes, the optimal design is subsequently identified through multi-level and multi-objective optimization.

4.1 Modelling of Different Design Solution Candidates and Their Product Configurations

4.1.1 Modeling and creation of design candidates using a hybrid AND-OR tree

From the design requirements, a hybrid AND-OR tree, as shown in Figure 4.1(a), is used to model generic design solution candidates. Each node in the generic AND-OR tree is used to model a partial design solution (e.g. a sub-assembly or a component). The relations among sub-nodes of a super-node in the generic AND-OR tree, as introduced in Section 3.2.1, are classified into three categories: AND relation (i.e., composition relation) in which all sub-nodes should be used together to obtain/enable the parent node, OR relation in design (OR-D) in which only one sub-node should be selected to obtain/enable the parent node to create a specific design candidate, and OR relation in configuration (OR-C) when different sub-nodes are used to model different product configurations for delivering different functions. For the example given in Figure 4.1(a), the reconfigurable machine tool (RMT) is modeled by three sub-systems: machine frame, drive system, and functional modules (Gadalla and Xue 2017b). The machine frame can be either open frame or closed frame (Slocum 1992) as shown in Figure 4.1(b). A drive system is modeled by the machine table drive system and the machine spindle drive system. In this example, the functional modules include milling modules for functions in the milling machine configuration and turning modules for functions in the turning machine configuration, since milling functions and turning functions are considered the most fundamental functions for a reconfigurable machine tool (Gadalla and Xue 2017a).



(a): A generic design AND-OR tree.

(b): RMTs with open frame and closed frame.

(c): The *i-th* design solution candidate (S_i).

(d): The *j-th* machine configuration for S_i (C_{ij}).

Figure 4.1: Modelling of design solution candidates and their operation configurations.

Each design solution candidate, S_i, is modeled by a collection of product configurations (C_{ij}), a collection of design nodes ($D_{i,m}$), and a collection of design parameters ($d_{i,n}$).

$$S_i = \{C_{i1}, C_{i2}, \dots, C_{iq}\},\ i = 1, 2, \dots, p$$

$$S_i = \{D_{i,1}, D_{i,2}, \dots, D_{i,m}, \dots,\ D_{i,m_i}\},\ i = 1, 2, \dots, p \tag{4.1}$$

$$S_i = \{d_{i,1}, d_{i,2}, \dots, d_{i,n}, \dots,\ d_{i,n_i}\},\ i = 1, 2, \dots, p$$

where p is the number of design candidates, m_i is the number of design nodes for the i-th design candidate, and n_i is the number of design parameters for the i-th design candidate.

Each design node in the generic AND-OR tree is defined by parameters and constraints among parameters. Parameters for the design node, $D_{i,m}$, are defined by:

$$D_{i,m} = \{d_{i,m,1},\ d_{i,m,1}, \ldots\}, \quad i=1,2,\ldots,p;\ m=1,2,\ldots,m_i \tag{4.2}$$

The constraints among parameters are defined by:

$$d_{i,n}^{(L)} \leq d_{i,n} \leq d_{i,n}^{(U)}, \quad i=1,2,\ldots,p;\ n=1,2,\ldots,n_i \tag{4.3}$$

where $d_{i,n}^{(L)}$ and $d_{i,n}^{(U)}$ are the lower and upper boundaries for $d_{i,n}$.

In this work, the following rules are used to generate different design solution candidates from the generic hybrid AND-OR tree (Hong et al. 2010):

(1) First, select the root node.

(2) For a selected node, when its sub-nodes have an AND relation, select all these sub-nodes.

(3) For a selected node, when its sub-nodes have an OR relation in configuration, select all these sub-nodes.

(4) For a selected node, when its sub-nodes have an OR relation in design, select only one of these sub-nodes.

A created design solution candidate is modeled by an AND-OR tree with AND relations and OR-C relations. Figure 4.1(c) shows a design solution candidate created from the AND-OR tree given in Figure 4.1(a).

4.1.2 Modelling of product configurations for each design solution candidate

Each product configuration, C_{ij}, is modeled by a collection of design nodes, $D_{ij,m_{ij}}$:

$$C_{ij} = \left\{D_{ij,1},\ D_{ij,2}, \ldots, D_{ij,m}, \ldots, D_{ij,m_{ij}}\right\}, \quad i=1,2,\ldots,p;\ j=1,2,\ldots,q \tag{4.4}$$

A product configuration, C_{ij}, can also be modeled by its parameters.

$$C_{ij} = \left\{d_{ij,1},\ d_{ij,2}, \dots, d_{ij,n}, \dots, d_{ij,n_{ij}}\right\},\quad i=1,2,\dots,p;\ j=1,2,\dots,q \tag{4.5}$$

Each AND-OR tree of the design solution candidate can be used to create multiple product configurations. A product configuration for a specific design solution candidate is created based on the following rules:

(1) First, select the root node.

(2) For a selected node, when its sub-nodes have an AND relation, select all these sub-nodes.

(3) For a selected node, when its sub-nodes have an OR relation in configuration, select only one of these sub-nodes.

A created product configuration is modeled by a tree with only AND relations. Figure 4.1(d) shows a product configuration created from the AND-OR tree given in Figure 4.1(c).

4.2 Modelling of Reconfiguration Processes

After creation of all design candidates and configurations for each of these design candidates, modelling of the reconfiguration processes among different configurations for each design candidate should be conducted. The reconfiguration processes are also modeled at two different levels: generic reconfiguration processes considering all feasible operations and specific reconfiguration processes created from the generic processes.

4.2.1 Modelling of generic reconfiguration processes for the selected design candidate

A process AND-OR graph, as shown in Figure 4.2(a), is used to model the generic reconfiguration process to change between two different configurations (Hong et al. 2010). A generic process AND-OR graph is composed of operation nodes, parameters of operation nodes, and sequential constraints among operation nodes. The relations among sub-nodes of a super-node in the generic reconfiguration process AND-OR graph, as introduced in Section 3.2.2, are classified into three categories: AND relation (i.e., composition relation) in which all sub-nodes should be used together to obtain/enable the parent node, OR relation in operation (OR-O) when alternative operations are considered to achieve the same operation function, and OR relation in design condition (OR-DC) through IF-THEN descriptions when alternative operations are considered for different design candidates to achieve the same operation function. Each operation node in the reconfiguration process AND-OR graph is composed of operation parameters.

For the design candidate shown in Figure 4.1(d), the generic reconfiguration process to change from the milling machine configuration to the turning machine configuration is modeled by a generic process AND-OR graph as shown in Figure 4.2(a). In this example, the reconfiguration is conducted by removing the milling fixture, changing location of the milling head based on whether the open frame structure or closed frame structure is used, and adding other modules for the turning function.

4.2.2 Creation of a specific reconfiguration process

From the generic reconfiguration process, a specific reconfiguration process is created through graph-based search in the following steps as shown in Figure 4.2 (Gadalla and Xue 2017b).

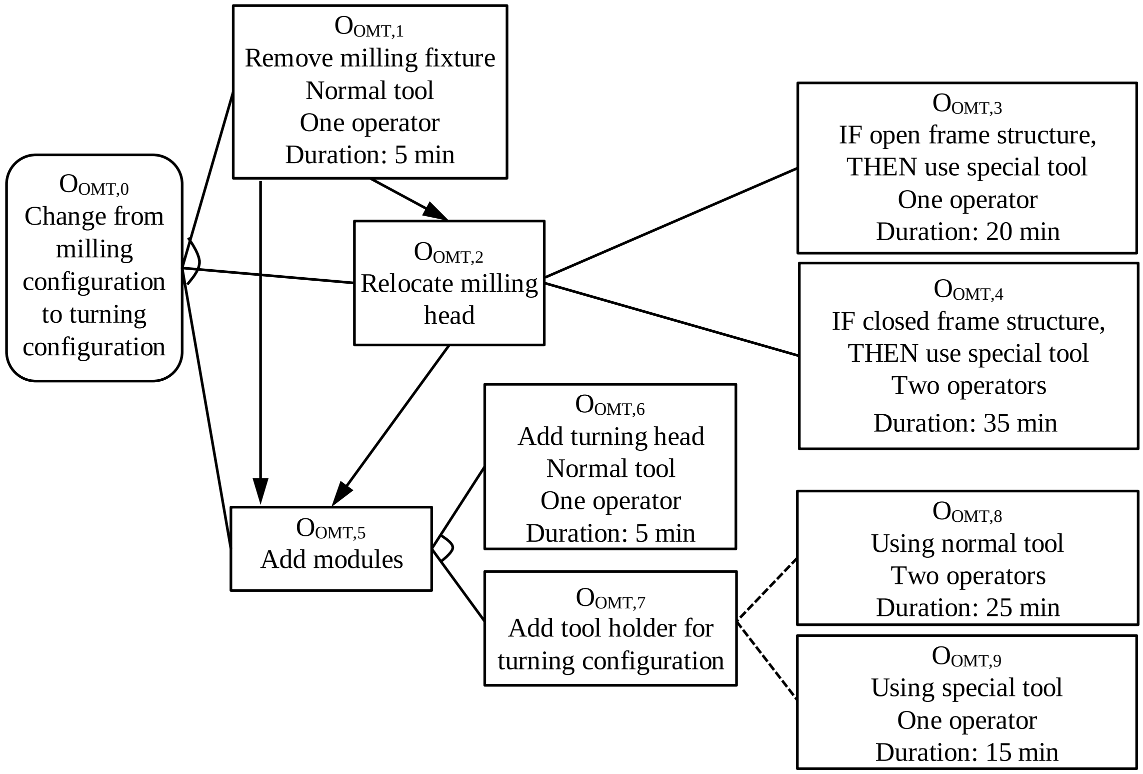
(1) Consider the generic reconfiguration process AND-OR graph (Figure 4.2(a)), and initialize an empty specific reconfiguration process graph. Select the root node of the generic graph and add it to the specific graph as the root node.

(2) From the generic reconfiguration process graph, select a node that has not been checked.

(2.1) If the sub-nodes of the selected node are associated with an AND relation in the generic reconfiguration process graph, these sub-nodes should be added to the specific reconfiguration process graph as the sub-nodes of the selected node.

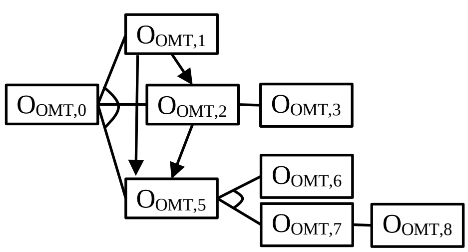
(2.2) If the sub-nodes of the selected node are associated with an OR relation in operation (OR-O) in the generic reconfiguration process graph, only one of these sub-nodes should be selected randomly and added to the specific reconfiguration process graph as the sub-node of the selected node.

(2.3) If the sub-nodes of the selected node are associated with an OR relation in design condition (OR-DC) described by IF-THEN conditions in the generic reconfiguration process graph, only one of these sub-nodes should be selected by matching the IF conditions with the created design candidate. This matched sub-node is then added to the specific reconfiguration process graph as the sub-node of the selected node.

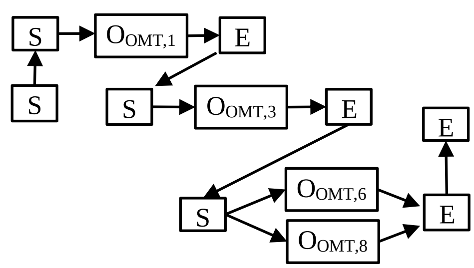


OMT: To change from milling configuration (*M*) to turning configuration (*T*)
for the design candidate with open frame structure (*O*)

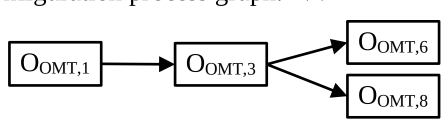
(a): A generic process AND-OR graph.



(b): Creation of a specific reconfiguration process graph.



(c): Creation of an operation sequence graph.



(d): Simplification of the operation sequence graph.

Operation node
AND relation
OR relation in design condition
OR relation in operation
Sequential constraint

Figure 4.2: Modelling and creation of reconfiguration processes.

(3) The specific reconfiguration process graph is completed after all nodes in the generic AND-OR graph have been checked (Figure 4.2(b)). Otherwise go to Step (2).

(4) Generate an operation sequence graph from the specific reconfiguration process graph.

(4.1) Select a bottom node from the specific reconfiguration process graph and remove it from this graph.

(4.2) When the selected node is a bottom operation node in the specific reconfiguration process graph, add this node to the operation sequence graph.

(4.3) When the selected node is not a bottom operation node in the specific reconfiguration process graph, add the sequential relations among the sub-nodes of this selected node as a sub-graph to the operation sequence graph. In this sub-graph, a start-node, S, and an end-node, E, should be added as the ancestor and descendant nodes for those nodes without ancestor nodes and descendant nodes, respectively. When a sub-node is not a bottom operation node in the specific reconfiguration process graph, the *S* node of the sub-graph created from this sub-node should be linked with the ancestor nodes defined in the sub-graph of the selected node, and the *E* node of the sub-graph created from this sub-node should be linked with the descendant nodes defined in the sub-graph of the selected node.

(4.4) When all nodes in the specific reconfiguration process graph have been removed, creation of the operation sequence graph is completed (Figure 4.2(c)). Otherwise, go to Step (4.1).

(5) The operation sequence graph is simplified as the final reconfiguration process by removing unnecessary *S* nodes and *E* nodes (Figure 4.2(d)).

4.3 Identification of the Optimal Design

In the optimal reconfigurable product design approach, first each of the feasible design solutions is modeled by its product configurations, reconfiguration processes, and parameters of product configurations and reconfiguration processes. The optimal solution is identified based on evaluations to all feasible design solutions.

4.3.1 Evaluation of design candidate

A feasible design solution, S_i (i = 1, 2, …, p), for reconfigurable product design is modeled by its product configurations C_{ij} ($i = 1, 2, ..., p; j = 1, 2, ..., q$), parameters of product configurations $d_{ij,1}, d_{ij,2}, \dots, d_{ij,u_{ij}}$ (i = 1, 2, …, p; j = 1, 2, …, q), reconfiguration processes R_{ijk} (i = 1, 2, …, p; j = 1, 2, …, q; k = 1, 2, …, q; $j \neq k$), and parameters of reconfiguration processes $p_{ijk,1}, p_{ijk,2}, \dots, p_{ijk,u_{ijk}}$ (i = 1, 2, …, p; j = 1, 2, …, q; k = 1, 2, …, q; $j \neq k$).

Suppose $\boldsymbol{X_i}$ is the vector to model all the parameters including design parameters and process parameters, the design solution S_i can be evaluated by m evaluation measures:

$$F_j = f_j(S_i) = f_j(\boldsymbol{X_i}),\ i = 1, 2, ..., p;\ j = 1, 2, ..., m \tag{4.6}$$

The design solutions of reconfigurable product are usually evaluated by different evaluation measures such as cost measures (e.g., total cost of configurations and reconfiguration processes), time measures (e.g., reconfiguration durations), and performance measures. Because these evaluation measures are usually described in different units, these evaluation measures are converted into comparable evaluation indices between 0 and 1, representing levels of satisfaction (Xue, Yang, and Tu 2006). In this research, the *j-th* evaluation measure is converted into a satisfaction index, $I_j(\boldsymbol{X_i})$, using:

$$I_j(\boldsymbol{X_i}) = g_j[f_j(\boldsymbol{X_i})],\ \ i = 1, 2, ..., p;\ \ j = 1, 2, ..., m \tag{4.7}$$

In Equation (4.7), the non-linear relation between the evaluation measure $f_j(\boldsymbol{X_i})$ and the evaluation index $I_j(\boldsymbol{X_i})$ is defined by a polynomial function, and the coefficients of this polynomial function are obtained using a least-square curve-fitting method based on the given data of evaluation measures and their corresponding evaluation indices. The overall evaluation index, $I(\boldsymbol{X_i})$, considering all evaluation measures can be calculated by:

$$I(\boldsymbol{X_i}) = \frac{\sum_{j=1}^{m}\left(W_j I_j(\boldsymbol{X_i})\right)}{\sum_{j=1}^{m} W_j} \tag{4.8}$$

where $W_1, W_2, \dots, W_m$ are the m weighting factors representing importance levels of these evaluation indices. Values of these weighting factors are selected by engineers and customers based on their experience and preference.

4.3.2 Multi-level and multi-objective optimization

Based on the evaluations to different design candidates using the overall evaluation index, the optimal design solution candidate and its parameters can then be identified through a multi-level and multi-objective optimization method. The optimization is conducted at two different levels: the candidate optimization level to identify the optimal design candidate that is modeled by its product configurations and reconfiguration processes, and the parameter optimization level to identify the optimal parameter values of product configurations and reconfiguration processes. When the i-th feasible candidate is modeled by a set of parameters, $\boldsymbol{X_i}$, including design parameters and process parameters, the parameter optimization problem is defined by:

$$\begin{aligned} &\max_{w.r.t.\boldsymbol{X}_i} I(\boldsymbol{X_i}) \\ &\quad s.t.\ \ \boldsymbol{X}_i^{(L)} \le \boldsymbol{X_i} \le \boldsymbol{X}_i^{(U)},\ \ i=1,2,...,p \\ &\qquad\quad h_{ik}(\boldsymbol{X_i}) \ge 0,\ \ i=1,2,...,p;k=1,2,... \\ &\qquad\quad g_{ik}(\boldsymbol{X_i}) = 0,\ \ i=1,2,...,p;k=1,2,... \end{aligned} \tag{4.9}$$

The optimal design candidate is identified by:

$$\max_{w.r.t.i} I(S_i^*) \tag{4.10}$$

where $I(S_i^*)$ is the i-th design candidate with the best overall evaluation index.

Parameter optimization is conducted through numerical search (Arora 1989). When constraints are considered, penalty functions are used to convert a constrained optimization problem into an unconstrained optimization problem. When the number of possible design candidates is large, genetic programming (Koza 1992) is then employed for design candidate optimization.

4.4 A Case Study

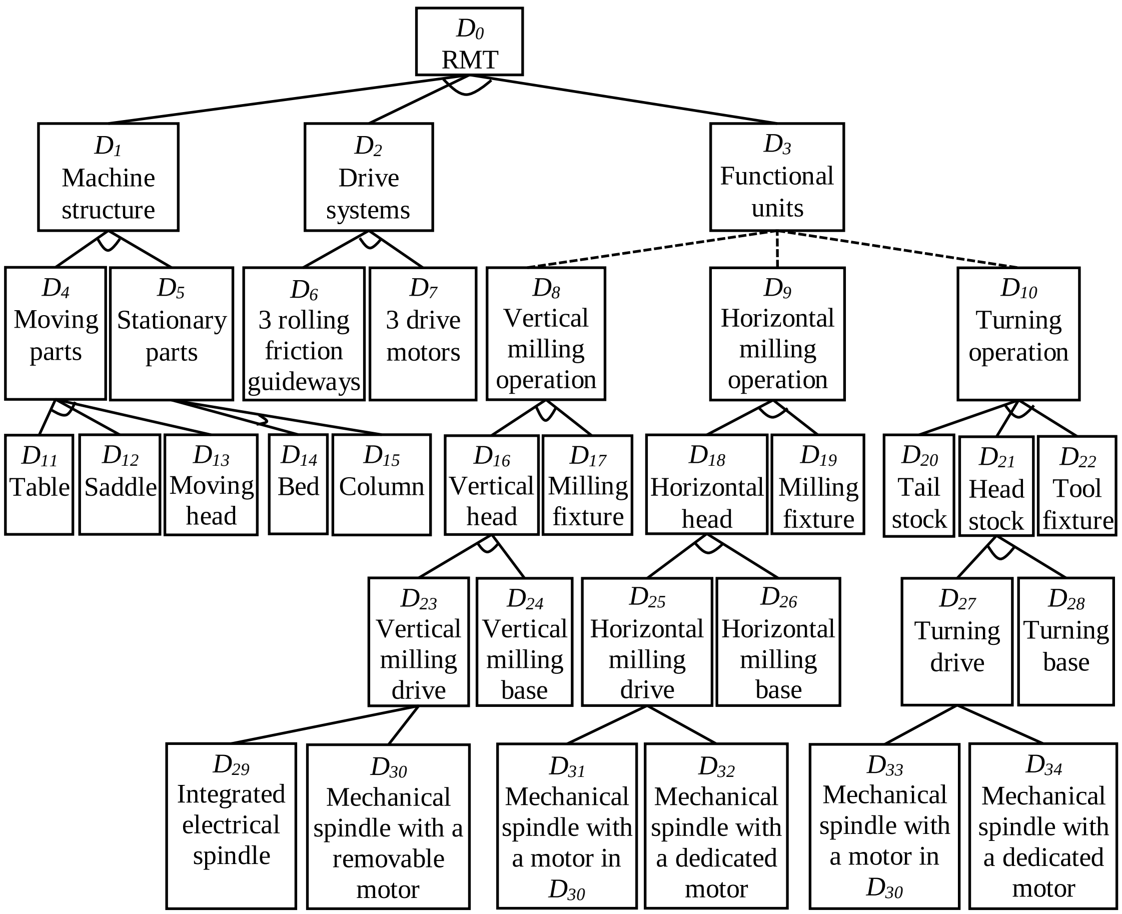
The problem of this case study was to decide some key parameters of a reconfigurable machine tool (RMT). This RMT had three configurations, a vertical milling configuration, a horizontal milling configuration, and a turning configuration, for machining large-size components. Since machining of large-size components was required only occasionally, a RMT was designed to serve as three machines to reduce the cost.

4.4.1 Modeling of different design solution candidates and their configurations

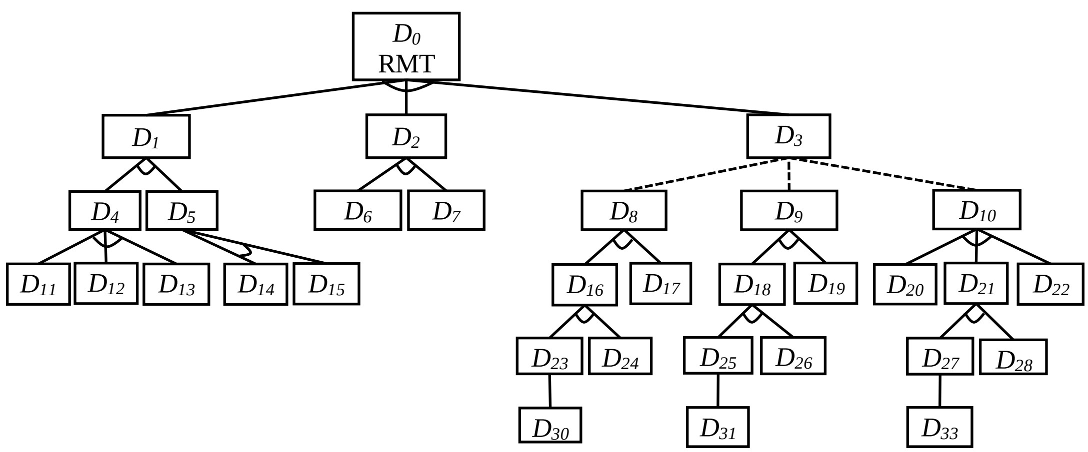
From the design requirements, a generic design AND-OR tree was developed to model design solution candidates as shown in Figure 4.3(a). Since three OR relations in design (OR-D) were used in this AND-OR tree, eight design solution candidates were created, and one of these candidates is shown in Figure 4.3(b). As discussed later in this work that the design candidate shown in Figure 4.3(b) was the best solution. In this design solution, a mechanical spindle module with a removable motor was used for the drive system in the vertical milling configuration, and the same motor was also used for the horizontal milling and turning configurations.

For the selected design candidate, three trees for three machine configurations and their geometric models, as shown in Figure 4.4, were generated from the AND-OR tree given in Figure 4.3(b) through tree-based search. Parameters of the major nodes in the machine configuration trees are shown in Table 4.1.

For the vertical milling configuration, the spindle head with the milling cutter attached to the moving head moved only in Z direction (i.e., vertical direction), and the table and the saddle moved in X and Y directions (i.e., horizontal directions). For the horizontal milling configuration, the motor for the vertical milling spindle was used to drive the horizontal milling spindle head which also moved in Z direction. For the turning configuration, the motor for the vertical milling spindle was used to drive the chuck to rotate the workpiece. Different from the traditional lathe in which the cutter moved in X and Y directions, the motions in X and Y directions in this configuration were provided by the table and the saddle, respectively.



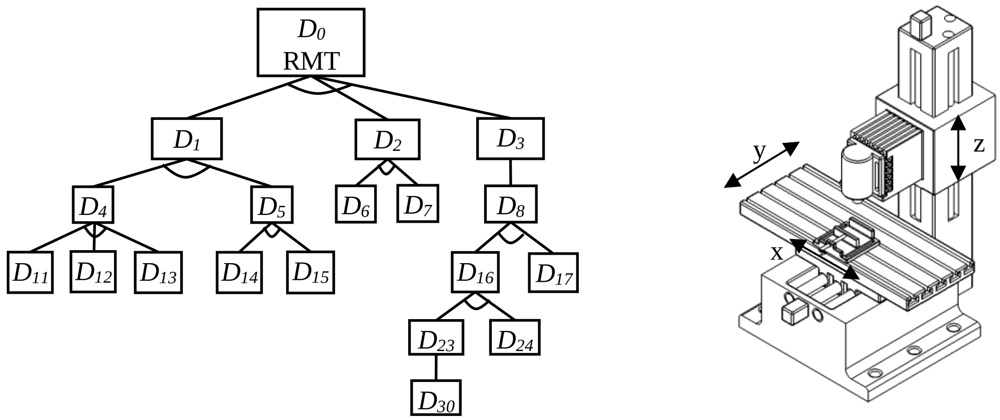
(a): A generic design AND-OR tree for modeling different design solution candidates.



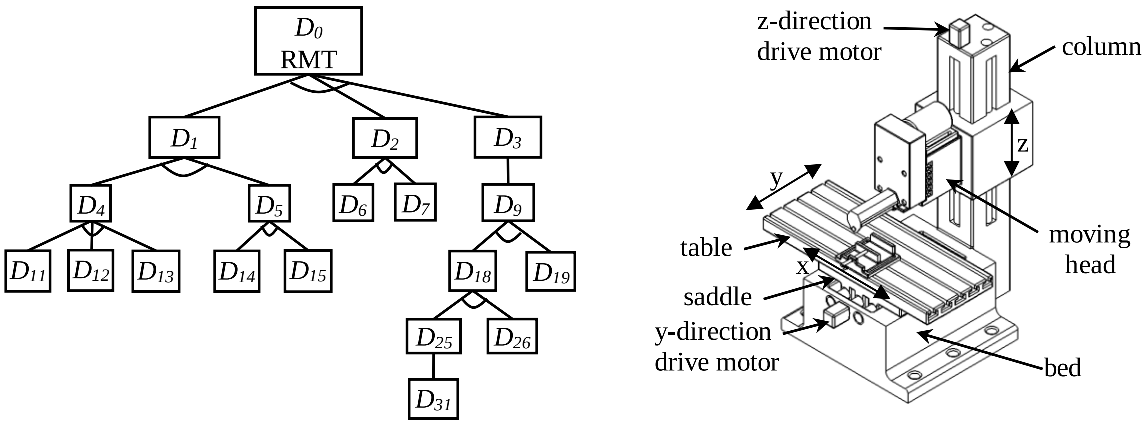
(b): A created AND-OR tree for modeling a specific design solution candidate.

Design node AND Relation OR relation in design OR relation in configuration

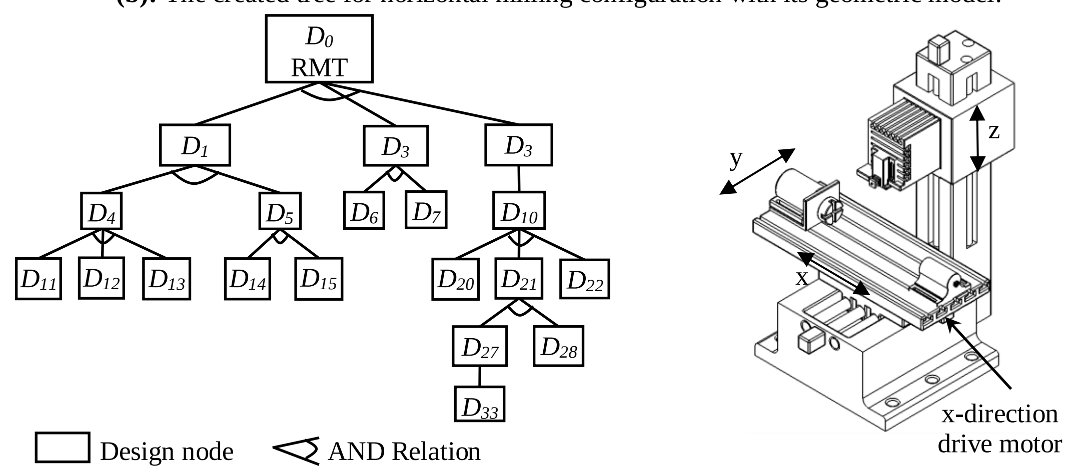
Figure 4.3: Modelling of design solutions.



(a): The created tree for vertical milling configuration with its geometric model.



(b): The created tree for horizontal milling configuration with its geometric model.



(c): The created tree for turning configuration with its geometric model.

Figure 4.4: Modeling of three machine configurations for a specific design solution.

Table 4.1: Parameters of the major nodes, D_1, D_2, D_8, D_9, D_{10}, for the selected design candidate.

Design node	Parameters
D_1	Bed dimensions (500×L_y×750 *mm*); saddle dimensions (L_x×500×150 *mm*); table dimensions (1500×750×150 *mm*); column dimensions (500×500×L *mm*); L_x, L_y, L_z are the screw lengths in x, y, and z-directions, L is the length of column (L=1050+L_z); moving head dimensions (300×300×300 *mm*).
D_2	Maximum mass of workpiece ($m_{workpiece}$=300 *kg*); density of feed screw (ρ=7861 kg/m^3); feed screw diameter (d=0.064 *m*); feed screw pitch (p=0.01 *m/rev*); inertia of motor's shaft itself ($J_3=2.373\times10^{-3}$ $kg{\cdot}m^2$).
D_8	Spindle motor power boundaries ($7.5 \leq P_m \leq 20$ *kw*); weight of milling fixture (50 *kg*); cost of milling fixture ($1000).
D_9	Weight of horizontal milling head (75 *kg*); cost of horizontal milling head ($7000).
D_{10}	Cost of turning head base ($2000); weight of 3-jaw chuck (20 *kg*); cost of chuck ($750); cost of tailstock ($1500); weight of tool fixture (25 *kg*); cost of tool fixture ($2000).

4.4.2 Modeling of reconfiguration processes

After the three machine configurations were created, identification of the six reconfiguration processes to change between machine configurations was carried out. Suppose the three machine configurations of vertical milling (V), horizontal milling (H) and turning (T) for the *1*-st design solution candidate are described by C_{1V}, C_{1H} and C_{1T}, the six reconfiguration processes are then described by R_{1VH}, R_{1VT}, R_{1HV}, R_{1HT}, R_{1TV} and R_{1TH}. In this paper, R_{1VT} is selected to show how a reconfiguration process is identified.

First the generic reconfiguration process was modeled by a generic process AND-OR graph as illustrated in Figure 4.5. Details of the operation nodes in the AND-OR graph are shown in Table 4.2. These nodes were used to model various possible operations such as to remove modules, to add modules, and to relocate modules.

From the generic AND-OR graph shown in Figure 4.5, a specific operation graph was created as shown in Figure 4.6. The algorithm provided in Section 4.2.2 was used to create this specific operation graph. For each of the operation nodes shown in Figure 4.6, various parameters were used to describe the details of the operation. In this case study, these parameters included the required base time, required number of operators, parameters related to insertion complexity and parameters related to handling complexity as shown in Table 4.3.

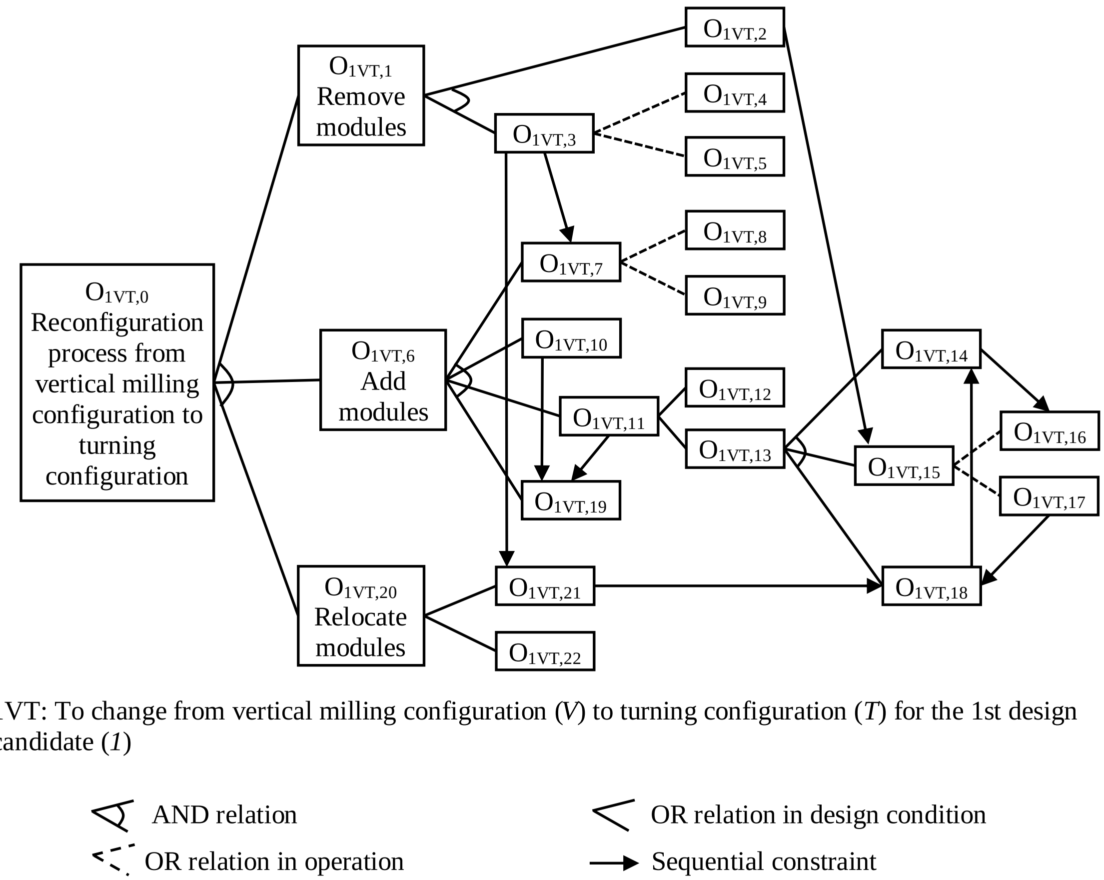


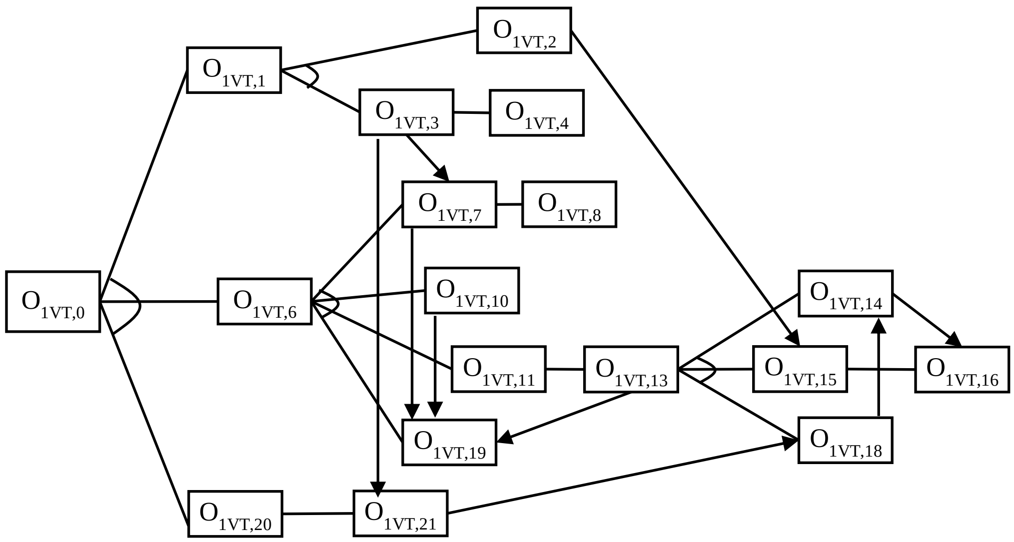
Figure 4.5: Modelling of the generic reconfiguration process R_{1VT} using an AND-OR graph.

Table 4.2: Descriptions of nodes in the AND-OR graph given in Figure 4.5.

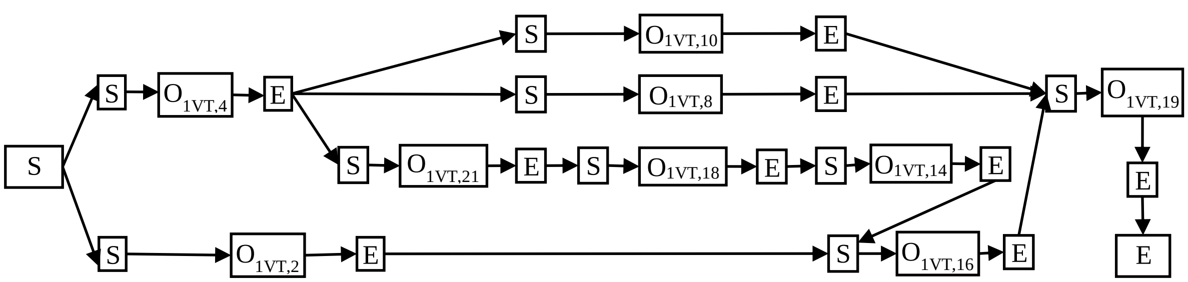
Node ID	**Node description**
$O_{1VT,1}$	Remove modules.
$O_{1VT,2}$	Remove milling fixture.
$O_{1VT,3}$	Remove milling head.
$O_{1VT,4}$	Remove milling head using workshop crane.
$O_{1VT,5}$	Remove milling head using normal tools.
$O_{1VT,6}$	Add modules.
$O_{1VT,7}$	Add tool holder for turning operation.
$O_{1VT,8}$	Add tool holder using workshop crane.
$O_{1VT,9}$	Add tool holder using normal tools.
$O_{1VT,10}$	Add tailstock.
$O_{1VT,11}$	Add head stock.
$O_{1VT,12}$	IF head stock with a dedicated motor, THEN add the turning head directly to the table.
$O_{1VT,13}$	IF head stock with the motor in D_{30}, THEN assemble the turning head and add it to the table.
$O_{1VT,14}$	Assemble 3-jaw chuck.
$O_{1VT,15}$	Assemble turning head base to machine table.
$O_{1VT,16}$	Assemble turning head base to machine table after assembling other parts.

Table 4.2: Descriptions of nodes in the AND-OR graph given in Figure 4.5 (continued).

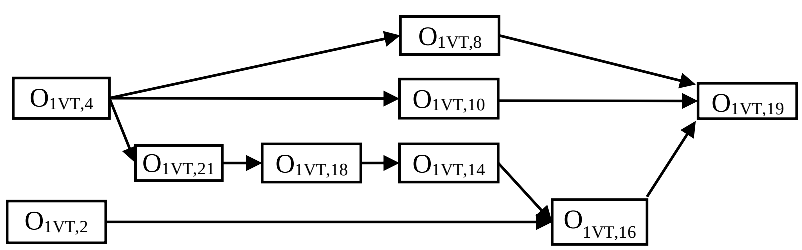
Node ID	Node description
$O_{1VT,17}$	Assemble turning head base to machine table before assembling other parts.
$O_{1VT,18}$	Assemble the motor in D_{30} to turning head base.
$O_{1VT,19}$	Align tail stock with head stock.
$O_{1VT,20}$	Relocate modules.
$O_{1VT,21}$	IF head stock with the motor in D_{30}, THEN relocate milling spindle.
$O_{1VT,22}$	IF head stock with a dedicated motor, THEN don't relocate milling spindle.



(a): Creation of a specific reconfiguration process graph.



(b): Creation of operation sequence graph.



(c): Simplification of the operation sequence graph.

Figure 4.6: Creation of a specific reconfiguration process.

Table 4.3: Parameters defined for the operation nodes given in Figure 4.6(c).

Node	Base time (min.)	Number of operators	Insertion complexity					Handling complexity		
			No. of tools	Alignment	Insertion resistance	Mechanical fastening process	Mechanical unfastening process	Specialized tools	Weight	Assistance
$O_{1VT,2}$	15	1	1	N/A	N/A	none	NB	involved	heavy	using two hands with assistance
$O_{1VT,4}$	30	2	2	N/A	N/A	none	NB	NV	heavy	using two hands
$O_{1VT,8}$	30	1	2	easy	resistance	ST	N/A	involved	heavy	using two hands with assistance
$O_{1VT,10}$	15	1	2	not easy	resistance	ST	N/A	N/A	heavy	using two hands
$O_{1VT,14}$	20	1	3	easy	resistance	ST	N/A	N/A	heavy	using two hands
$O_{1VT,16}$	20	1	3	not easy	resistance	ST	N/A	N/A	heavy	using two hands
$O_{1VT,18}$	30	1	2	not easy	NR	ST	N/A	N/A	heavy	using two hands
$O_{1VT,19}$	15	1	1	not easy	N/A	none	N/A	involved	heavy	using two hands
$O_{1VT,21}$	10	1	1	N/A	N/A	none	NB	NV	heavy	using two hands

ST: Screw tightening, NR: No resistance, NI: Not involved, NB: Nut and bolt.

The parameters related to the complexity were used as correction coefficients to calculate the actual required time from the required base time considering the complexity of this operation. Based on the studies by Samy and ElMaraghy (2010) and Soh, Ong, and Nee (2016), the factors influencing operation complexity of assembly and disassembly activities were classified into two categories considering insertion operations and handling operations. Typical attributes for insertion operations included number of required tool, difficulty in alignment (e.g., easy, not easy), insertion resistance (e.g., resistance, no resistance), mechanical fastening process (e.g., screw tightening, riveting) and mechanical unfastening process (e.g., nut and bolt, key). Typical attributes for handling operations included whether special tools are involved, weight (e.g., light, heavy) and assistance (e.g., using two hands, using two hands with assistance).

4.4.3 Evaluation to design solutions

Since eight design solution candidates were created, different reconfiguration processes were identified for each of these design solution candidates, and machine configurations and reconfiguration processes were further modeled by design parameters and operation parameters, evaluations to these different designs were conducted for achieving the optimal design.

Among the numerous design and operation parameters in the design nodes and operation nodes, only four design parameters related to major machine specifications were selected as variables to be decided in the design process. These four design variable parameters are:

- P_m: Power of the spindle motor (kw), $7.5 \le P_m \le 20$.
- L_x: Travel range of the table in x-direction (m), $0.5 \le L_x \le 1.7$.
- L_y: Travel range of the saddle in y-direction (m), $0.5 \le L_y \le 1.2$.
- L_z: Travel range of the spindle head in z-direction (m), $0.7 \le L_z \le 2$.

These design parameters were also considered to evaluate each of the design solutions. In addition, three new evaluation measures were considered in this case study:

- C_C: Total cost considering machine configurations (*$*).
- C_R: Total cost considering all the reconfiguration processes in the whole RMT life span ($).
- ξ: Stiffness rating of the RMT considering machining accuracy (*%*).

(1) Evaluation to the total cost considering machine configurations.

C_C is the total cost of components and modules for all the three machine configurations. C_C was calculated by:

$$C_C = C_{machine\ structure} + C_{drive\ systems} + C_{functional\ units} \tag{4.11}$$

where $C_{machine\ structure}$ was the cost for machine structure modules including modules with motions such as table, saddle and moving head, and modules without motions such as bed and column. $C_{drive\ systems}$ was the cost of the drive systems for translational motions in x, y, and z directions. $C_{functional\ units}$ was the cost of other machine modules for vertical milling, horizontal milling and turning. $C_{drive\ systems}$ was calculated by:

$$\begin{aligned} C_{drive\ systems} &= C_{drive\ system,\ x\ direction} + C_{drive\ system,\ y\ direction} + C_{drive\ system,\ z\ direction} \\ &= (C_{stepper\ motor,\ x\ direction} + C_{guidway,\ x\ direction}) + (C_{stepper\ motor,\ y\ direction} + C_{guidway,\ y\ direction}) \\ &\quad + (C_{stepper\ motor,\ z\ direction} + C_{guidway,\ z\ direction}) \end{aligned} \tag{4.12}$$

where $C_{stepper\ motor,\ i\ direction}$ was the cost of the stepper motor to drive the moving parts in i-th direction, and $C_{guidway,\ i\ direction}$ was the cost of the guideway for the moving parts in i-th direction.

From the product catalogues for WanTai stepper motors, NEMA 34 models with the driver and power supply systems, the relation between torque of the stepper motor, $T_{stepper\ motor}$ ($N{\cdot}m$), and its cost, $C_{stepper\ motor}$ *($)*, was developed by curve fitting as shown in Equation (4.13).

$$C_{steeper\ \ motor} = -0.033T^{\ 3}_{\ steeper\ motor} + 2.9T^{\ 2}_{\ steeper\ motor} - 37.1T_{\ steeper\ motor} + 329.3 \tag{4.13}$$

Torque in Equation (4.13) was calculated by:

$$T_{steeper\ \ motor} = J\frac{d\omega}{dt} + T_{static} = J\frac{a}{\left(p/2\pi\right)} + T_{static} \tag{4.14}$$

where J was the total moment of inertia with respect to the stepper motor's shaft, T_{static} was the total static torque, $d\omega/dt$ was the angular acceleration of the motor's shaft (rad/s^2), a was the maximum linear acceleration of the machine table (a =1.7 m/s^2), and p was the pitch of the feed screw (p = 0.01 m/rev) (Liang and Shih 2015). J was calculated by:

$$J = J_1 + J_2 + J_3 \tag{4.15}$$

where J_1 was the moment of inertia of the table and workpiece with respect to the leadscrew shaft ($kg{\cdot}m^2$), J_2 was the moment of inertia of the leadscrew ($kg{\cdot}m^2$), and J_3 was the inertia of motor's shaft itself ($J_3 = 2.373 \times 10^{-3}$ $kg{\cdot}m^2$). J_1 and J_2 were calculated by:

$$J_1 = \left(m_{table} + m_{workpiece}\right)\left(\frac{p}{2\pi}\right)^2 \tag{4.16}$$

$$J_2 = \frac{\pi}{32}\rho L r^4 \tag{4.17}$$

where m_{table} was the mass of the machine table (m_{table} = 200 kg), $m_{workpiece}$ was the maximum mass of workpiece ($m_{workpiece}$ = 300 kg), r *was* the radius of the feed screw (r =0.032 m), L was the length of the screw (i.e., L_x, L_y, or L_z), and ρ was the material density of the screw (ρ = 7861 kg/m^3). T_{static} was calculated by:

$$T_{static} = T_1 + T_2 + T_3 \tag{4.18}$$

where T_1, T_2, T_3 were the torques ($N{\cdot}m$) consumed to overcome the friction in guideway, to overcome the bearing friction, and to overcome the cutting force applied on the leadscrew shaft, respectively.

$$T_1 = \frac{p}{2\pi}\mu_g\left(w_{table} + w_{workpiece} + F_{perpendicular}\right) \tag{4.19}$$

$$T_2 = \mu_b \frac{r}{4}\left(F_{feed} + F_{preload}\right) \tag{4.20}$$

$$T_3 = \frac{p}{2\pi}F_{feed} \tag{4.21}$$

where μ_g was the friction coefficient on the guideway (μ_g =0.1), μ_b was the friction coefficient on the thrust bearings (μ_b =0.005), w_{table} and $w_{workpiece}$ were the weights of machine table and workpiece respectively (N), $F_{perpendicular}$ was the maximum cutting force perpendicular to the machining table ($F_{perpendicular}$ = 2500 N), F_{feed} was the maximum feed force (F_{feed} = 8000 N), and $F_{preload}$ was the preload ($F_{preload}$ = 10000 N).

The cost of the functional units was calculated by:

$$C_{functional\ units} = C_{spindle\ motor} + C_{accessories} \tag{4.22}$$

where $C_{spindle\ motor}$ was the cost of the spindle motor, $C_{accessories}$ was the cost of the functional components that are used for different machining operations including milling fixture, horizontal and vertical milling heads, 3-jaw chuck, tailstock, headstock, and tool fixture.

From the product catalogues for Siemens cast-iron series motors with *1800 rpm* and *60 Hz*, the relation between the motor power, P_m (kw), and the cost of spindle motor, $C_{spindle\ motor}$, was obtained by curve fitting as follows:

$$C_{spindle\ motor} = -0.046P_m^3 + 3.2566P_m^2 + 145.774P_m + 875.3822 \tag{4.23}$$

(2) Evaluation to the total cost considering all the reconfiguration processes in the life-span.

The total cost for all reconfiguration processes in the RMT life span was calculated by:

$$C_R = \frac{N}{6}\left[C_{iVT} + C_{iTV} + C_{iHT} + C_{iTH} + C_{iVH} + C_{iHV}\right] + C_{tools} \tag{4.24}$$

where C_{ijk} was the cost of the reconfiguration process to change from the *j*-th configuration to the *k*-th configuration for the *i*-th design candidate, *T*, *V* and *H* were the symbols for turning, vertical milling and horizontal milling configurations respectively, *N* was the expected number of reconfiguration processes throughout the RMT's life (N =100 based on the assumption that the RMT would be used for 15 years and six reconfiguration processes would be required in each year), and C_{tools} was the cost of the required tools for the reconfiguration processes (C_{tools} = $8000).

The cost of the reconfiguration process, C_{ijk}, was the total summation of all costs for the operations in the reconfiguration process. C_{ijk} was calculated by:

$$C_{ijk} = \sum_{o_{ijk}=1}^{B_{ijk}} \left(T_{o_{ijk}} \times C_l \times l_{o_{ijk}}\right),\ i = 1,2,...,p;\ j = 1,2,...,q;\ k = 1,2,...,q,\ j \neq k \tag{4.25}$$

where o_{ijk} was the o_{ijk}-*th* operation in the reconfiguration process R_{ijk} to change from the *j-th* configuration to the *k-th* configuration for the *i-th* design candidate, B_{ijk} was the total number of operations, C_l was the labor cost rate for one operator (C_l = 150 *$/hr*), $l_{o_{ijk}}$ was the number of required operators, and $T_{o_{ijk}}$ was the time of this o_{ijk}-*th* operation (*hr).*

The operation time in Equation (4.25) was calculated from the base time for the operation defined in Table 4.3 and its complexity coefficient using the complexity factors defined in Table 4.4. According to the studies by Samy and ElMaraghy (2010) and Soh, Ong, and Nee (2016), the assembly and disassembly operations can be classified into two categories: handling operations (e.g., acquiring, orienting, and moving the parts) and insertion/fastening operations (e.g., mating a part to another part or group of parts). For each type of operation, a difficulty factor, D_f, can be used to define its complexity. Typical assembly/disassembly operations and difficulty factors of these operations are defined in Table 4.4.

Table 4.4: Assembly/disassembly operations and difficulty factors of these operations (Samy and ElMaraghy 2010; Soh, Ong, and Nee 2016).

Category	Attribute	Description	Difficulty factor
Attributes for insertion operations ($D_{I,f}$)	Tool required	0 tool 1-3 tool >4 tools	0 0.6 1
	Alignment	Easy to align or position Not easy to align or position	0.86 1
	Insertion resistance	No resistance Resistance to insertion	0.87 1
	Mechanical fastening process	Bending Riveting Screw tightening Bulk plastic deformation	0.34 0.58 0.42 1
	Mechanical unfastening process	Screw/bolt standard head Screw/bolt special head Nut and bolt Interference fit Key	0.56 0.88 0.84 0.72 0.6
Attributes for handling operations ($D_{H,f}$)	Specialized tools	Not involved involved	0 1
	Weight	< 4.5 kg (light) > 4.5 kg	0.5 1
	Size	> 15 mm 6-15 mm < 6 mm	0.75 0.81 1
	Assistance	Using one hand Using one hand with grasping aids Using two hands Using two hands with assistance	0.34 1 0.75 0.57

The time of the o_{ijk}-th reconfiguration operation in Equation (4.25), $T_{o_{ijk}}$, considering handling and insertion difficulties was calculated by:

$$T_{o_{ijk}} = T_{Base_{o_{ijk}}} (D_{H_{o_{ijk}}} + D_{I_{o_{ijk}}}) \tag{4.26}$$

where $T_{Base_{ijk}}$ was the base time for the o_{ijk}-th reconfiguration operation defined in Table 4.3, and $D_{H_{o_{ijk}}}$ and $D_{I_{o_{ijk}}}$ were the handling complexity factor and insertion complexity factor respectively for the o_{ijk}-th reconfiguration operation defined in Table 4.4.

(3) Evaluation to the stiffness rating considering machining accuracy.

Due to various forces exerted to the machine tool structure and the workpiece such as cutting forces and gravity forces, the cutting tool deviated from its desired position during the machining process, leading to machining errors in the manufactured components. In this case study, stiffness of the machine tool (Shneor and Portman 2010; Portman et al. 2015) was selected to evaluate the accuracy performance. In this work, the machine stiffness rating (ξ) was defined by the ratio of the stiffness considering linear displacement of the moving head and the stiffness of the drive system. The stiffness measures of drive systems in x and y directions were used as references because they were integral parts of the machine tool and weak points of the serial-type mechanical system (Shneor and Portman 2010; Portman et al. 2015).

In this case study, only the stiffness rating for the vertical milling configuration was considered, since the chances to use this machine configuration were considered far more than the chances for the horizontal milling configuration and the turning configuration. Stiffness rating was obtained by:

$$\xi = \left(1 + \frac{\mu K_{act}}{K_m}\right)^{-1} \tag{4.27}$$

where ξ was the dimensionless measure representing the stiffness rating of the moving head, which was calculated from the stiffness of the translational drive systems (K_{act} = 2.5×10^7 *N/m*), the stiffness of the machine column (K_m = 2.5×10^8 *N/m*), and the dimensionless factor, μ, which was determined by the maximum active forces for the three machine configurations and structural dimensions of the machine tool. μ was calculated by:

$$\mu = \left(\frac{L_h^3 - 3l_p L_h\left(L_h - l_p\right) + 3L\left(L_h - 2l_p\right)\left(b_p\, F_z/F_y - b_G\, G_z/F_y\right)}{{L_m}^3}\right) \tag{4.28}$$

where F_z was the maximum cutting force perpendicular to the machining table, F_y was the maximum cutting force parallel to the machining table, and G_z was the maximum force exerted on the moving head by the influence of the drive system on z-direction. The ratios of the maximum active forces were selected as F_z/F_y = 0.5 and G_z/F_y = 2.5. Other relevant parameters include the horizontal distance between the vertical center of spindle and the vertical center of machine column (b_p= 650 *mm*), the horizontal distance between the vertical center of the moving head to hold the spindle and the vertical center of the machine column (b_G= 400 *mm*), the vertical distance between the horizontal center of spindle and the end of the tool (l_p= 350 *mm*), the vertical distance between machine bed and the horizontal center of the moving head to hold the spindle (L_h), and the mean height of the moving head (L_m=1250 *mm*).

4.4.4 Identification of the optimal design

In this case study, seven measures were selected to evaluate each design solution candidate. These evaluation measures include: P_m, L_x, L_y, L_z, C_C, C_R and ξ. Since these evaluation measures were in different units, they were converted into comparable evaluation indices representing degrees of satisfaction.

In this case study, various customers and engineers were asked to rate the satisfaction degrees with the different evaluation measures. For P_m, L_x, L_y, L_z and ξ, with the increases of their values, their satisfaction indices were also increased. For C_C and C_R, however, with the increases of their values, their satisfaction indices were decreased usually in linear manner. The least-square curve-fitting method was used to obtain the relations between the evaluation measures and the evaluation indices. The relations are summarized as follows:

$$I_{P_m} = -5.6 \times 10^{-6} P_m^3 - 0.0011 P_m^2 + 0.071 P_m - 0.0037 \tag{4.29}$$

$$I_{L_x} = -0.34 L_x^3 + 0.58 L_x^2 + 0.56 L_x - 0.03 \tag{4.30}$$

$$I_{L_y} = -0.695 L_y^3 + 1.031 L_y^2 + 0.61 L_y - 0.017 \tag{4.31}$$

$$I_{L_z} = -0.021 L_z^3 - 0.17 L_z^2 + 0.94 L_z - 0.032 \tag{4.32}$$

$$I_\xi = -0.7\xi^3 + 0.55\xi^2 + 1.2\xi - 0.05 \tag{4.33}$$

$$I_{C_C} = -4 \times 10^{-6} C_C + 1.7 \tag{4.34}$$

$$I_{C_R} = -4 \times 10^{-6} C_R + 1.7 \tag{4.35}$$

The multi-objective optimization problem was formulated by assigning weighting factors to the seven evaluation indices. These weighting factors were also determined from expert experience. The overall satisfaction index was defined as:

$$I = \frac{0.25I_{P_m}+0.25I_{L_x}+0.25I_{L_y}+0.25I_{L_z}+0.5I_{\xi}+0.5I_{C_C}+0.5I_{C_R}}{0.25+0.25+0.25+0.25+0.5+0.5+0.5} \tag{4.36}$$

For the design solution candidate given in Figure 4.6(b), the optimal values of the four design variable parameters were achieved using constrained numerical optimization:

- P_m =10.5 *kw.*
- L_x =1.5 *m.*
- L_y =1.1 *m.*
- L_z =1.3 *m.*

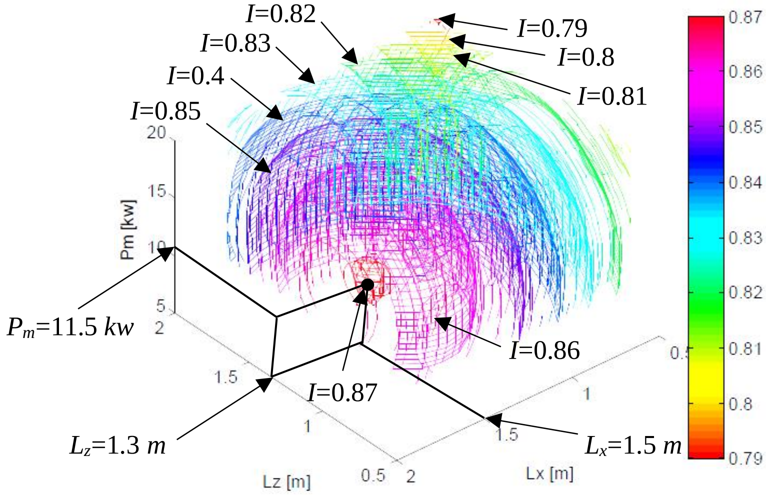
The overall evaluation index corresponding to this design was achieved as:

- I = 0.87.

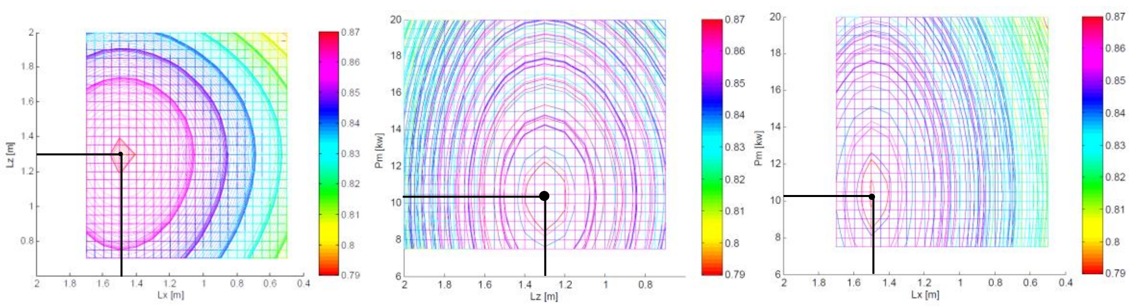
4.4.5 Discussions

The relations between different design parameters and the overall evaluation indices are illustrated in Figure 4.7. In the 2-D and 3-D plots of Figure 4.7, L_y was selected with its optimal value (i.e., Ly = 1.1 m). P_m, L_x, and L_z were selected based on their boundaries (i.e., $7.5 \leq P_m \leq 20$, $0.5 \leq L_x \leq 1.7$, $0.7 \leq L_z \leq 2$). In Figure 4.7(a), the different values for P_m, L_x, and L_z with the same overall evaluation index were used to construct a surface in MatLab. When the overall evaluation index, I, was selected as 0.79, 0.80, 0.81, 0.82, 0.83, 0.84, 0.85, 86 and 0.87, a total of nine surfaces were constructed as shown in Figure 4.7(a). From this plot, we can see the point (i.e., P_m, L_x, and L_z) with the maximum overall evaluation index, I, was matched with the result achieved through constrained numerical optimization. Three 2-D contour maps considering only 2 parameters out of the 3 parameters were created as given in Figure 4.7(b) to better show the optimization results.

The optimal parameter values achieved in Section 4.4.4 are only for the design solution candidate shown in Figure 4.3(b). In this design solution candidate, a mechanical spindle module with a removable motor was used for the drive system in the vertical milling machine configuration, and the same motor was also used for the horizontal milling and turning machine configurations. From the generic design AND-OR tree shown in Figure 4.3(a), a total of eight feasible design solutions were created. For each of these eight design solutions, parameter optimization was conducted to achieve the optimal parameter values and the corresponding overall evaluation index. From the eight design solutions, the one with the maximum overall evaluation index was selected as the optimal design solution. In this case study, the design solution given in Figure 4.3(b) was identified as the best design solution. When the number of design solutions is large, genetic programming has to be used to identify the optimal design solution effectively.



(a): The 3-D view of the overall satisfaction index contour map for L_y=1.1 *m*.



L_x=1.5 *m*; L_z=1.3 *m* L_z=1.3 *m*; P_m=10.5*Kw* L_x=1.5 *m*; P_m=10.5*Kw*

(b): The contour map views to get the optimal values of P_m, L_x and L_z.

Figure 4.7: The contour maps of the overall satisfaction index for the optimal design.

4.5 Summary

A new design approach is introduced in this research for identification of the optimal reconfigurable product considering both the product configurations and the reconfiguration processes. Characteristics of this new design approach are summarized as follows.

1. The tree and graph data structures are effective methods to model reconfigurable products considering both the product configurations and reconfiguration processes. A generic

design AND-OR tree is used to describe different design candidates, product configurations, and design parameters. A generic reconfiguration process AND-OR graph is used to describe reconfiguration process candidates and process parameters. A tree/graph-based search is used to create different design/process candidates.

2. The developed evaluation scheme is effective to evaluate a reconfigurable product considering both the product configurations and reconfiguration processes. This evaluation method evaluates a reconfigurable product from different perspectives through converting different evaluation measures described in different units into the same environment.

3. The developed optimization scheme is effective to identify the optimal design solution of a reconfigurable product considering product configurations and reconfiguration processes. The optimization scheme is used to identify optimal product configurations, reconfiguration processes, and parameters of product configurations and reconfiguration processes through multi-level and multi-objective optimization.

Chapter 5 An Efficient Optimization Approach for Design of Reconfigurable Products Considering Large Numbers of Design Candidates and Reconfiguration Process Candidates

In Chapter 4, a new method was developed for modeling, evaluation and optimization of a reconfigurable product considering both the product configurations and reconfiguration processes (Gadalla and Xue 2017c). Despite the effectiveness of this method, it required considerable computation effort when numbers of design candidates and reconfiguration process candidates were large.

A new method is developed in this research to improve the computation efficiency of the optimal reconfigurable product design method that was introduced in Chapter 4 (Gadalla and Xue 2017c). As discussed in Section 3.3, the design in the new approach is conducted in three steps.

(1) Modeling of design candidates and identification of importance weights for different design nodes.

Based on design requirements, generic design considering different candidates is modeled by a design AND-OR tree with design nodes and their parameters. The relations among nodes are classified into three categories: AND relation (AND), OR relation in design (OR-D), and OR relation in configuration (OR-C). Different design candidates are created from the AND-OR tree through tree-based search. For a node in the design tree, its importance related to other nodes with an AND or OR relation is modeled by a local weight W_L. The global weight of a design node W_G, representing the importance of this node related to other nodes in the whole design tree, is calculated based on its local weight and AND/OR relations in the design tree.

(2) Modeling of reconfiguration processes and identification of importance weights for different operation nodes.

For a reconfigurable product with multiple configurations, the generic reconfiguration process to change between two configurations is modeled by a generic process AND-OR graph with operation nodes and their parameters. The relations among nodes are classified into three categories: AND relation (AND), OR relation in operation (OR-O), and OR relation in design condition (OR-DC). Since sequential constraints among operations are also considered, the generic reconfiguration process is defined by a graph instead of a tree. Different reconfiguration process candidates are created from the AND-OR graph through graph-based search considering sequential constraints among operations. Same as the design tree, local weight W_L and global weight W_G of an operation node are used to describe its importance related to other nodes with an AND or OR relation and its importance related to other nodes in the whole reconfiguration process graph.

(3) Optimization to identify the best design candidate and its reconfiguration processes.

When the numbers of design candidates and reconfiguration process candidates are large, pruning is conducted based on a new modified beam search to cut off some nodes that are unlikely to lead to the optimal solution. At least one node has to be selected from the sub-nodes with an OR relation in design (OR-D) in a generic design AND-OR tree or sub-nodes with an OR relation in operation (OR-O) in a generic reconfiguration process AND-OR graph. Some of the remaining nodes are pruned based on the global importance weights of these nodes to reduce the search space. When the numbers of created design candidates and reconfiguration process candidates are still large after the pruning process, an initial ranking of the created candidates is carried out to evaluate these candidates using a simple heuristic function. The top-ranked candidates are selected for detailed evaluation to identify the optimal design solution with its product configurations, configuration parameters, reconfiguration processes, and process parameters through multi-level and multi-objective optimization.

5.1 Identification of Importance Weights for all Nodes in the Generic Design AND-OR Tree and Generic Reconfiguration Process AND-OR Graphs

As discussed in Chapter 4, a generic design AND-OR tree is used to model different design candidates. Since a design candidate of the reconfigurable product has multiple operation configurations, a generic reconfiguration process AND-OR graph is used to model different process candidates for reconfiguration between two operation configurations. A generic design AND-OR tree or a generic reconfiguration process AND-OR graph is described by a hierarchical data structure with a root node at the top of the tree/graph and leaf nodes at the bottom of the tree/graph (Xue, Cheing, and Gu 2006; Hong et al. 2008). Based on the distances from the nodes to the root node, all nodes in a tree/graph are arranged at different levels as shown in Figure 5.1. The root node is defined as level 0. Each node in the AND-OR tree/graph is described by its sequence number at the level and the level number. For example, $D_{1,i}$ is the first node at the *i-th* level.

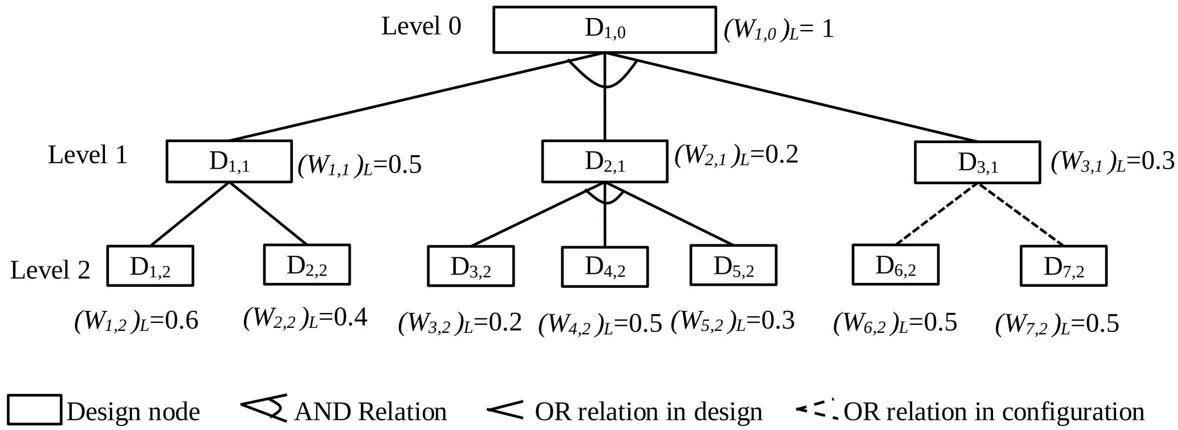


Figure 5.1: Generic design AND-OR tree with local weights.

For any node in the design tree or reconfiguration process graph, its importance related to other nodes with an AND or OR relation is modeled by a local weight W_L, and its importance related to other nodes in the whole design tree or reconfiguration process graph is modeled by a global weight W_G. Weight is defined by a value between 0 and 1, corresponding to the least important measure and the most important measure respectively.

5.1.1 Identification of local importance weights

A local weight, W_L, of a node represents the relative importance of the node with other nodes in the same group of sub-nodes in an AND or OR relation as summarized in Tables 3.2 and 3.3. For a group of nodes with an AND relation, the local weights of these nodes represent their contributions to the parent node. For example, since a car is composed of body, mechanical system, electrical system, and other modules, the local weights for these sub-systems can be assigned as 0.4, 0.3, 0.15, and 0.15 respectively. For a group of nodes with an OR relation in design for a design tree and a group of nodes with an OR relation in operation for a reconfiguration process graph, a local weight represents how good this node is by comparing it with other alternative nodes in the same group. For example, the material for a car body can be alloy or aluminum with local weights of 0.6 and 0.4 respectively. For the nodes with an OR relation in configuration for the design tree, since all these nodes should be selected for a design candidate for delivering different functions in different product operation configurations, the local weights of these nodes are their contributions to the parent node in different product operation configurations. For the nodes with an OR relation in design condition for the reconfiguration process graph, each node takes the local weight of its parent node since only one can be selected according to the selected design candidate.

Two methods are used to obtain the local weights of nodes with an AND/OR relation: direct method and indirect method.

In the direct weight assignment method, engineers and customers are asked to assign local weights directly based on their experiences and preferences. In addition, a rank order centroid (ROC) method can be used for listing the nodes in order from the most important one to the least important one, and then calculating the weights of these nodes using different formulas such as the rank sum formula, rank exponent formula, rank reciprocal formula, and rank order centroid formula (Buede 2009).

Indirect assessment of local weights can be conducted using the pairwise comparison method (Koczkodaj and Szybowski 2015). The pairwise comparison method is effective to obtain weights when the number of nodes is large and contradiction may be introduced due to the large number of the nodes to be compared. Pairwise comparison is conducted in two steps:

1. Create a $n \times n$ matrix to compare n nodes $D_1, D_2, \ldots, D_n$, and compare each pair of two different nodes to form a pairwise comparison matrix A.

$$A = \begin{bmatrix} 1 & a_{12} & \ldots & a_{1n} \\ a_{21} & \ldots & a_{ij} & \ldots \\ \ldots & a_{ji} = 1/a_{ij} & \ldots & \ldots \\ a_{n1} & \ldots & \ldots & 1 \end{bmatrix} \tag{5.1}$$

where a_{ij} is the comparison between the *i-th* node and the *j-th* node. a_{ij} can take any number from 1 to 9 (i.e., 1, 3, 5, 7, 9 are equal, moderate, strong, very strong, and extremely strong, and 2, 4, 6, 8 are intermediate values) and a_{ji} is a reciprocal for a_{ij}.

2. Find a set of priorities as the local weights $W_{D_1}, W_{D_2}, \ldots, W_{D_n}$:

   (1) Obtain sum of the elements in the *i-th* row: $S_i = \sum_{j=1}^{n} a_{ij}$, $(i=1,2,\ldots,n)$

   (2) Divide each S_i by the sum of all S_i: $W_{D_i} = \frac{S_i}{\sum_{i=1}^{n} S_i}$, $(i=1,2,\ldots,n)$

5.1.2 Identification of global importance weights

The global weight of a node in the design tree or reconfiguration process graph is used to model the importance of this node in the whole design tree or reconfiguration process graph. In this research, the global weight, W_G, of a node is calculated based on the hierarchical relations in the AND-OR tree/graph considering different types of relations in the AND-OR tree/graph.

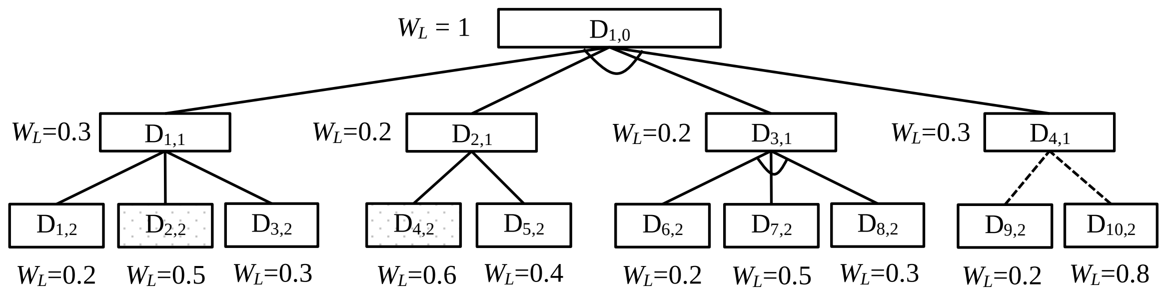
(1) Calculation of global weight for a sub-node with an AND relation (AND).

The global weight for the *m-th* node with an AND relation at the *i-th* level, $(W_{m,i})_G$, is calculated by multiplication of the local weight of this node, $(W_{m,i})_L$, and the global weight of its parent node in the *(i-1)-th* level, $(W_{m\text{-}parent,i-1})_G$, as shown in Figure 5.2(b). So, the sum of the global weights of the nodes with an AND relation is equal to the global weight of their parent node. For example, if the parent node $D_{3,1}$ shown in Figure 5.2(a) has three sub-nodes with an AND relation, the global weight of the parent node is 0.2, and the local weights of the 3 sub-nodes are 0.2, 0.5, and 0.3. Since the first sub-node contributes 0.2 to its parent's global weight, the global weight of this first sub-node can be calculated by multiplying its local weight of 0.2 and the parent's global weight of 0.2 to get 0.2×0.2 = 0.04. The global weights for the other two sub-nodes are calculated as 0.1 and 0.06 respectively. The global weight is calculated by:

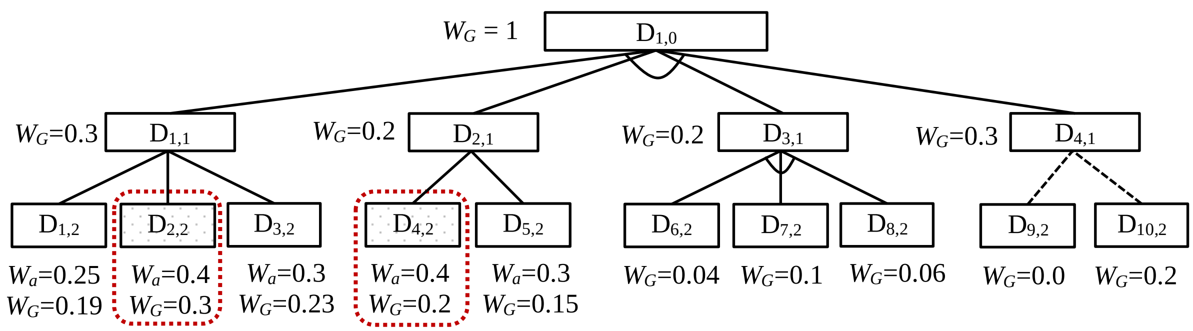
$$\left(W_{m,i}\right)_G = \left(W_{m,i}\right)_L \times \left(W_{m-parent,i-1}\right)_G \tag{5.2}$$

(2) Calculation of global weight for a sub-node with an OR-relation in design (OR-D) for the design tree and a sub-node with an OR-relation in operation (OR-O) for the reconfiguration process graph.

For the sub-nodes with an OR relation in design (OR-D) for the design tree and OR relation in operation (OR-O) for the reconfiguration process graph, since only one sub-node can be selected from these sub-nodes to create a candidate, this sub-node should take the same global weight as its parent node when this sub-node is selected. However, since several alternative sub-nodes with an OR relation in design (OR-D) for the design tree and OR relation in operation (OR-O) for the reconfiguration process graph are considered, both the local weight of the sub-node and the global weight of its parent node should be considered to determine the global weight of the sub-node.



(a): Identification of local weights in a design AND-OR tree.



(b): Calculation of global weights in a design AND-OR tree.

Node	***$D_{1,2}$***	***$D_{2,2}$***	***$D_{3,2}$***	***$D_{4,2}$***	***$D_{5,2}$***
Relative importance *(R)*	(0.25/0.4)= 0.63	1	(0.3/0.4)= 0.75	1	(0.3/0.4)= 0.75
Global weight $(W)_G$	0.63×0.3= 0.19	1×0.3= 0.3	0.75×0.3= 0.23	1×0.2= 0.2	0.75×0.2= 0.15

(c): Calculation of global weights for nodes with OR relations in design.

Design node AND Relation OR relation in design OR relation in configuration

Figure 5.2: Calculation of global weights.

In this work, calculation of the global weight for a sub-node with an OR relation in design (OR-D) for the design tree and OR relation in operation (OR-O) for the reconfiguration process graph is conducted through the following steps. Since some sub-nodes can be pruned by the new modified beam search, only the remaining sub-nodes are considered for the subsequent calculation.

1. Calculate the average weight of the *m-th* node at the *i-th* level, $(W_{m,i})_a$, using the node's local weight, $(W_{m,i})_L$ (Figure 5.2(a)), and its parent's global weight, $(W_{m\text{-}parent,i-1})_G$ as shown in Figure 5.2(b).

$$\left(W_{m,i}\right)_a = \frac{\left(W_{m,i}\right)_L + \left(W_{m-parent,i-1}\right)_G}{2} \tag{5.3}$$

2. Choose one sub-node that has the highest average weight in each group (e.g., nodes $D_{2,2}$ and $D_{4,2}$ in Figure 5.2(b)), and assign the global weight using the global weight of its parent.

3. Calculate the relative importance, $R_{m,i}$, of each node to the node with the highest average weight through dividing the average weight of the m-th node, $(W_{m,i})_a$, by the average weight of the node with the highest average weight, $(W_{max,i})_a$ as shown in Figure 5.2(c).

$$R_{m,i} = \frac{(W_{m,i})_a}{(W_{max,i})_a} \tag{5.4}$$

4. Calculate the global weight for the *m-th* node by multiplying its relative importance $R_{m,i}$ by the global weight of its parent node as shown in Figure 5.2(c).

$$(W_{m,i})_G = R_{m,i} \times (W_{m-parent,i-1})_G \tag{5.5}$$

(3) Calculation of global weight for a sub-node with an OR relation in configuration (OR-C) for the design tree.

For the sub-nodes with an OR relation in configuration for the design tree, since all these sub-nodes should be selected for creating a design candidate to deliver different functions through different product configurations, the global weight for each sub-node is calculated by Equation (5.2).

(4) Calculation of global weight for a sub-node with an OR relation in design condition (OR-DC) for the reconfiguration process graph.

For the sub-nodes with an OR relation in design condition (OR-DC) for the reconfiguration process graph, since only one sub-node is selected based on the selected design candidate, the global weight for the sub-node is assigned using the global weight of its parent node.

5.2 Pruning of Nodes in the AND-OR Tree and Graphs for Reducing the Numbers of Design Candidates and Reconfiguration Process Candidates

When large numbers of design candidates and reconfiguration process candidates can be created due to the OR relations in design (OR-D) for the design tree and OR relations in operation (OR-O) for the reconfiguration process graphs, computation efficiency has to be improved, such that the optimal solution can be identified within the given time period. In this research, a pruning method is developed to remove the nodes in the design tree and reconfiguration process graphs that are unlikely to lead to the optimal solutions. Since only nodes with OR relations in design (OR-D) for the design tree and OR relations in operation (OR-O) for the reconfiguration process graphs provide alternative solutions for design candidates and reconfiguration process candidates, only these types of nodes can be removed in the pruning process. The global weights achieved using the method given in Section 5.1 can be used to evaluate the nodes in the design tree and reconfiguration process graphs. The pruning method is based on a new modified beam search.

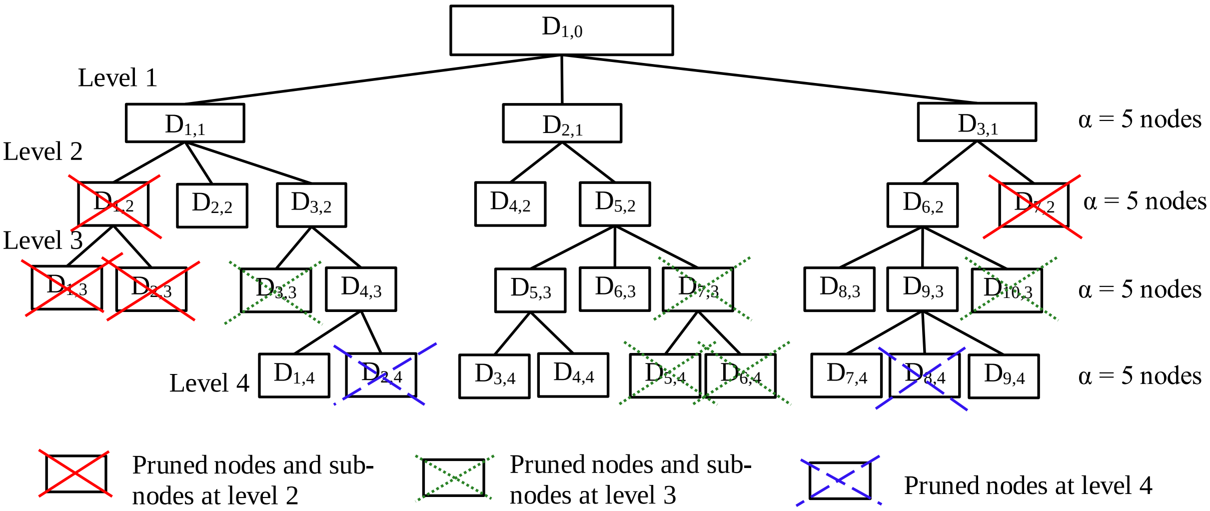
5.2.1 Modified beam search

In beam search (Edelkamp and Schroedl 2011), the total number of nodes at each level of the search tree has to be smaller than a pre-defined threshold (α) as shown in Figure 5.3(a). When more nodes are created, heuristic function is then used to evaluate these nodes. The nodes with poor evaluation measures and their sub-trees are pruned. In this research, a new modified beam search method is introduced to remove the nodes with OR relations in design for the design tree and nodes with OR relations in operation for the reconfiguration process graphs that are unlikely to achieve the optimal solutions based on global weights of these nodes.

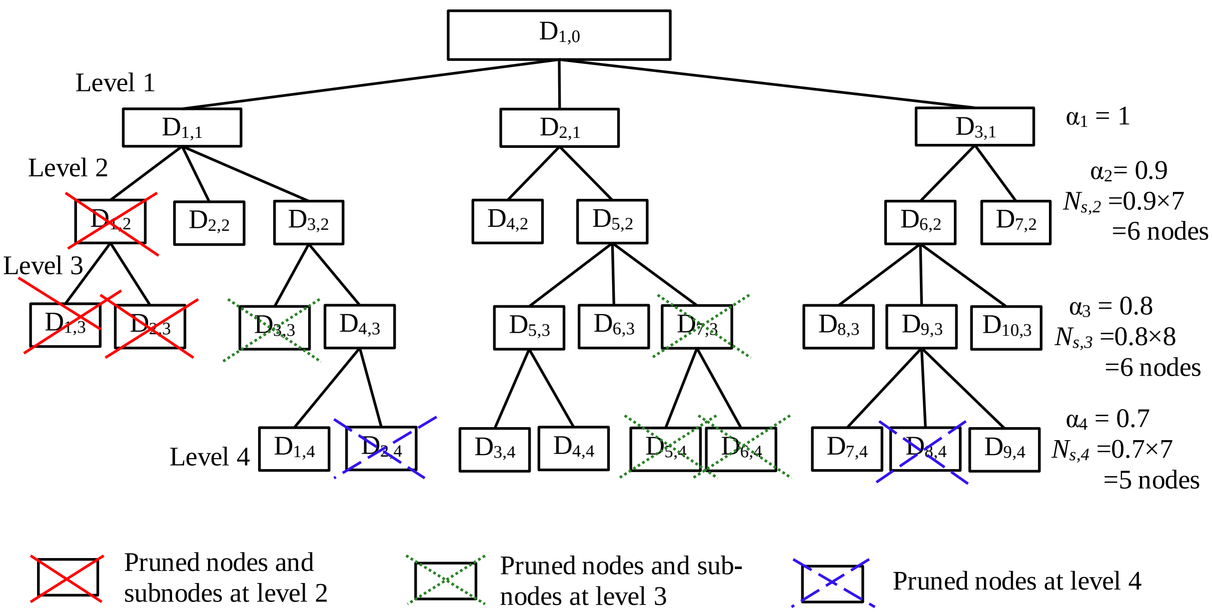
In this research, the threshold for the *i*-th level of the tree/graph is defined by a beam width ratio α_i representing the percentage of the alternative nodes to be selected as shown in Figure 5.3(b). First, all the nodes with OR relations in design for the design tree or OR relations in operation for the reconfiguration process graph are selected. Then, for each group of relations, the node with the highest global weight should be selected. All the remaining nodes are sorted based on their global weights. The number of nodes to be selected from these remaining nodes, $N_{s,i}$, is determined by:

$$N_{s,i} = \alpha_i \times N_{OR\text{-}relation,I} \tag{5.6}$$

where $N_{OR\text{-}relation,i}$ is the total number of remaining nodes with OR relations in design for the design tree or nodes with OR relations in operation for the reconfiguration process graph at the *i-th* level.



(a): Standard beam search.



$N_{s,i}$: Number of selected nodes at the *i*-th level.

(b): Modified beam search.

Figure 5.3: Standard and modified beam search methods.

5.2.2 Pruning of nodes with OR relations in design for the design tree

Pruning of nodes with OR relations in design for the design tree is conducted through the following steps:

1. First, choose the root node.
2. For the nodes at the first level with only AND relations

   2.1 Select the first level as the current level.

   2.2 Select all the nodes at this level, and then calculate their global weights.

3. Select the next level as the current level.

   3.1 For all nodes with AND relations and all nodes with OR relations in configuration, select all these nodes and then calculate their global weights.

   3.2 For all nodes with OR relations in design, decide what nodes to be selected and what nodes to be cut off using the following steps:

   3.2.1 Calculate the average weights of these nodes using Equation (5.3).

   3.2.2 Select one node that has the highest average weight in each group of nodes with an OR relation in design, because at least one node should be selected to generate one solution.

   3.2.3 Rank all the remaining nodes with OR relations in design at this level from the one with the highest average weight to the one with the lowest average weight.

   3.2.4 Calculate the number of nodes to be selected from these remaining nodes, $N_{selected,i}$, using Equation (5.6). Select the top $N_{selected,i}$ nodes from these remaining nodes, and cut off other remaining nodes and their sub-trees.

   3.2.5 Calculate the global weights for all the selected nodes with OR relations to be used at this level using Equations (5.4-5.5).

4. If the current level is the bottom level, stop. Otherwise, go to Step 3.

5.2.3 Pruning of nodes with OR relations in operation for the reconfiguration process graph

The method for pruning of nodes with OR relations in operation for the reconfiguration process graph is developed based on the method given in Section 5.2.2 through the following modifications.

- For nodes with an OR relation in design condition, select only one node by matching the IF condition with the created design candidate, and then assign a global weight for this node using the global weight of its parent node.
- For nodes with an AND relation, select all these nodes and then calculate their global weights.
- For all nodes with OR relations in operation at a certain level of the graph, decide what nodes to be selected and what nodes to be cut off using the Steps 3.2.1-3.2.5 given in Section 5.2.2.

5.3 Identification of the Optimal Design Solution

Although the pruning method can lead to reduced spaces for design candidates and reconfiguration process candidates, it is still impossible to evaluate all the solutions to achieve the optimal one. In this research, a heuristic function is employed to rank these solutions further. The top-ranked solutions are subsequently selected and evaluated with the sophisticated evaluation functions to achieve the optimal one through the multi-level and multi-objective optimization method (Gadalla and Xue 2017c).

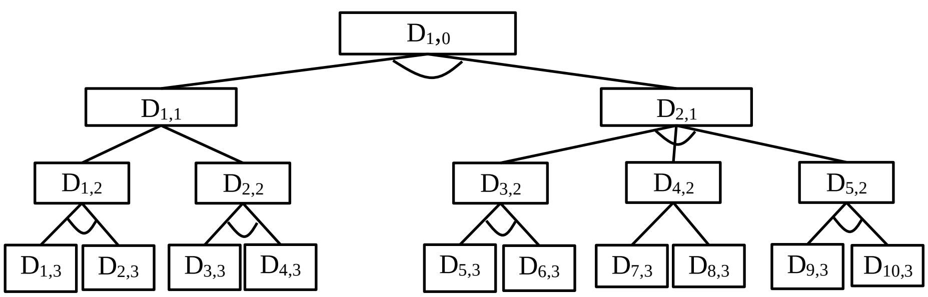
5.3.1 Initial ranking of different candidates using a heuristic function

A specific design candidate is created from the pruned generic design AND-OR tree through tree-based search, and a specific reconfiguration process candidate is created from the pruned generic reconfiguration process AND-OR graph though graph-based search (Hong et al. 2010; Gadalla and Xue 2017c). A specific design candidate is modeled by a tree with AND relations and OR relations in configuration, and a specific reconfiguration process candidate is modeled by a graph with only AND relations.

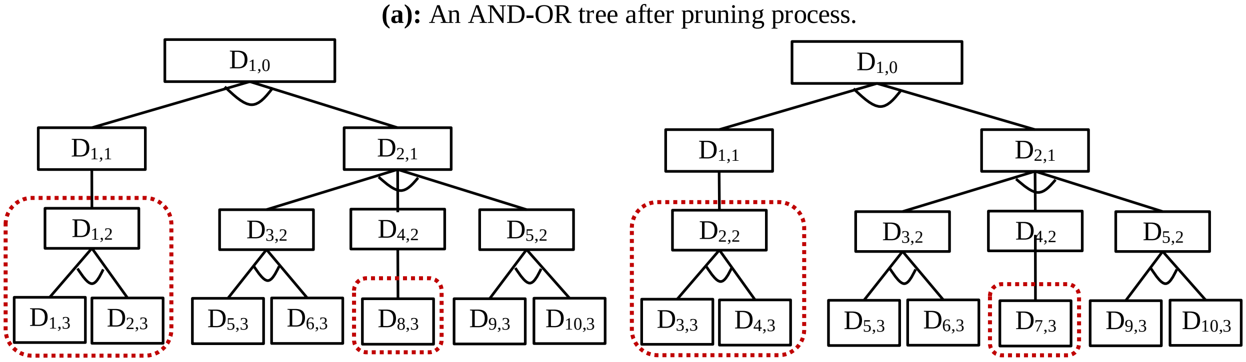
For the generated specific design or process candidates, some nodes are used by all these candidates and some nodes are used only by some of these candidates. To improve the computation efficiency, only the nodes that are not shared by all candidates are considered for the ranking process. In this work, the sum of the global weights of these nodes in a candidate is used as the heuristic function to evaluate this candidate. For the two design candidates shown in Figure 5.4(b), $D_{5,3}$, $D_{6,3}$, $D_{9,3}$, and $D_{10,3}$ are shared by all these two candidates, and $D_{1,3}$, $D_{2,3}$, and $D_{8,3}$ are used for the first design candidate and $D_{3,3}$, $D_{4,3}$, and $D_{7,3}$ are used for the second design candidate. The global weights of the different nodes in these candidates are used to rank these two candidates.

The top *n* candidates are identified from the pruned tree/graph based on the heuristic function. In this research, two approaches are used for identification of the top candidates.

- Exhaustive approach: to create all the candidates and evaluate all these candidates using the heuristic function when the search space is not large.
- Evolutionary approach: to create *n* candidates randomly as the first population and evolve the populations using genetic programming (Hong et al. 2010) when the search space is large.



(a): An AND-OR tree after pruning process.



(b): Two design candidates created from the pruned generic AND-OR tree.

Figure 5.4: Selection of nodes for ranking of design candidates.

5.3.2 Identification of the final optimal design solution

For the selected design candidates and reconfiguration process candidates through ranking with heuristic functions using the method given in Section 5.3.1, specific solutions can be created. Each solution is modeled by a design candidate and reconfiguration processes to change between product configurations of this design candidate. A specific solution is evaluated by the selected evaluation measures.

Since the design candidate and its reconfiguration processes are usually evaluated by different evaluation measures that are described in different units, conversion of these evaluation measures into comparable evaluation indices is then conducted. In this work, the method developed by Xue, Yang, and Tu (2006) is used to convert the evaluation measures into comparable satisfaction indices using their non-linear relationships. These evaluation indices take values between 0 and 1, representing the lowest satisfaction level and the highest satisfaction level respectively.

Suppose that the i-th design solution, S_i $(i=1, 2, \ldots, p)$, can be evaluated by m evaluation measures, F_j $(j=1, 2, \ldots, m)$, where p is the number of design solutions and m is the number of evaluation measures:

$$F_j = f_j(S_i), \quad i = 1, 2, \ldots, p; \; j = 1, 2, \ldots, m \tag{5.7}$$

Each evaluation measure, F_j, is converted into a satisfaction index, I_j, by:

$$I_j(S_i) = g_j(F_j) = g_j[f_j(S_i)], \; i = 1, 2, \ldots, p; \; j = 1, 2, \ldots, m \tag{5.8}$$

A polynomial function can be used to define the non-linear relationship between the evaluation measure and the evaluation index. The least-square curve-fitting method is usually used to obtain the polynomial function based on the collected data of different evaluation measures and their corresponding satisfaction indices. The overall evaluation index, $I(S_i)$, considering all evaluation indices is defined by:

$$I(S_i) = \frac{\sum_{j=1}^{m}\left(W_j I_j(S_i)\right)}{m\sum_{j=1}^{m} W_j} \tag{5.9}$$

where W_1, W_2, . . ., W_m are the weighting factors of the m evaluation indices representing their importance levels. Values of these weighting factors are obtained through a pairwise comparison method.

In this research, the optimal design solution is also identified by two approaches.

- Exhaustive approach: to evaluate all the top *n* solutions using Equation (5.9) to identify the optimal one when the *n* is not large.
- Evolutionary approach: to create the first population of solutions from the top *n* solutions and evolve the populations using genetic programming (Hong et al. 2010) based on the optimization objective function given in Equation (5.9) when the *n* is large.

When a design solution is modeled by design and process variable parameters, numerical optimization can be conducted to achieve the optimal values of these design and process variable parameters. Suppose the i-th design solution, S_i $(i=1,\ 2,\ \ldots,\ p)$, is modeled by a set of parameters, $\boldsymbol{X_i}$, including design parameters and process parameters, the parameter optimization problem is defined by:

$$\begin{aligned} &\max_{w.r.t.\boldsymbol{X}_i} I(\boldsymbol{X}_i) \\ &\qquad s.t.\quad \boldsymbol{X}_i^{(L)} \le \boldsymbol{X}_i \le \boldsymbol{X}_i^{(U)},\ \ i=1,2,\ldots,p \\ &\qquad\qquad h_{ik}(\boldsymbol{X}_i) \ge 0,\ \ i=1,2,\ldots,p;k=1,2,\ldots \\ &\qquad\qquad g_{ik}(\boldsymbol{X}_i) = 0,\ \ i=1,2,\ldots,p;k=1,2,\ldots \end{aligned} \tag{5.10}$$

The optimization through both the tree/graph-based search and the numerical search considering various evaluation measures is also called multilevel and multi-objective optimization (Gadalla and Xue 2017c).

5.4 A Case Study

The problem of this case study was to identify the optimal design solution of a reconfigurable bicycle considering large numbers of design candidates and reconfiguration process candidates using the newly developed efficient optimal reconfigurable product design method.

This reconfigurable bicycle was designed to operate as mountain bicycle, snow bicycle, and electrical bicycle to reduce the cost to buy three different types of bicycles. Figure 5.5 shows the three configurations of this reconfigurable bicycle. Six reconfiguration processes are also required to change between the 3 configurations.

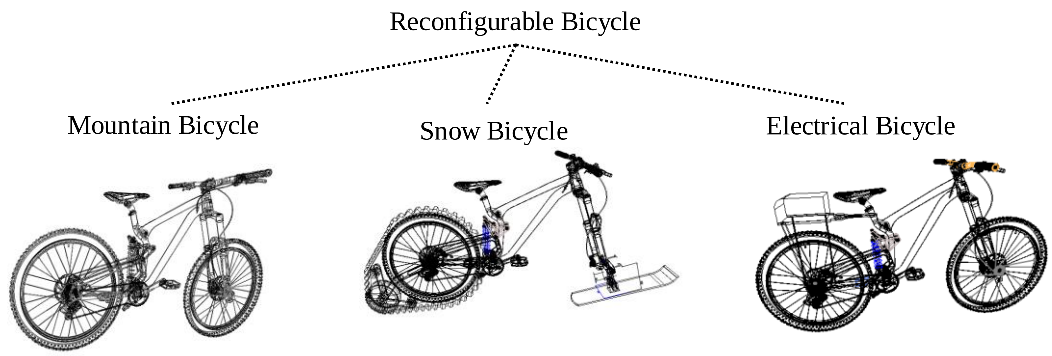


Figure 5.5: A reconfigurable bicycle with three configurations.

5.4.1 Modeling of different design candidates and reconfiguration process candidates with weights of nodes

A generic design AND-OR tree was used to model all design candidates. Figure 5.6 shows the major nodes of this design AND-OR tree. The local weights were determined using either the direct assignment method when the number of nodes in a group was small or the pairwise comparison method when the number of nodes in a group was large. From the generic design AND-OR tree, a large number of specific design candidates were created.

A generic AND-OR graph was used to model all the reconfiguration process candidates to change between two product configurations. Figure 5.7 shows the generic reconfiguration process from electrical bicycle configuration to snow bicycle configuration. Descriptions of the nodes in Figure 5.7 are given in Table 5.1. From the generic reconfiguration process AND-OR graph, a large number of specific reconfiguration process candidates were created.

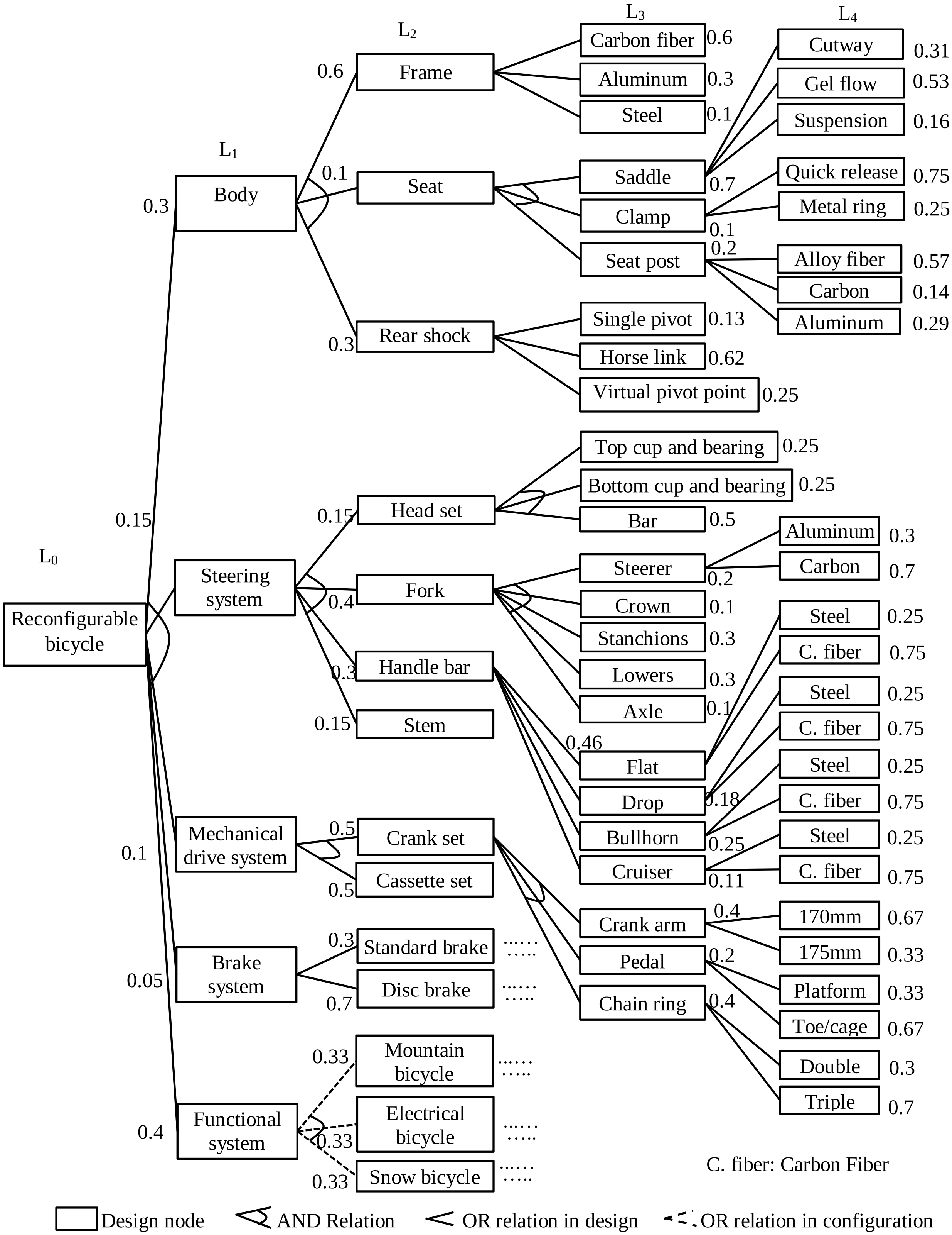


Figure 5.6: Generic design AND-OR tree for modeling a reconfigurable bicycle.

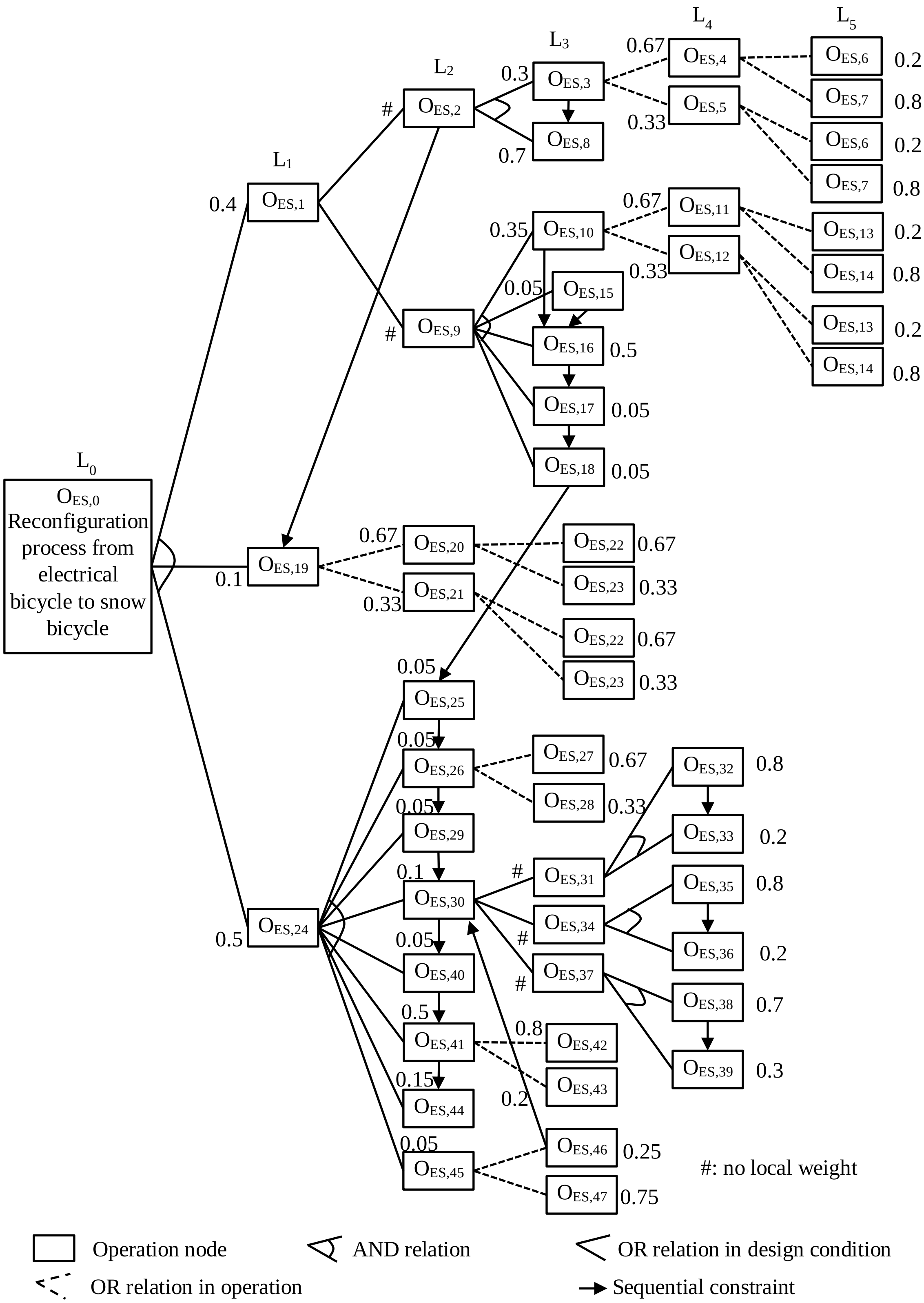


Figure 5.7: Generic AND-OR graph for modeling a reconfiguration process to change from electrical bicycle configuration to snow bicycle configuration.

Table 5.1: Descriptions of nodes in the AND-OR graph given in Figure 5.7.

Node	Description	Node	Description
$O_{ES,0}$	Reconfiguration process from electrical bicycle to snow bicycle	$O_{ES,24}$	Add a snow track system
$O_{ES,1}$	Remove the drive system	$O_{ES,25}$	Remove the rear wheel
$O_{ES,2}$	IF the system is front drive, THEN remove the front drive system	$O_{ES,26}$	Assemble the snow cover on the rear wheel to increase the friction with the track
$O_{ES,3}$	Remove motor control	$O_{ES,27}$	Install using special tools
$O_{ES,4}$	Detach the control cables and then remove the controller from the handlebar	$O_{ES,28}$	Install using standard tools
$O_{ES,5}$	Remove the controller from the handlebar and then detach the control cables	$O_{ES,29}$	Add the snow track
$O_{ES,6}$	Open the quick release or the metal rings to detach cables	$O_{ES,30}$	Add the chain to drive the left side track
$O_{ES,7}$	Cut the plastic straps to detach cables	$O_{ES,31}$	IF it is rear drive, THEN relocate the brake caliber to the lower position and add the chain
$O_{ES,8}$	Remove the front drive system	$O_{ES,32}$	Relocate the brake caliper to the lower place
$O_{ES,9}$	IF the system is rear drive, THEN remove the rear drive system	$O_{ES,33}$	Add the chain to drive the left side track
$O_{ES,10}$	Remove motor control	$O_{ES,34}$	IF it is front drive with rear disc brake system, THEN relocate the brake caliber to the lower position and add the chain
$O_{ES,11}$	Detach the control cables and then remove the controller from the handlebar	$O_{ES,35}$	Relocate the brake caliper to the lower place
$O_{ES,12}$	Remove the controller from the handlebar and then detach the control cables	$O_{ES,36}$	Add the chain
$O_{ES,13}$	Open the quick release or the metal rings to detach cables	$O_{ES,37}$	IF it is front drive with rear standard brake system, THEN assemble the rotary disc and then add the chain
$O_{ES,14}$	Cut the plastic straps to detach cables	$O_{ES,38}$	Install the rotary disc for the engine drive chain
$O_{ES,15}$	Remove the cover for the engine drive chain	$O_{ES,39}$	Add the chain
$O_{ES,16}$	Remove the rear drive system	$O_{ES,40}$	Close the quick release of the rear wheel
$O_{ES,17}$	Open the quick release of the rear wheel	$O_{ES,41}$	Install the main support system with idler wheel for the main track
$O_{ES,18}$	Remove the engine drive chain	$O_{ES,42}$	Install using special tools
$O_{ES,19}$	Add a front ski	$O_{ES,43}$	Install using standard tools
$O_{ES,20}$	Assemble ski to connecting bar and then install it to the bicycle	$O_{ES,44}$	Install the two side-track systems
$O_{ES,21}$	Install the connecting bar to the bicycle first and then assemble the ski	$O_{ES,45}$	Connect the rear brake system
$O_{ES,22}$	Install using special tools	$O_{ES,46}$	Assemble the rear brake pads
$O_{ES,23}$	Install using standard tools	$O_{ES,47}$	Connect the wire to the brake lever

5.4.2 Beam search to reduce numbers of design candidates and reconfiguration process candidates through pruning

Since the numbers of the created design candidates and reconfiguration process candidates were large, the new modified beam search was then employed to reduce the search spaces by cutting off the nodes that were unlikely to lead to the optimal candidates. In this case study, first, the global weights were calculated from the local weights for each tree/graph. The beam width ratios were then specified to different levels of the tree/graph. The methods introduced in Sections 5.1 and 5.2 were subsequently used to prune the tree and graphs. Parameters of the beam width ratios were selected considering different computation efficiency requirements.

5.4.3 Initial ranking of design candidates and reconfiguration process candidates after the pruning process

The number of the design candidates created from the pruned design AND-OR tree was still large. In this case study, an initial ranking was conducted to evaluate all the design candidates created from the pruned design AND-OR tree using the method introduced in Section 5.3.1. In the initial ranking process, only the global weights of those nodes with OR relations in design were considered. For the reconfigurable bicycle, the nodes for the frame, rear shock, saddle, seat post, steerer, handlebar, crank arm, pedal, chaining, brake system, and the drive for the electric bicycle were the nodes with OR relations in design. The top 10 design candidates were selected based on the initial ranking for the subsequent detailed evaluations to identify the optimal design candidate.

After the pruning of the AND-OR graphs for the 6 reconfiguration processes, the search space to identify the reconfiguration process candidates for each of these 6 reconfiguration processes was also reduced. Among these 6 reconfiguration processes, reconfiguration from electrical bicycle to mountain bicycle (EM) and reconfiguration from snow bicycle to mountain bicycle (SM) only had single process candidates. The reconfiguration from mountain bicycle to electrical bicycle (ME) had 5 process candidates, the reconfiguration from mountain bicycle to snow bicycle (MS) had 3 process candidates, the reconfiguration from electrical bicycle to snow bicycle (ES) had 2 process candidates, and the reconfiguration from snow bicycle to electrical bicycle (SE) had 2 process candidates. A complete solution was composed of 6 reconfiguration processes to change considering the 3 product configurations, where each reconfiguration process was selected from the feasible process candidates. Among all the solutions, the top 5 solutions were selected based on the initial ranking for the subsequent detailed evaluations to identify the optimal design solution.

5.4.4 Evaluation of design solutions

A complete design solution was composed of the specific design candidate and the 6 specific reconfiguration process candidates for this design candidate. In this case study, the top 10 design candidates, and the top 5 reconfiguration process solutions (i.e., each with 6 specific reconfiguration process candidates) were considered for constructing the complete design solutions.

In this case study, a complete design solution was evaluated by 3 measures: the total cost considering product configurations and reconfiguration processes, the reliability considering product components, and the reputation considering product components.

(1) Total cost.

The total cost of the *i*-th reconfigurable product design candidate, C_{iT}, was calculated from the configuration cost, C_{iC}, and the reconfiguration process cost, C_{iR}.

$$C_{iT} = C_{iC} + C_{iR} \tag{5.11}$$

The configuration cost was calculated by adding the costs of all components in the reconfigurable product. The reconfiguration process cost was calculated by adding all the costs for the 6 reconfiguration processes during the whole product lifespan. For this case study, it was assumed that the reconfigurable bicycle needed 14 reconfiguration processes (60% for ME and EM, 25% for MS and SM, and 15% for SE and ES) each year and its lifespan was 5 years. During the 5 years, the total number of ME and EM reconfiguration processes, $N_{ME\&EM}$, was 42, the total number of MS and SM reconfiguration processes, $N_{MS\&SM}$, was 18, and the total number of SE and ES reconfiguration processes, $N_{ES\&SE}$, was 10. The total reconfiguration process cost was calculated by:

$$C_{iR} = \left[\frac{N_{ME\&EM}}{2} \times \left(C_{iME} + C_{iEM}\right)\right] + \left[\frac{N_{MS\&SM}}{2} \times \left(C_{iMS} + C_{iSM}\right)\right] + \left[\frac{N_{ES\&SE}}{2} \times \left(C_{iES} + C_{iSE}\right)\right] + C_{tools} \tag{5.12}$$

where C_{tools} was the cost of all tools required for all reconfiguration processes (C_{tools} = \$250), and C_{ijk} was the cost of reconfiguration process from the *j-th* configuration to the *k-th* configuration of the *i-th* design candidate. C_{ijk} was calculated by adding all costs for all operations in the reconfiguration process (Gadalla and Xue 2017c):

$$C_{ijk} = \sum_{o_{ijk}=1}^{B_{ijk}} \left(T_{o_{ijk}} \times C_l \times l_{o_{ijk}} \right), \; i = 1, 2, ..., p; \; j = 1, 2, 3; \; k = 1, 2, 3, \; j \neq k \tag{5.13}$$

$$T_{o_{ijk}} = T_{Base_{o_{ijk}}} (D_{H_{o_{ijk}}} + D_{I_{o_{ijk}}}) \tag{5.14}$$

where:

o_{ijk} was the o_{ijk}-*th* operation to change from the *j-th* configuration to the *k-th* configuration for the *i-th* design candidate,

B_{ijk} was the total number of operations,

C_l was the labor cost rate for one operator (C_l = \$20/*hr*), assuming that an operator executed the reconfiguration process,

$l_{o_{ijk}}$ was the number of required operators for the o_{ijk}-*th* operation,

$T_{o_{ijk}}$ was the effective time required to finish the o_{ijk}-*th* operation (*hr),*

$T_{Base_{o_{ijk}}}$ was the base time required to finish the o_{ijk}-*th* operation (*hr)* without considering any complexities of handling and insertion of components through the o_{ijk}-*th* reconfiguration process,

$D_{H_{o_{ijk}}}$ *and* $D_{I_{o_{ijk}}}$ were the handling and insertion complexity factors for the o_{ijk}-th reconfiguration process (Samy and ElMaraghy 2010; Soh, Ong, and Nee 2016).

(2) Reliability considering components.

The reliability of the *i-th* design candidate, η_i, was calculated by getting the average of the reliabilities of all components (nodes) in this design candidates:

$$\eta_i = \sum_{D_{m_i=1}}^{M_i} \left(\eta_{D_{m_i}}\right) \times \left(W_{D_{m_i}}\right)_G, i = 1, 2, \dots, p \tag{5.15}$$

where $\eta_{D_{m_i}}$ and $\left(W_{D_{m_i}}\right)_G$ represented the reliability and global weight of the *m-th* node in the *i-th* design candidate, and M_i was the total number of nodes in the *i-th* design candidate.

(3) Reputation considering components.

The reputation of the *i-th* design candidate, γ_i, was calculated by getting the average of the reputations of all components (nodes) in this design candidates. In this case study, reputation rankings were classified into four levels: 4 for excellent reputation, 3 for good reputation, 2 for acceptable reputation, and 1 for poor reputation. The reputation of the design candidate, γ_i, was calculated by:

$$\gamma_i = \sum_{D_{m_i}=1}^{M_i} \left(\gamma_{D_{m_i}}\right) \times \left(W_{D_{m_i}}\right)_G, \ \ i = 1,2, \dots, p \qquad (5.16)$$

where $\gamma_{D_{m_i}}$ represented the reputation of the *m-th* node in the *i-th* design candidate, M_i was the total number of nodes in the *i-th* design candidate, and $\left(W_{D_{m_i}}\right)_G$ was the global weight of the *m-th* node.

5.4.5 Identification of the optimal design solution

The reconfigurable product in this case study was evaluated by three measures: total cost, reliability, and reputation. To compare the different evaluation aspects, these evaluation measures with different units were converted into comparable satisfaction indices. First, the data of different satisfaction levels for different evaluation measure values were collected from the customers and engineers. Then, the relations between evaluation measures and their evaluation indices were obtained by using the least-square curve fitting method. In this case study, with the increase of the values for η and γ, their satisfaction indices I_η and I_γ were also increased. With the increase of the value for C_T, however, its satisfaction index I_{C_T} was decreased. These non-linear relations are summarized as follows:

$$I_{C_T} = -0.0001 C_T + 1.65 \qquad (5.17)$$

$$I_\eta = -0.0166\,\eta^2 + 1.3862\,\eta - 0.3818 \qquad (5.18)$$

$$I_\gamma = -0.025\,\gamma^2 + 0.295\,\gamma + 0.225 \tag{5.19}$$

The overall satisfaction index representing the optimization objective function was defined by:

$$I = \frac{I_{C_T} + 0.6 I_\eta + 0.3 I_\gamma}{1 + 0.6 + 0.3} \tag{5.20}$$

where the weighting factors for these 3 evaluation indices were selected as 1, 0.6, and 0.3 respectively.

5.4.6 Discussions

Improvement of computation efficiency is discussed in this section by comparing the optimization results obtained using the newly developed method and the results obtained using other methods.

The top 10 solutions with their overall satisfaction indices obtained using the newly developed method are shown in Figure 5.8. In this case study, the newly developed method was compared with the following two methods:

- The method to obtain 10 random solutions from the pruned tree and graphs.
- The method to obtain 10 random solutions from the original tree and graphs.

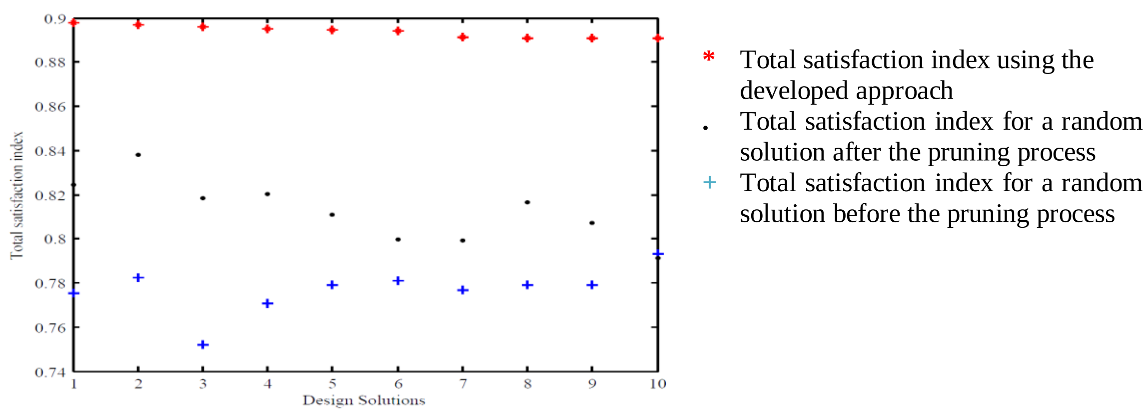


Figure 5.8: Overall satisfaction indices for different solutions obtained using different methods.

It is obvious that the pruning method can be used to reduce the search spaces and improve optimization efficiency by cutting off the nodes that are unlikely to lead to the optimal solutions. The initial ranking based on the heuristic function can be used to further improve the optimization efficiency by ignoring the solutions that have poor initial rankings.

Table 5.2 shows the detailed results for comparing the three methods. From this table, we can see that the solution space can be reduced considerably by pruning of the design tree and reconfiguration process graphs. By comparing the quality of the identified results, the average evaluation index of the 10 solutions achieved using each of the three methods was obtained. By using the pruning method, the quality was improved by 4.16%. By using both the pruning method and the initial ranking method, the quality was improved by 11.62%. In addition, the numbers of solutions in these three methods were also compared. By using the pruning method, the number of solutions was reduced to only 7×10^{-11} of the original number of solutions without pruning. By using both the pruning method and the initial ranking method, the number of solutions was reduced to 2.1×10^{-14}.

Table 5.2: Comparison among the three methods.

	Method without pruning of tree and graphs	**Method with pruning of tree and graphs**	**Method with pruning and initial ranking**
Total number of solutions	24×10^{14}	2.683×10^{5}	50
Average evaluation index of the 10 solutions	0.7777 (random solutions)	0.8193 (random solutions)	0.8939 (top ranking solutions)
Quality improvement from the method without pruning	------	4.16%	11.62%
Efficiency improvement from the method without pruning considering the ratio of numbers of possible solutions	------	7×10^{-11}	2.1×10^{-14}

5.5 Summary

A new efficient optimal reconfigurable product design method is introduced in this research considering large numbers of design candidates and reconfiguration process candidates. Compared with the method that was developed in our previous research (Gadalla and Xue 2017c) which was introduced in Chapter 4, pruning of the tree and graphs is conducted to reduce the search spaces. In addition, heuristic function is used to rank the design/process candidates after pruning to further improve the computation efficiency by using the top-ranked candidates for detailed optimization. The characteristics of this newly developed method are summarized as follows.

1. The new modified beam search is an effective method to cut off some nodes in the tree/graphs that are unlikely based on the heuristic evaluation to achieve the optimal design solution within the acceptable computation time. The global weights of nodes, representing the contributions of these nodes to the whole product or reconfiguration process, are good

heuristic measures to evaluate the nodes in the design tree and reconfiguration process graphs efficiently.

2. The initial ranking is an effective method to further improve the computation efficiency based on heuristic evaluations to the created design/process candidates after pruning and use the top-ranked design/process candidates for detailed optimization.
3. By improving the computation efficiency of the previously developed optimal reconfigurable product design method considering both the product configurations and reconfiguration processes (Gadalla and Xue 2017c), the new method is effective for reconfigurable product design when the numbers of design candidates and reconfiguration process candidates are large.

Chapter 6 An Extended Analytic Network Process (EANP) Method for Optimal Design of Reconfigurable Products Considering Dependency Relations among Descriptions of Design/Process Candidates and Evaluation Criteria

As discussed in Chapter 4, an optimization approach was introduced for the design of reconfigurable products considering both product configurations and reconfiguration processes (Gadalla and Xue 2017c). The efficiency of this optimization method was improved by the method introduced in Chapter 5 when numbers of design candidates and process candidates were large. Despite the effectiveness of the previously developed efficient optimization method for design of a reconfigurable product as discussed in Chapter 5, dependency relations among descriptions of design/process candidates and evaluation measures were not considered for identification of the weights of design/process nodes. To solve this problem, a new method based on analytic network process (ANP) (Saaty 1999), called extended analytic network process (EANP), is introduced in this research to model dependency relations among design/process nodes in the design/process AND-OR tree/graphs and evaluation criteria.

6.1 An Extended ANP (EANP) Method for Identification of Weights of Nodes Considering Dependency Relations among Design/Process Nodes and Evaluation Criteria

In this research, a new extended analytic network process (EANP) method is introduced to identify the weights of nodes to solve the following problems:

- The relative importance weights of nodes with an AND or OR relation explained in Chapter 5 were assigned directly or using the pairwise comparison method (Saaty 1977). The influences of a design/process node to other nodes without an AND or OR relation were not considered.
- The relative importance weights of design/process nodes explained in Chapter 5 were identified based on the single criterion considering importance factors of these nodes in a design/process. These weights were used for pruning of design/process tree/graph and initial rankings of design/process candidates for improving computation efficiency. The final solution was obtained considering multiple evaluation measures through multi-objective optimization. Therefore, identification of weights of nodes considering multiple evaluation criteria is expected.

6.1.1 Dependency relations among design/process nodes and evaluation criteria

In the efficient optimal reconfigurable product design method developed in Chapter 5, the relative importance measures for nodes with an AND relation or OR relation were defined by local weights. Figure 6.1(a) shows a generic design AND-OR tree for modeling a mechanism which is composed of a motion transmission module and a power module with the assigned local weights without considering the dependency relations.

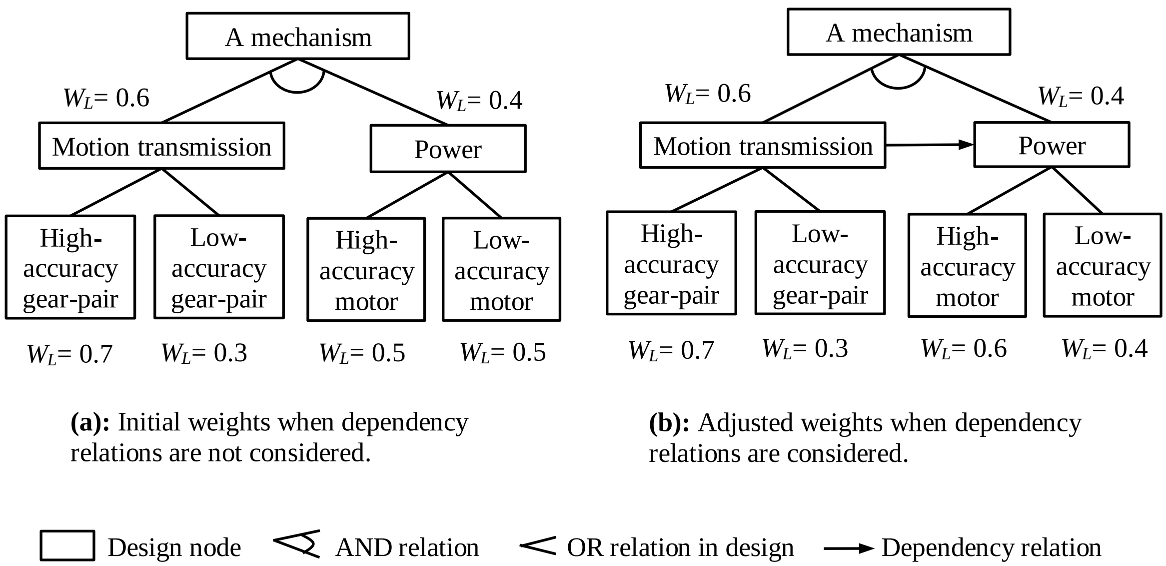


Figure 6.1: Influences of dependency relations on weights of nodes in a design AND-OR tree.

For this mechanism, when a high-accuracy gear-pair is preferred for the motion transmission function, a high-accuracy motor should also be preferred for the power function. In other words, the decision to select the design solution for the power module is influenced by the decision to choose the design solution for the motion transmission module. Due to the dependency relation, the weight of the high-accuracy motor is increased from the initial value of 0.5 to the adjusted one of 0.6, and weight of the low-accuracy motor is decreased from the initial value of 0.5 to the adjusted one of 0.4 as shown in Figure 6.1(b).

For the evaluation, when evaluation criteria are defined at different levels in a hierarchical structure, dependency relations among these evaluation criteria sometimes also need to be considered. Figure 6.2(a) shows the hierarchical structure of criteria to evaluate the mechanism given in Figure 6.1 with the assigned weights without considering the dependency relations among the evaluation criteria.

For this mechanism, when high quality is preferred for the motion transmission module, high quality should also be preferred for the power module. In other words, weights to the evaluation criteria for the power module are influenced by the weights to the evaluation criteria for the motion transmission module. Due to the dependency relation, weight of the quality for the power module is increased from the initial value of 0.5 to the adjusted one of 0.7, and weight of the cost for the power module is decreased from the initial value of 0.5 to the adjusted one of 0.3 as shown in Figure 6.2(b).

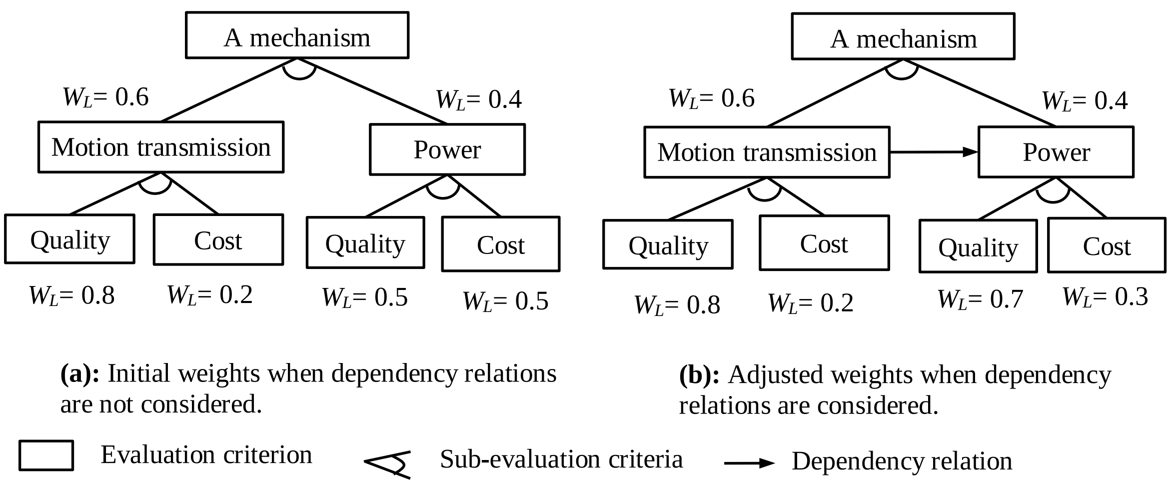


Figure 6.2: Influences of dependency relations on weights of evaluation criteria.

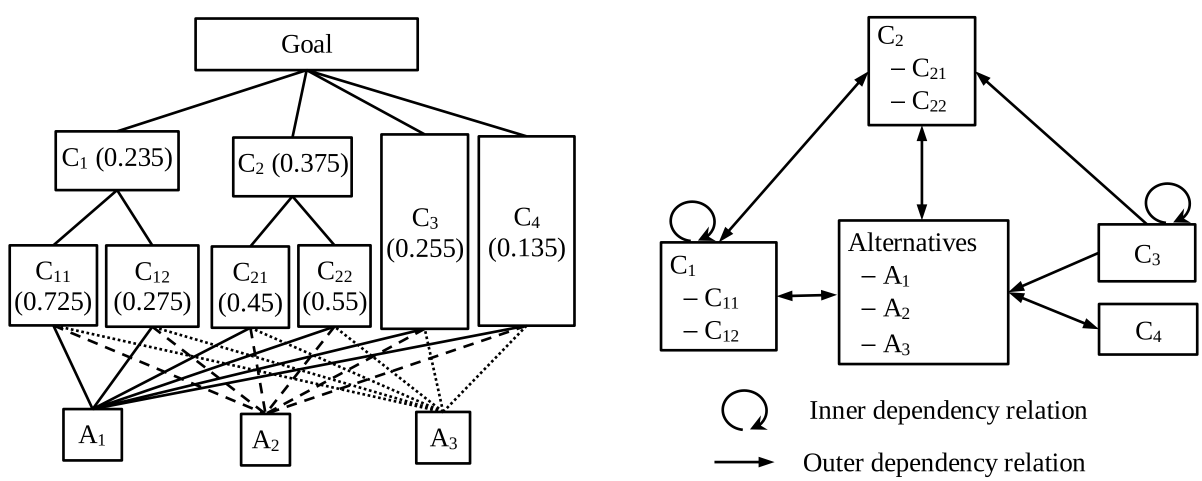
The analytic network process (ANP) (Saaty 1999) method is effective to compare and rank a group of candidates called alternatives considering dependency relations among these alternatives and evaluation criteria. In the present ANP method, only the weights for one cluster of alternatives can be calculated. Since weights of many clusters of nodes with AND and OR relations need to be identified for efficient optimal reconfigurable product design, the traditional ANP method needs to be extended. The ANP method will be explained in Section 6.1.2. The extended ANP (EANP) method developed in this research will be introduced in Section 6.1.3.

6.1.2 Analytic network process (ANP) method

The ANP method was developed from the popular analytic hierarchy process (AHP) method (Saaty 1999) considering the dependency relations among alternatives and evaluation criteria. Both AHP and ANP are used to rank a group of alternatives through obtaining the preference weights of these alternatives.

As discussed in Section 2.4.1, the evaluation criteria in AHP are defined in a hierarchical data structure as shown in Figure 6.3(a). When an evaluation criterion is defined by a group of sub-criteria, the relative importance factors of these sub-criteria are defined by weights. The relative weights of criteria are obtained through the pairwise comparison method (Saaty 1977). All alternatives are then compared considering one evaluation criterion through pairwise comparison. The comparison results for all alternatives considering one evaluation criterion are also described by weights. The preference of an alternative, called overall priority, is calculated based on the weights of this alternative against all the evaluation criteria and the weights of these evaluation criteria in the hierarchical data structure.

When the dependency relations among the alternatives and evaluation criteria have to be considered, the AHP will lead to inaccurate priorities/weights. The analytic network process (ANP) developed by Saaty and Takizawa (1986) can be used to overcome this problem by considering both the inner and outer dependency relations of the alternatives and evaluation criteria. So, ANP is a generalization of the AHP. In ANP, several clusters of evaluation criteria and one cluster of alternatives are modeled by nodes of the network as shown in Figure 6.3(b). Inner dependency relations and outer dependency relations among nodes are defined by arcs of the network.



C: Criterion, A: Alternative

(a): Analytic hierarchy process (AHP) method.

C: Criterion, A: Alternative

(b): Analytic network process (ANP) method.

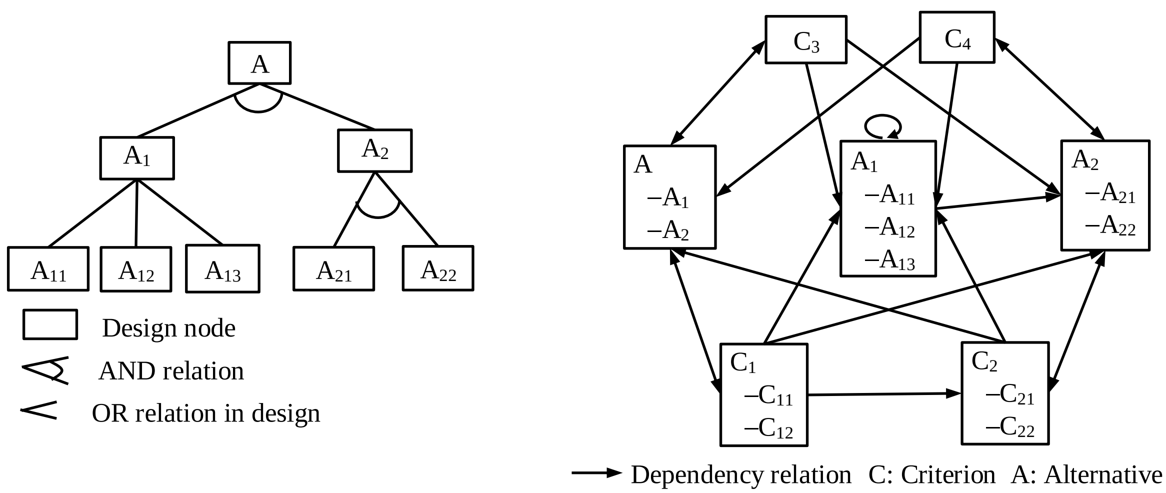
Figure 6.3: AHP and ANP methods.

The dependency relations among clusters are defined by weights in a cluster matrix $\mathbf{M_c}$. For each influencing cluster, its total influencing weight to other clusters is 1. Then different alternatives are compared considering each of the evaluation criteria through pairwise compassion. The evaluation results are described by priorities between 0 and 1, and the sum of the priorities for all alternatives considering one evaluation criterion is 1. In addition, the influences among alternatives and evaluation criteria in the clusters given in Figure 6.3(b), representing the inner and outer dependency relations, are also defined by weights between 0 and 1. The sum of the dependency weights for alternatives/criteria in a cluster against one alternative/criterion is 1. Then different alternatives in different clusters are compared considering each of the evaluation criteria through pairwise comparison in the unweighted matrix $\mathbf{M_u}$. The sum of the dependency weights for alternatives/criteria in a cluster considering one alternative/criterion is 1. The unweighted supermatrix $\mathbf{M_u}$ considering all alternatives and criteria in clusters is then converted into a weighted supermatrix $\mathbf{M_w}$ by multiplying each weight in $\mathbf{M_u}$ with its corresponding weight in $\mathbf{M_c}$. The weighted supermatrix $\mathbf{M_w}$ is further converted into the limited supermatrix $\mathbf{M_l}$ by raising its limiting power k through $(\mathbf{M_w})^k$ using the method developed by Saaty (1999). In the limited supermatrix, the weights in each row are converged into global priorities with the same value. Details of the methods to calculate $\mathbf{M_c}$, $\mathbf{M_u}$, $\mathbf{M_w}$, and $\mathbf{M_l}$ were provided in Section 2.4.2.

6.1.3 Extended analytic network process (EANP) method

In AHP and ANP, all the alternatives are modeled at the same level. Since the design/process nodes of a design/process tree/graph developed in our previous research, representing partial design/process solutions, were modeled at different levels, the traditional ANP needs to be extended to solve our optimal reconfigurable product design problem. In the new method developed in this research, the nodes with AND and OR relations are modeled as clusters in the ANP in addition to clusters of evaluation criteria. Dependency relations among nodes and evaluation criteria are also defined in the ANP. The final weights of the nodes considering the dependency relations, representing the true contributions of these partial solutions to the optimal reconfigurable product design, are achieved through ANP. Compared with the traditional ANP where alternatives are modeled at the same level in a single cluster of the network, the alternatives in our new method are modeled at different levels in different clusters of the network. Therefore, our ANP is called extended ANP (EANP) in this research.

Figure 6.4 shows a generic design AND-OR tree and the network considering 3 clusters of alternatives and 4 clusters of evaluation criteria. In the EANP method, the groups of nodes with AND and OR relations are defined by clusters of alternative. The 2 AND relations and 1 OR relation in Figure 6.4(a) are defined by 3 clusters of alternatives in Figure 6.4(b). Please note the sub-nodes with an AND relation are not the mutually exclusive alternatives, but the partial solutions for a parent node. Since a sub-node may have sub-sub-nodes with an OR relation, contributions of these sub-nodes with an AND relation to the optimal solution also need to be identified. To keep the consistency with the terms used in the ANP, nodes with an AND relation are also called alternatives in the EANP. The 4 groups of criteria, C_1, C_2, C_3 and C_4, are defined by 4 clusters of criteria. C_1 and C_2 are composed of sub-criteria. In the EANP network shown in Figure 6.4(b), the dependency relations among clusters of alternatives and criteria are also defined.



(a): A design AND-OR tree. **(b):** A network of clusters of alternatives and criteria.

Figure 6.4: Extended Analytic Network Process (ANP) method.

The procedure to identify the weights of the alternatives in EANP is the same as the procedure in ANP. Compared with the ANP where only the weights of alternatives in one cluster are identified for ranking these alternatives, the weights of alternatives in multiple clusters can be identified in the EANP.

In EANP, the dependency relations among clusters are first defined by weights in a matrix $\mathbf{M_c}$ as shown in Table 6.1. For each influencing cluster, its total influencing weights to other clusters is 1.

Table 6.1: Dependency relations among clusters (i.e., $\mathbf{M_c}$) given in Figure 6.4(b).

Influenced cluster	**Influencing clusters**						
	A	A_1	A_2	C_1	C_2	C_3	C_4
A	0	0	0	0.343	0.540	0.540	0.540
A_1	0	0.667	0	0.243	0.163	0.163	0.163
A_2	0	0.333	0	0.171	0.297	0.297	0.297
C_1	0.500	0	0	0	0	0	0
C_2	0	0	0.500	0.243	0	0	0
C_3	0.500	0	0	0	0	0	0
C_4	0	0	0.500	0	0	0	0

Then different alternatives in different clusters are compared considering each of the evaluation criteria through pairwise comparison as shown in the unweighted matrix $\mathbf{M_u}$ in Table 6.2(a). The evaluation results are described by priorities between 0 and 1, and the sum of the priorities for all alternative in a cluster considering one evaluation criterion is 1. Also, the influences among alternatives and evaluation criteria in the clusters given in Figure 6.4(b), representing the inner and outer dependency relations, are also defined by weights between 0 and 1 in the unweighted matrix $\mathbf{M_u}$ in Table 6.2(a). The sum of the dependency weights for alternatives/criteria in a cluster considering one alternative/criterion is 1. The unweighted supermatrix $\mathbf{M_u}$ is then converted into a weighted supermatrix $\mathbf{M_w}$ shown in Table 6.2(b) by multiplying each weight in Table 6.2(a) by the corresponding weight for the cluster defined in Table 6.1. Weights in each column of $\mathbf{M_w}$ is normalized such that the sum of weights is 1. The weighted supermatrix $\mathbf{M_w}$ is further converted into the limited supermatrix $\mathbf{M_l}$ as shown in Table 6.2(c) by raising its limiting power k through $(\mathbf{M_w})^k$ using the method developed by Saaty (1999). In the limited supermatrix, the weights in each row are converged into global priorities with the same value.

Table 6.2: Evaluation and ranking of different alternatives in different clusters by EANP.

(a): Unweighted supermatrix $\mathbf{M_u}$.

		A		**A_1**			**A_2**		**C_1**		**C_2**		**C_3**	**C_4**
		A_1	A_2	A_{11}	A_{12}	A_{13}	A_{21}	A_{22}	C_{12}	C_{12}	C_{21}	C_{22}		
A	A_1	0	0	0	0	0	0	0	0.750	0.667	0.750	0.667	0.667	0.750
	A_2	0	0	0	0	0	0	0	0.250	0.333	0.250	0.333	0.333	0.250
A_1	A_{11}	0	0	0.550	0	0	0	0	0.540	0.500	0.540	0.625	0.550	0.500
	A_{12}	0	0	0.210	0	0	0	0	0.300	0.250	0.297	0.238	0.210	0.250
	A_{13}	0	0	0.240	0	0	0	0	0.160	0.250	0.163	0.137	0.240	0.250
A_2	A_{21}	0	0	0.330	0.667	0.667	0	0	0.667	0.667	0.667	0.667	0.750	0.750
	A_{22}	0	0	0.670	0.333	0.333	0	0	0.333	0.333	0.333	0.333	0250	0.250
C_1	C_{11}	0.670	0.500	0	0	0	0	0	0	0	0	0	0	0
	C_{12}	0.330	0.500	0	0	0	0	0	0	0	0	0	0	0
C_2	C_{21}	0	0	0	0	0	0.750	0.667	0.750	0	0	0	0	0
	C_{22}	0	0	0	0	0	0.250	0.333	0.250	0	0	0	0	0
C_3		0	1	0	0	0	0	0	0	0	0	0	0	0
C_4		0	0	0	0	0	1	0	0	0	0	0	0	0

(b): Weighted supermatrix $\mathbf{M_w}$.

		A		**A_1**			**A_2**		**C_1**		**C_2**		**C_3**	**C_4**
		A_1	A_2	A_{11}	A_{12}	A_{13}	A_{21}	A_{22}	C_{12}	C_{12}	C_{21}	C_{22}		
A	A_1	0	0	0	0	0	0	0	0.257	0.302	0.405	0.360	0.360	0.405
	A_2	0	0	0	0	0	0	0	0.086	0.151	0.135	0.180	0.180	0.135
A_1	A_{11}	0	0	0.367	0	0	0	0	0.131	0.161	0.088	0.102	0.090	0.082
	A_{12}	0	0	0.140	0	0	0	0	0.072	0.080	0.049	0.039	0.034	0.041
	A_{13}	0	0	0.160	0	0	0	0	0.040	0.080	0.027	0.022	0.039	0.041
A_2	A_{21}	0	0	0.111	0.667	0.667	0	0	0.114	0.151	0.198	0.198	0.198	0.223
	A_{22}	0	0	0.222	0.333	0.333	0	0	0.057	0.076	0.099	0.099	0.099	0.074
C_1	C_{11}	0.667	0.250	0	0	0	0	0	0	0	0	0	0	0
	C_{12}	0.333	0.250	0	0	0	0	0	0	0	0	0	0	0
C_2	C_{21}	0	0	0	0	0	0.375	0.67	0.182	0	0	0	0	0
	C_{22}	0	0	0	0	0	0.125	0.33	0.061	0	0	0	0	0
C_3		0	0.500	0	0	0	0	0	0	0	0	0	0	0
C_4		0	0	0	0	0	0.500	0	0	0	0	0	0	0

(c): Limited supermatrix $\mathbf{M_l}$.

		A		**A_1**			**A_2**		**C_1**		**C_2**		**C_3**	**C_4**
		A_1	A_2	A_{11}	A_{12}	A_{13}	A_{21}	A_{22}	C_{12}	C_{12}	C_{21}	C_{22}		
A	A_1	0.152	0.152	0.152	0.152	0.152	0.152	0.152	0.152	0.152	0.152	0.152	0.152	0.152
	A_2	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059
A_1	A_{11}	0.077	0.077	0.077	0.077	0.077	0.077	0.077	0.077	0.077	0.077	0.077	0.077	0.077
	A_{12}	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036
	A_{13}	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
A_2	A_{21}	0.129	0.129	0.129	0.129	0.129	0.129	0.129	0.129	0.129	0.129	0.129	0.129	0.129
	A_{22}	0.075	0.075	0.075	0.075	0.075	0.075	0.075	0.075	0.075	0.075	0.075	0.075	0.075
C_1	C_{11}	0.116	0.116	0.116	0.116	0.116	0.116	0.116	0.116	0.116	0.116	0.116	0.116	0.116
	C_{12}	0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065
C_2	C_{21}	0.119	0.119	0.119	0.119	0.119	0.119	0.119	0.119	0.119	0.119	0.119	0.119	0.119
	C_{22}	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048
C_3		0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029
C_4		0.064	0.064	0.064	0.064	0.064	0.064	0.064	0.064	0.064	0.064	0.064	0.064	0.064

The final normalized weights for the nodes in 2 AND relations and 1 OR relation in Figure 6.4(a) are shown in Table 6.3.

Table 6.3: Identified weights for nodes with AND and OR relations in the design AND-OR tree.

Parent node	Sub-nodes	EANP weights	Normalized weights
A	A_1	0.152	0.722
	A_2	0.058	0.278
A_1	A_{11}	0.077	0.540
	A_{12}	0.036	0.249
	A_{13}	0.030	0.211
A_2	A_{21}	0.129	0.632
	A_{22}	0.075	0.368

6.2 Optimal Reconfigurable Product Design Based on Extended ANP (EANP) Considering Dependency Relations among Design/Process Nodes and Evaluation Criteria

Based on the method developed in our previous work explained in Chapter 5, the optimal reconfigurable product design based on extended ANP (EANP) considering dependency relations among design/process nodes and evaluation criteria is conducted through the following steps.

6.2.1 Modeling of generic AND-OR tree/graphs with the weights considering dependency relations among design/process nodes and evaluation criteria

The generic design considering different candidates is modeled by a design AND-OR tree with design nodes and their parameters. The relations among sub-nodes of parent nodes are defined by AND relations (AND), OR relations in design (OR-D), and OR relations in configuration (OR-C). The generic reconfiguration process to change between two product configurations is defined by a generic process AND-OR graph with operation nodes and their parameters. The relations among sub-nodes of parent nodes are defined by AND relations (AND), OR relations in operation (OR-O), and OR relations in design condition (OR-DC). Sequential constraints among operation nodes are also defined in the generic process AND-OR graphs.

For sub-nodes with an AND or OR relation, the relative importance factors of these sub-nodes are defined by local weights W_L. These weights are obtained using the extended ANP (EANP) method. The global weights of the nodes W_G in an AND-OR tree/graph, representing the important factors of these nodes in the whole design/process candidate, are calculated from the local weights and the relations defined in the AND-OR tree/graph. The rules explained in Section 5.2.1 are employed to calculate the global weights of the nodes.

6.2.2 Pruning of nodes in the generic AND-OR tree/graphs

When large numbers of design and process candidates can be created from the generic design AND-OR tree and reconfiguration process AND-OR graphs, some nodes with OR relations in design for the design tree and some nodes with OR relation in operations for the process graphs, which are unlikely to lead to the optimal solutions, are pruned to reduce the numbers of design and process candidates. Pruning of the nodes is conducted through a modified beam-search method which was explained in Section 5.2. In the modified beam-search, a threshold value, α_i, is defined as the percentage of the alternative nodes to be selected at the *i*-th level of the AND-OR tree/graph. The global weights of the nodes are used as heuristic evaluation measures to decide what nodes should be cut off. For each group of nodes with an OR relation in design for the design tree or nodes with an OR relation in operation for the reconfiguration process graph, the node with the highest global weight should be selected before the pruning process.

6.2.3 Creation of candidates, and initial ranking of these candidates using a heuristic function

A specific design candidate is created from the pruned generic design AND-OR tree through tree-based search. A specific design candidate is modeled by a tree with AND relations and OR relations in configuration. A specific reconfiguration process candidate is created from the pruned generic reconfiguration process AND-OR graph though graph-based search. A specific reconfiguration process candidate is modeled by a graph with only AND relations.

The top *n* candidates are identified from the pruned tree/graph based on the heuristic function. In this research, two approaches are used for identification of the top-ranked candidates.

- Exhaustive approach: to create all the candidates and evaluate all these candidates using the heuristic function when the search space is not large.
- Evolutionary approach: to create *n* candidates randomly as the first population and evolve the populations using genetic programming (Hong et al. 2010) when the search space is large.

6.2.4 Identification of the final optimal design solution

For the top-ranked design solutions created from the top-ranked design candidates and process candidates, optimization is conducted to identify the optimal design solution that is modelled by a design candidate with different configurations, its reconfiguration processes, and design and process parameter values.

First, the *j*-th evaluation measure, F_j, is converted into the *j*-th evaluation index, I_j, by:

$$I_j(S_i) = g_j(F_j) = g_j[f_j(S_i)],\ i = 1, 2, \ldots, p;\ j = 1, 2, \ldots, m \tag{6.1}$$

where S_i $(i=1, 2, \ldots, p)$ is the *i*-th design solution, *p* is the number of design solutions, F_j $(j=1, 2, \ldots, m)$ is the *j*-th evaluation measure, *m* is the number of evaluation measures, and $g_i()$ is the non-linear relation between an evaluation measure and its evaluation index. In this research, the method developed by Xue, Yang, and Tu (2006) is used to obtain the non-linear relations between the evaluation measures and their comparable evaluation indices.

The overall evaluation index, $I(S_i)$, is calculated from all the evaluation indices by:

$$I(S_i) = \frac{\sum_{j=1}^{m}\left(W_j I_j(S_i)\right)}{\sum_{j=1}^{m} W_j} \tag{6.2}$$

where $W_1, W_2, \ldots, W_m$ are the importance weights for the *m* evaluation indices.

When values of design/process parameter variables also need to be identified, numerical optimization is then conducted to obtain the optimal values of these parameters. Numerical optimization considering constraints is defined by:

$$\max_{w.r.t.\boldsymbol{X_i}} I(\boldsymbol{X_i}), \quad i = 1,2,\ldots,p \tag{6.3}$$

subject to:

$$\boldsymbol{X_i^{(L)}} \leq \boldsymbol{X_i} \leq \boldsymbol{X_i^{(U)}}$$
$$h_{i,k}(\boldsymbol{X_i}) \geq 0, \quad k = 1,2,\ldots$$
$$g_{i,k}(\boldsymbol{X_i}) = 0, \quad k = 1,2,\ldots$$

where $\boldsymbol{X_i}$ is a vector with design and process parameter variables for the *i*-th design solution, S_i $(i=1, 2, \ldots, p)$.

6.3 A Case Study

The problem of this case study was to identify the optimal design solution of a reconfigurable truck. The reconfigurable truck provided different functions through three different configurations: standard truck (S), recreational vehicle (R), and canopy truck (C) as shown in Figure 6.5. A total of six reconfiguration processes (i.e., SR, RS, SC, CS, RC, and CR) were considered to change between any two different configurations of this reconfigurable truck.

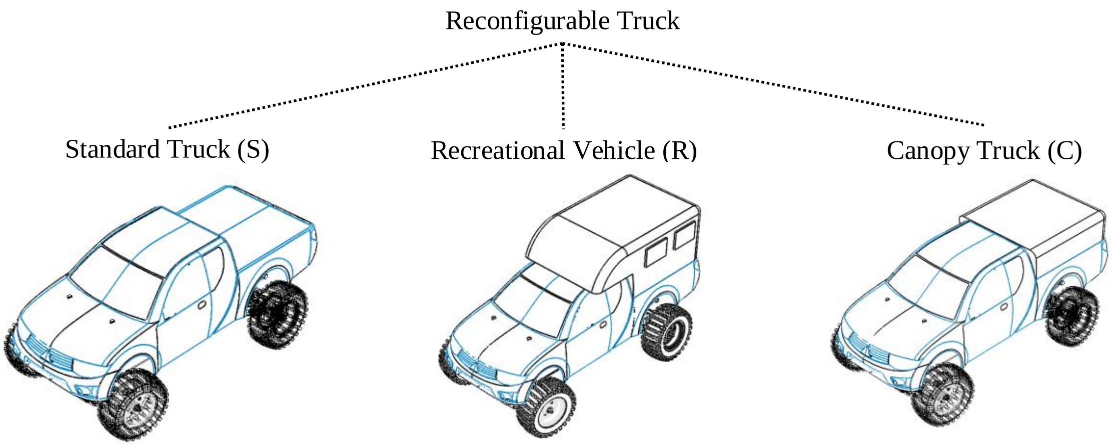


Figure 6.5: A reconfigurable truck with three configurations.

6.3.1 Modeling of generic AND-OR tree/graphs with the weights considering dependency relations among design/process nodes and evaluation criteria

A generic design AND-OR tree as shown in Figure 6.6 and 6 generic reconfiguration process AND-OR graphs were used to model all available design candidates and reconfiguration process candidates. Due to the large numbers of design candidates and reconfiguration process candidates, design nodes and operation nodes in these AND-OR tree/graphs were associated with importance weights such that the nodes with low weights were pruned to improve the computation efficiency.

In this case study, the extended ANP (EANP) method was employed to identify the local weights of some nodes in the AND-OR tree/graphs considering dependency relations among design/process nodes and evaluation criteria. The dependency relations between the design nodes in 3 clusters selected from Figure 6.6 and evaluation criteria in 2 clusters are shown in Figure 6.7. EANP was used to identify the local importance weights for these selected design nodes. Since total cost, reliability, and payload were used as three objective functions in the final optimization process, three relevant criteria, including product cost, reliability, and payload, were selected in the EANP. The three criteria were grouped into two clusters: cost and performance.

A pairwise comparison was then conducted to get the matrix to model the dependency relations among clusters $\mathbf{M_c}$ as shown in Table 6.4(a) and the unweighted supermatrix $\mathbf{M_u}$ to model the evaluation results and dependency relations among sub-nodes/sub-criteria in the clusters as shown in Table 6.4(b).

The unweighted supermatrix $\mathbf{M_u}$ showed in Table 6.4(b) was then converted into the weighted supermatrix $\mathbf{M_w}$ using the weights defined in $\mathbf{M_c}$ of Table 6.4(a), and this weighted supermatrix $\mathbf{M_w}$ was subsequently converted into the limited supermatrix $\mathbf{M_l}$. The weights in each row of the limited supermatrix were converged into the same value. Table 6.5 shows the converged weights for the sub-nodes in the selected 3 clusters. The local weights for nodes in each cluster with an AND or OR relation were achieved by normalizing the EANP weights in the cluster. These identified local weights are shown in Figure 6.6.

6.3.2 Pruning of nodes in the generic AND-OR tree/graphs to reduce the numbers of design/process candidates

In this step, some nodes with OR relations in design for the design tree and some nodes with OR relations in operations for the process graphs were pruned using a modified beam-search method with the threshold ratio values, α_i, at different levels in the tree/graph. First, the global weights of the nodes, representing contributions of these nodes to the whole design/process, were calculated from the local weights using the methods given in Section 5.1.2. Figure 6.8 shows the design AND-OR tree with the global weights. Then the best node from each group of nodes with an OR relation in design for the design tree or with an OR relation in operation for the reconfiguration process graph was selected. The rest nodes at each level were then ranked and the nodes with poor global weights were cut off based on the threshold ratios, α_i. In this case study, the threshold ratios for the levels 1-4 of the design nodes in Figure 6.8 were selected as α_1=1, α_2=0.7, α_3=0.6, and α_4=0.5 respectively. For the second level in this AND-OR tree, only 6 sub-nodes of the Engine node were associated with an OR relation in design. The best node, Japanese 3.5 L, was selected first from these sub-nodes. Among the rest 5 sub-nodes, 3 sub-nodes were selected and 2 sub-nodes were cut off based on the ratio α_2=0.7.

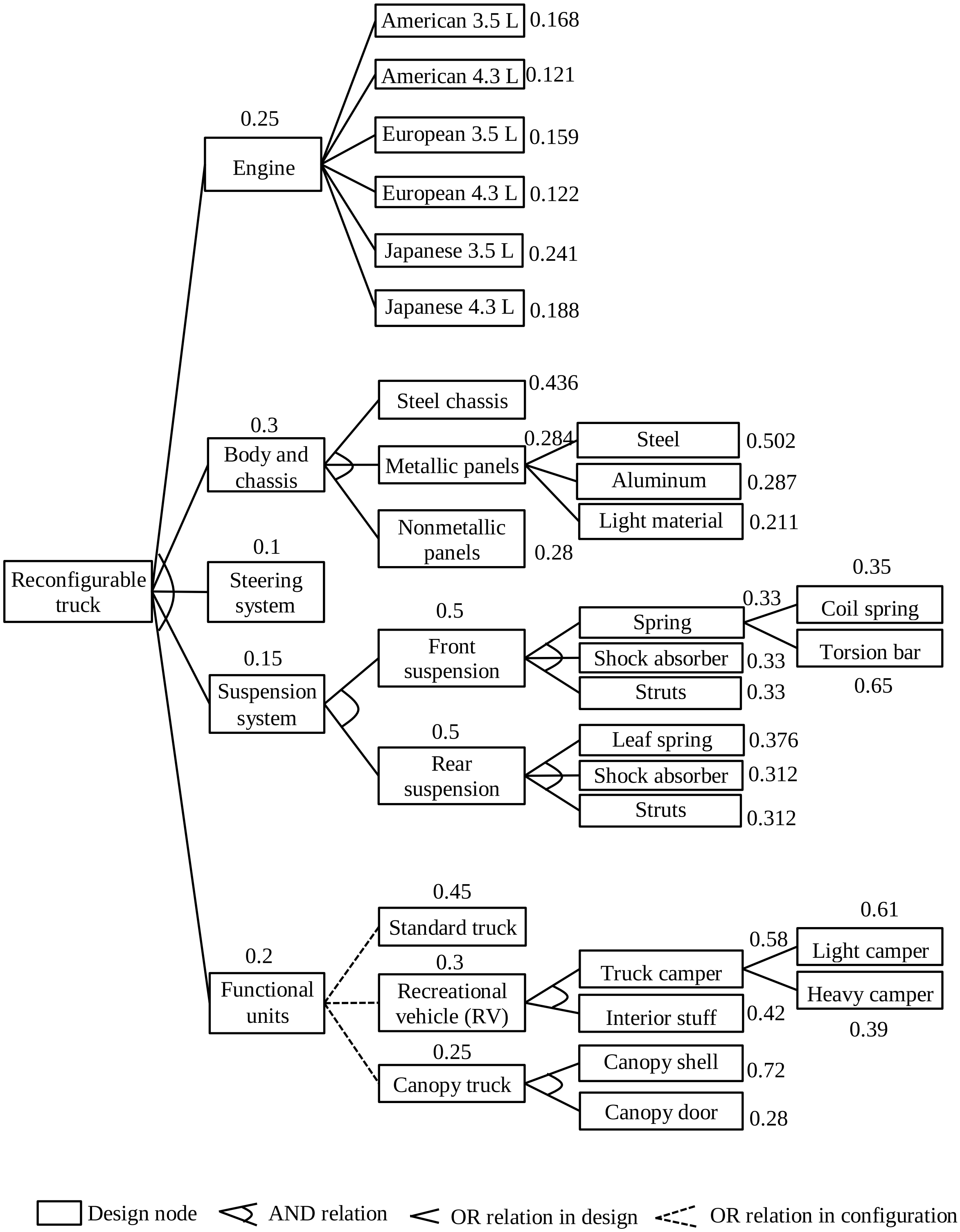


Figure 6.6: Generic design AND-OR tree with local importance weights for modeling a reconfigurable truck.

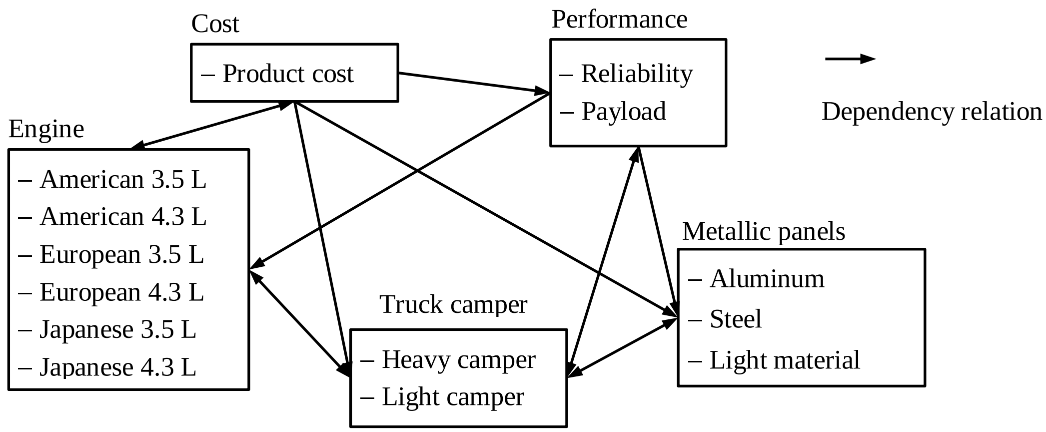


Figure 6.7: A network of clusters of alternatives and criteria with their dependency relations.

Table 6.4: Identification of the matrices based on EANP.

(a): Dependency relations among clusters ($\mathbf{M_c}$).

Influenced cluster	**Influencing clusters**				
	Engine	Metallic panels	Truck camper	Cost	Quality
Engine	0	0	0.545	0.618	0.600
Metallic panels	0	0	0.182	0.198	0.200
Truck camper	0.333	1	0	0.138	0.200
Cost	0.667	0	0	0	0
Performance	0	0	0.273	0.046	0

(b): Unweighted supermatrix ($\mathbf{M_u}$).

		Engine						**Metallic panels**			**Truck camper**		**Cost**	**Performance**	
		A3.5L	A4.3L	E3.5L	E4.3L	J3.5L	J4.3L	AL	LM	ST	HC	LC	PC	P	R
Engine	A3.5L	0	0	0	0	0	0	0	0	0	0.041	0.160	0.229	0.057	0.100
	A4.3L	0	0	0	0	0	0	0	0	0	0.147	0.124	0.106	0.160	0.100
	E3.5L	0	0	0	0	0	0	0	0	0	0.096	0.247	0.111	0.107	0.200
	E4.3L	0	0	0	0	0	0	0	0	0	0.239	0.111	0.064	0.282	0.200
	J3.5L	0	0	0	0	0	0	0	0	0	0.130	0.247	0.295	0.111	0.200
	J4.3L	0	0	0	0	0	0	0	0	0	0.318	0.111	0.194	0.282	0.200
Metallic panels	AL	0	0	0	0	0	0	0	0	0	0.297	0.333	0.230	0.286	0.462
	LM	0	0	0	0	0	0	0	0	0	0.163	0.333	0.122	0.143	0.308
	ST	0	0	0	0	0	0	0	0	0	0.540	0.333	0.648	0.571	0.231
Truck camper	HC	0.250	0.333	0.333	0.500	0.333	0.500	0.333	0.333	0.500	0	0	0.333	0.667	0.333
	LC	0.750	0.667	0.667	0.500	0.667	0.500	0.667	0.667	0.500	0	0	0.667	0.333	0.667
Cost	PC	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Perfor-mance	P	0	0	0	0	0	0	0	0	0	0.667	0	0.333	0	0
	R	0	0	0	0	0	0	0	0	0	0.333	0	0.667	0	0

A3.5L: American 3.5 L, A4.3L: American 4.3 L, E3.5L: European 3.5 L, E4.3L: European 4.3 L,

J3.5L: Japanese 3.5 L, J4.3L: Japanese 4.3 L, AL: Aluminum, LM: Light material, ST: Steel,

HC: Heavy camper, LC: Light camper, PC: Product cost, P: Payload, R: Reliability

Table 6.5: Identified weights for nodes with AND and OR relations in the design AND-OR tree.

Parent node	Sub-nodes	EANP weights	Normalized weights
Engine	American 3.5 L	0.058	0.168
	American 4.3 L	0.042	0.121
	European 3.5 L	0.055	0.159
	European 4.3 L	0.042	0.122
	Japanese 3.5 L	0.084	0.241
	Japanese 4.3 L	0.065	0.188
Metallic panels	Aluminum	0.033	0.287
	Light material	0.024	0.211
	Steel	0.057	0.502
Truck camper	Heavy camper	0.105	0.39
	Light camper	0.164	0.61

6.3.3 Creation of design candidates and their reconfiguration process candidates, and initial ranking of these candidates using a heuristic function

To further improve the computation efficiency, initial ranking to the design candidates and reconfiguration process candidates was conducted using a heuristic function, such that only the top-ranked candidates were selected for detailed evaluation and optimization. In this case study, global weights of nodes in design/process candidates were used for initial ranking.

For the design AND-OR tree given in Figure 6.8, since only the sub-nodes for the engine, metallic panels and truck camper with OR relations led to different design candidates, initial evaluations considering only the differences of these sub-nodes were conducted in this work. The top 5 design candidates are shown in Table 6.6.

Table 6.6: The top-ranked design candidates using EANP.

Design candidate	Engine		Metallic panels		Truck camper		Total global weights
	Type	Weight	Type	Weight	Type	Weight	
1	Japanese 3.5 L	0.25	Steel	0.085	Light camper	0.035	0.370
2	Japanese 3.5 L	0.25	Steel	0.085	Heavy camper	0.023	0.358
3	Japanese 4.3 L	0.223	Steel	0.085	Light camper	0.035	0.343
4	Japanese 3.5 L	0.25	Aluminum	0.054	Light camper	0.035	0.339
5	American 3.5 L	0.212	Steel	0.085	Light camper	0.035	0.332

6.3.4 Identification of the final optimal design solution

Complete evaluations of the top-ranked design solutions considering the five top-ranked design candidates and the five top-ranked reconfiguration process candidates for each of the six reconfiguration processes were conducted to identify the optimal design solution.

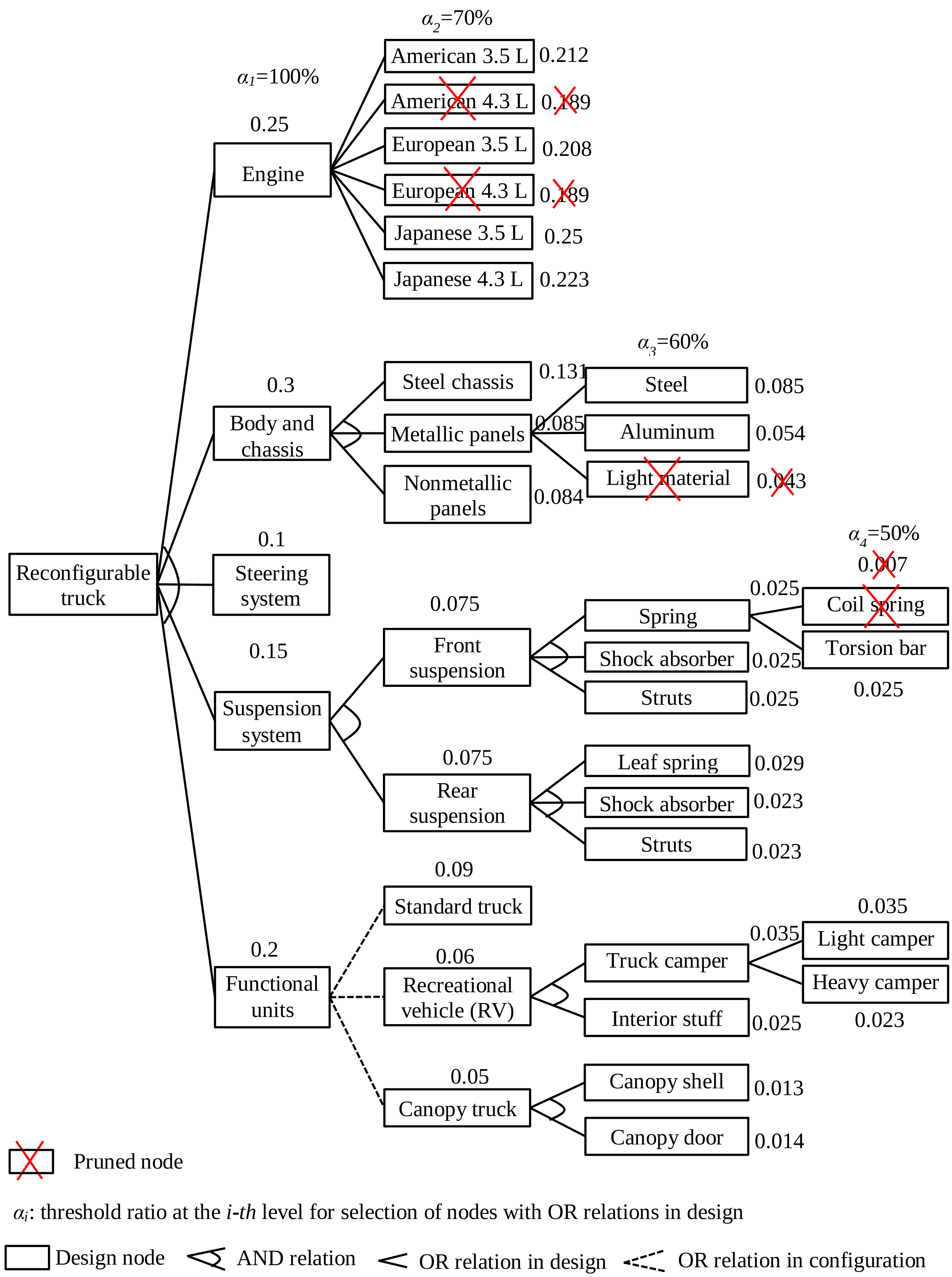


Figure 6.8: Pruning of nodes in the generic design AND-OR tree.

6.3.4.1 Evaluation measures

In this case study, the total cost considering different truck configurations and reconfiguration processes, the overall reliability considering the reliabilities of the components and modules of the reconfigurable truck, and the payload of the truck were selected as the three evaluation measures.

(1) Total cost.

The total cost for the i-th candidate of the reconfigurable product, C_{iT}, was calculated by adding the product cost considering all configurations, C_{iC}, and the costs considering all reconfiguration processes, C_{iR}.

$$C_{iT} = C_{iC} + C_{iR}\ ,\ i = 1, 2, \dots, p \tag{6.4}$$

The product cost, C_{iC}, considering all configuration was calculated by adding all the costs of components and modules of this reconfigurable truck. The reconfiguration process cost, C_{iR}, was calculated by adding all the costs for the 6 reconfiguration processes during the whole product lifespan. For this case study, it was assumed that the reconfigurable truck needed 10 reconfiguration processes (60% for *SC* and *CS*, 25% for *RS* and *SR*, and 15% for *CR* and *RC*) each year and its lifespan was 15 years. During the 15 years, the total number for *SC* and *CS* reconfiguration processes, $N_{SC\&CS}$, was 90, the total number for *CR* and *RC* reconfiguration processes, $N_{CR\&RC}$, was 22, and the total number for *SR* and *RS* reconfiguration processes, $N_{SR\&RS}$, was 38. The total reconfiguration process cost, C_{iR}, was calculated by:

$$C_{iR} = \left[\frac{N_{SC\&CS}}{2} \times (C_{iSC} + C_{iCS})\right] + \left[\frac{N_{RC\&CR}}{2} \times (C_{iRC} + C_{iCR})\right] + \left[\frac{N_{SR\&RS}}{2} \times (C_{iSR} + C_{iRS})\right] + C_{tools} \tag{6.5}$$

where C_{ijk} was the cost for all reconfiguration operations to change from the j-th configuration to the k-th configuration of the i-th design candidate (Gadalla and Xue 2017c), and C_{tools} was the cost of all tools required for all reconfiguration operations during the whole product lifespan (C_{tools} = \$10,000). The cost of the o_{ijk}-*th* operation was calculated based on the effective time required to finish the o_{ijk}-*th* operation, $T_{o_{ijk}}$ (*hour)*, the labor cost rate for one operator, C_l (C_l =\$60/hour), and the number of required operators for the o_{ijk}-*th* operation, $l_{o_{ijk}}$.

$$C_{ijk} = \sum_{o_{ijk}=1}^{B_{ijk}} \left(T_{O_{ijk}} \times C_l \times l_{O_{ijk}}\right), \quad i = 1,2,\ldots,p; j = 1,2,\ldots; k = 1,2,\ldots; j \neq k \tag{6.6}$$

where B_{ijk} was the total number of operations. The required time was calculated by:

$$T_{O_{ijk}} = T_{Base_{O_{ijk}}} \left(D_{H_{O_{ijk}}} + D_{I_{O_{ijk}}}\right) \tag{6.7}$$

where $T_{Base_{o_{ijk}}}$ was the base time required to finish the o_{ijk}-*th* operation (*hour)* without considering any complexities of handling and insertion of components, and $D_{H_{o_{ijk}}}$ and $D_{I_{o_{ijk}}}$ were the handling and insertion complexity factors for the o_{ijk}-th reconfiguration process (Samy and ElMaraghy 2010; Soh, Ong, and Nee 2016).

(2) *Overall reliability*.

The overall reliability of the design candidate was calculated by considering the reliabilities of all components/modules of the candidate. The overall reliability of the i-th design candidate, η_i, was calculated by:

$$\eta_i = \sum_{D_{m_i=1}}^{M_i} \left(\eta_{D_{m_i}} \times W_{D_{m_i}}\right), \quad i = 1,2,\ldots,p \tag{6.8}$$

where $\eta_{D_{m_i}}$ and $W_{D_{m_i}}$ were the reliability and global importance weight of the m-th node in the i-th design candidate, and M_i was the total number of nodes in the i-th design candidate.

(3) Payload.

For this reconfigurable truck, the payload was selected to evaluate the carrying capacity of the truck. This payload P_i was calculated based on two selected design variables, the length d_i and width w_i of the truck for the i-th design solution. Sophisticated equations in a computer program were used to model the relation between the two design variables and the payload evaluation measure.

$$P_i = f(w_i, d_i) \tag{6.9}$$

6.3.4.2 *Optimization*

The three different evaluation measures in different units were converted into comparable evaluation indices representing customer satisfaction levels. The least-square curve fitting method was used to obtain the non-linear relations between evaluation measures and their evaluation indices using the collected data of customers' and engineers' satisfaction levels considering different evaluation measures. With the increase of the value for C_T, its satisfaction index I_{C_T} was decreased, however, with increases of the values for η and P, their satisfaction indices I_η and I_{P} were increased. These non-linear relations are summarized as follows:

$$I_{C_T} = -9 \times 10^{-6} C_T + 1.8 \tag{6.10}$$

$$I_\eta = 2 \times 10^{-13} \eta^2 + 0.5\eta - 0.5 \tag{6.11}$$

$$I_{\mathrm{P}} = -0.0024P^2 + 0.1095P + 0.929 \tag{6.12}$$

The overall satisfaction index representing the optimization objective function was defined by:

$$I = \frac{I_{C_T} + 0.6\, I_\eta + 0.2\, I_{\mathrm{P}}}{1 + 0.6 + 0.2} \tag{6.13}$$

where the weighting factors for these 3 evaluation indices were selected as 1, 0.6, and 0.2 respectively.

In this case study, only two parameters related to the major truck specifications were selected as variables to be optimized. These parameters were the truck width, *w (m)*, and the car length, *d (m)*, considering constraints $1.8 \le w \le 2.2$ and $4.0 \le d \le 4.5$. The parameter optimization was defined as:

$$\max_{w.r.t.\boldsymbol{X_i}} I(w, d_i) \tag{6.14}$$

subject to:

$1.8 \le w \le 2.2$

$4.0 \le d \le 4.5$

The optimal design was identified as:

- Configuration: Engine = Japanese 3.5 L, Metallic panels = Steel, Truck camper = Light camper.
- Parameters: w =2.2 m, d =4 m.
- The overall evaluation index: I = 0.918.

6.3.5 Discussions

Improvement of the optimization quality using the new extended analytic network process (EANP) method considering dependency relations among descriptions of design/process candidates and evaluation criteria is discussed in this section through comparing the optimization results obtained using the importance weights identified by EANP and the results obtained using the analytic hierarchy process (AHP). In AHP, the importance weights of nodes were achieved without considering any dependency relations.

The overall satisfaction indices for the top ten design solutions using the two methods are shown in Figure 6.9. It is clear that the satisfaction indices for the designs using the new extended analytic network process method are higher than the satisfaction indices for the designs using the analytic hierarchy process. The average satisfaction index for the ten solutions using the EANP was 0.913, and the average satisfaction index for the ten solutions using the AHP was 0.895. This difference was not significant for this case study, because only small number of dependency relations were considered. However, in the real-world applications for design of reconfigurable products, the large numbers of design/process nodes and evaluation criteria may lead to large numbers of dependency relations. In these cases, the EANP has to be used to improve the optimization quality for design of reconfigurable products.

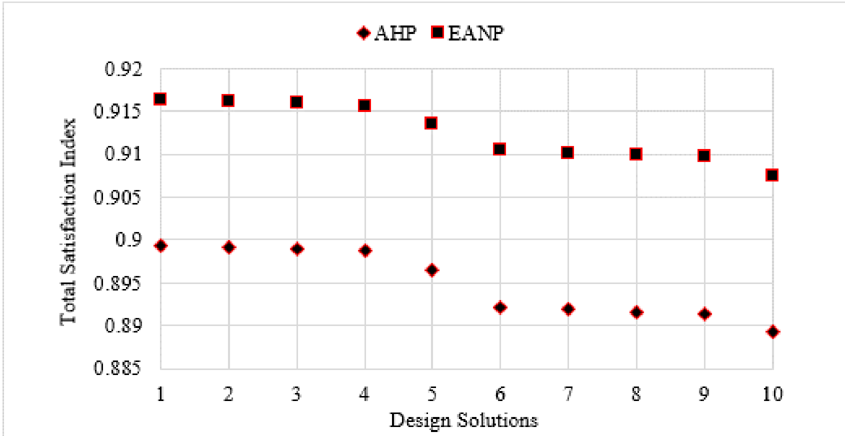


Figure 6.9: Overall satisfaction indices for different solutions obtained using AHP and EANP.

6.4 Summary

A new method, called extended analytic network process (EANP) method, is introduced in this research to identify the importance weights of design/process nodes considering the dependency relations among design/process nodes and evaluation criteria. The advantages of this newly developed method are summarized as follows.

1. Compared with the traditional methods to achieve the importance weights of the nodes in design/process tree/graphs, including the direct assignment method, the pairwise comparison method and the AHP method without considering the dependency relations among these nodes, the newly developed EANP method can be used to achieve better weights. The newly developed EANP method truly reflects the importance factors of the nodes in design/process tree/graphs considering the dependency relations among these nodes. So, better optimal designs of reconfigurable products can be achieved through pruning of the truly poor nodes that are unlikely to achieve the optimal solutions.

2. Compared with the methods developed in our previous research in which the importance weights of design/process nodes were used for pruning of design/process tree/graphs and initial rankings of design/process candidates while the final solution was obtained considering multiple evaluation measures through multi-objective optimization, the multiple evaluation criteria are considered in the newly developed EANP method to identify the weights of design/process nodes. The EANP method is used to identify better weights which truly reflect the contributions of these nodes to the optimal solutions by

considering evaluation criteria like the evaluation measures in detailed optimization. Since nodes with poor contributions to the optimal solutions are pruned, better optimal design solution can be achieved.

Chapter 7 Conclusions and Future Work

In this chapter, summary and contributions of the research on the optimal design of reconfigurable products considering product configurations and reconfiguration processes are presented first. Then the future work is provided.

7.1 Summary of This Research

In this research, a systematic approach for optimal design of reconfigurable products considering product configurations and reconfiguration processes has been developed. The research was conducted in three steps. First, new modeling, evaluation, and optimization schemes have been developed to identify the optimal design solution of a reconfigurable product considering product configurations and reconfiguration processes as explained in Chapter 4. Second, a new method has been developed to improve optimization efficiency for optimal design of reconfigurable products when large numbers of design candidates and reconfiguration process candidates are considered as explained in Chapter 5. Third, a new method has been developed to improve optimization quality for optimal design of reconfigurable products when dependency relations among design/process descriptions and evaluation criteria are considered as explained in Chapter 6.

(1) An approach to identify the optimal configurations and reconfiguration processes for the design of reconfigurable products.

In this research, a modelling scheme was introduced to describe different product configurations, reconfiguration processes, and design/process parameters. A generic design AND-OR tree was used to model different design solution candidates, their product configurations and parameters of these configurations. For a reconfiguration process to change between two product configurations, a generic process AND-OR graph was used to model reconfiguration operation candidates, sequential constraints among operations and operation parameters. Then, an evaluation scheme was developed to evaluate both the product configurations and reconfiguration processes through multiple evaluation measures described in different units. In the developed evaluation method, these evaluation measures with different units were integrated into the same environment by converting these evaluation measures into comparable evaluation indices. Finally, a multi-level and multi-objective optimization scheme was developed to identify the optimal design solution.

The optimization was conducted at two different levels: the candidate optimization level to identify the optimal design candidate that was modeled by its product configurations and reconfiguration processes, and the parameter optimization level to identify the optimal parameter values of product configurations and reconfiguration processes.

(2) *An efficient optimization approach for the design of reconfigurable products considering large numbers of design candidates and reconfiguration process candidates.*

When the numbers of design candidates and reconfiguration process candidates are large, computation efficiency has to be considered such that the optimal solution can be achieved within the required computation time period. In this research, a new method was developed to improve the optimization efficiency for optimal design of reconfigurable products considering large numbers of design candidates and reconfiguration process candidates. First, a heuristic was developed to evaluate tree/graphs nodes considering importance of each node to the tree/graphs. Then, a new modified beam search method was developed for pruning of some nodes with poor weights in the design tree and reconfiguration process graphs that were unlikely to achieve the optimal solution. When numbers of design candidates and reconfiguration process candidates were still large, the optimization search space was further reduced through ranking of the created design/process candidates and the top-ranked design/process candidates were used for detailed optimization. Finally, a multi-level and multi-objective optimization was used to identify the optimal design solution based on evaluations of the top-ranked design candidates and reconfiguration process candidates.

(3) *An EANP method for optimal design of reconfigurable products considering dependency relations among descriptions of design/process candidates and evaluation criteria*.

In the previous research, dependency relations between design/process nodes and evaluation criteria were not considered for identification of importance weights of design/process nodes. In this research, the quality of the identified importance weights of nodes was improved through considering dependency relations among design/process nodes and evaluation criteria. A new method, called an extended analytic network process (EANP) method, was developed to determine importance weights of design/process nodes considering dependency relations for the rating of contributions of these design/process nodes to the optimal design solution. The new method was used to achieve better initial evaluations for design/process nodes which truly reflected the contributions of these nodes to the optimal solutions when dependency relations among these nodes were considered. Since the truly poor nodes were cut off in this new method, better optimal designs of reconfigurable products were achieved. In addition, by selecting the evaluation criteria in the EANP method which were similar to the evaluation measures used in the final optimization process, better top-ranked solutions were identified for detailed optimization.

7.2 Research Contributions

The overall contribution of this research is the development of a systematic approach for optimal design of reconfigurable products considering product configurations and reconfiguration processes.

The contributions of the developed approach are in the following three aspects.

- A method for modeling, evaluation, and optimization of a reconfigurable product considering different product configurations and reconfiguration processes to change between its configurations.
- A method to improve the optimization efficiency for optimal design of reconfigurable products considering large numbers of design candidates and reconfiguration process candidates.
- A method to improve the optimization quality for optimal design method of reconfigurable products considering dependency relations among descriptions of design/process candidates and evaluation criteria.

(1) A new method for modeling, evaluation, and optimization of a reconfigurable product considering different product configurations and reconfiguration processes to change between its configurations.

As discussed in Chapter 2, in the existing methods for modelling, evaluation, and optimization of reconfigurable products, only product configurations were considered to identify the optimal design solutions. However, efforts of reconfiguration processes, such as time and cost, also influence customer satisfaction of the reconfigurable product. To overcome these limitations in the previous methods, new modelling, evaluation, and optimization schemes have been developed to identify the optimal design of a reconfigurable product considering both the product configurations and reconfiguration processes.

- A new modelling scheme has been introduced to describe different design candidates, product configurations, reconfiguration processes, and parameters of product configurations and reconfiguration processes using design/process AND-OR tree/graphs.
- A new evaluation scheme has been developed to evaluate a reconfigurable product considering both the product configurations and reconfiguration processes. In addition, the new evaluation method integrates the evaluations to both product configurations and reconfiguration processes from different perspectives into the same environment.
- A new optimization method, called multi-level and multi-objective optimization, has been developed to identify the optimal design solution of a reconfigurable product that is modeled by product configurations, reconfiguration processes, and parameters of product configurations and reconfiguration processes.

(2) A new method to improve optimization efficiency for optimal design of reconfigurable products considering large numbers of design candidates and reconfiguration process candidates.

In the existing methods for optimal design of reconfigurable products, considerable computation effort is required to identify the optimal design solution when numbers of design candidates and reconfiguration process candidates are large. A new method has been developed to improve the computation efficiency of the optimal reconfigurable product design method when the numbers of design candidates and reconfiguration process candidates are large.

- A new pruning method, called modified beam search, has been developed to improve the computation efficiency by cutting off some design/process nodes that are unlikely to achieve the optimal design solution. In this method, the contributions of design/process nodes to the optimal design are defined by importance weights. During the search process, percentages of nodes to be selected from nodes with OR-relations in design for the design tree and nodes with OR-relations in operation for the reconfiguration process graph are defined such that the nodes with poor contributions to the optimal solution are cut off from the design/process AND-OR tree/graphs.

- A new ranking method has been developed for further improvement of the computation efficiency through sorting the created design/process candidates using heuristics, such that only the top-ranked design candidates and reconfiguration process candidates are used for detailed optimization.

(3) *A new method to improve optimization quality for optimal design of reconfigurable products considering dependency relations among descriptions of design/process candidates and evaluation criteria.*

Despite the effectiveness of the previously developed optimization method and efficient optimization method for optimal design of reconfigurable products, dependency relations among descriptions of design/process candidates and evaluation measures were not considered for identification of importance weights of design/process nodes. To solve this problem, a new method based on analytic network process, called extended analytic network process (EANP), has been introduced in this research to model the dependency relations among design/process nodes in the design/process AND-OR tree/graphs and evaluation criteria.

- The new extended analytic network process (EANP) method can be used to achieve better initial evaluations to design/process nodes which truly reflect the contributions of these nodes to the optimal solutions when the dependency relations among these nodes and evaluation criteria have to be considered.
- A new method has been developed to identify the importance weights of design/process nodes using multiple evaluation criteria. Since these evaluation criteria are similar to the evaluation measures used in detailed optimization process, the identified importance weights can better reflect their contributions to the optimal solutions. When these importance weights are used for pruning of the design/process tree/graphs and initial rankings of design/process candidates, better optimal designs of reconfigurable products can be achieved.

7.2 Future Work

Despite the effectiveness of the developed approach for optimal design of reconfigurable product considering product configurations and reconfiguration processes, a number of issues need to be further addressed in our future research. The major issues in our future work are summarized as follows.

(1) Development of industrial applications using the developed approach is required.

Although the feasibility and effectiveness of the developed approach for optimal design of reconfigurable product have been demonstrated by case studies, real industrial applications need to be implemented using the newly developed approach for optimal design of reconfigurable products considering product configurations and reconfiguration processes.

In addition to industrial applications considering both design configurations and reconfiguration processes, the applications with large numbers of design candidates and reconfiguration process candidates, and applications with dependency relations among design/process descriptions and evaluation criteria should also be developed.

(2) An effective computer environment is needed for the optimal design of reconfigurable product considering both the product configurations and reconfiguration processes.

In our current research, only specific computer programs for the optimal design of the case study reconfigurable products have been implemented. Since the developed optimal reconfigurable product design approach can be employed to develop various applications, a generic computer environment is required for modelling, evaluation, and optimization of various reconfigurable products.

In this computer environment, first generic design/process AND-OR tree/graphs are defined by users. When large numbers of design/process candidates can be created, contributions of the design/process nodes to the optimal design need to be defined by importance weights using the simple heuristics or through the new extended analytic network process method, such that the less important nodes are cut off from the AND-OR tree/graphs. Specific design candidates and reconfiguration process candidates have been generated automatically from the generic AND-OR tree/graphs. Optimization is then conducted to identify the best reconfigurable product design.

(3) *Extension of the developed method for optimal design of reconfigurable products that can be adapted during their utilization stages is needed.*

In our currently developed approach, the reconfigurable product can only be changed for achieving multiple planned functions during its utilization stage. The reconfigurable product cannot be changed and adapted to achieve the unplanned functions which may be raised during its utilization stage. The recent advances in adaptable product design approach (Gu, Xue, and Nee 2009) provide opportunity to design new products that can not only be changed into planned configurations to achieve pre-defined functions but also be adapted into unplanned configurations to achieve new functions.

In this work, new methods will be developed for modeling of reconfigurable and adaptable products. Moreover, new evaluation measures considering product adaptability, such as extendibility of functions, upgradeability of modules, and customizability of components, will be developed for identification of the optimal design.

References

Abdi, M. R. (2009). Fuzzy multi-criteria decision model for evaluating reconfigurable machines. *International Journal of Production Economics,* **117**(1), 1-15.

Aguilar, A., Roman-Flores, A., and Huegel, J. C. (2013). Design, refinement, implementation and prototype testing of a reconfigurable lathe-mill. *Journal of Manufacturing Systems,* **32**(2), 364-371.

Andersen, A. L., Brunoe, T. D., Nielsen, K., and Rösiö, C. (2017). Towards a generic design method for reconfigurable manufacturing systems – analysis and synthesis of current design methods and evaluation of supportive tools. *Journal of Manufacturing Systems,* **42**, 179-195.

Arora, J. S. (1989). *Introduction to Optimum Design.* New York, NY: McGraw-Hill, Inc.

Azulay, H., Mills, J. K., and Benhabib, B. (2014). A multi-tier design methodology for reconfigurable milling machines. *Journal of Manufacturing Science and Engineering,* **136**(4).

Barbarino, S., Bilgen, O., Ajaj, R. M., Friswell, M. I., and Inman, D. J. (2011). A review of morphing aircraft. *Journal of Intelligent Material System and Structures,* **22**(9), 823-877.

Bensmaine, A., Dahane, M., and Benyoucef, L. (2013). A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment. *Computers and Industrial Engineering,* **66**(3), 519-524.

Bi, Z. M., Lang, S. Y., Shen, W., and Wang, L. (2008). Reconfigurable manufacturing systems: the state of the art. *International Journal of Production Research,* **46**(4), 967-992.

Bi, Z. M., and Wang, L. (2009). Optimal design of reconfigurable parallel machining systems. *Robotics and Computer-Integrated Manufacturing,* **25**(6), 951-961.

Blum, C. (2005). Ant colony optimization: introduction and recent trends. *Physics of Life Reviews,* **2**(4), 353-373.

Bonvoisin, J., Halstenberg, F., Buchert, T., and Stark, R. (2016). A systematic literature review on modular product design. *Journal of Engineering Design,* **27**(7), 488-514.

Buede, D. M. (2009). *The Engineering Design of Systems: Models and Methods* (2nd ed.). Hoboken, NJ: John Wiley & Sons, Inc.

Chatzopoulos, C. G. (2014). Flow customizer: an algorithm to design lean-flow production systems for mass customization. *International Journal of Industrial Engineering and Management*, **5**(4), 179-194.

Chen, L., Xi, F. J., and Macwan, A. (2005). Optimal module selection for preliminary design of reconfigurable machine tools. *Journal of Manufacturing Science and Engineering*, **127**(1), 104-115.

Dhupia, J., Powalka, B., Katz, R., and Ulsoy, A. G. (2007). Dynamics of the arch-type reconfigurable machine tool. *International Journal of Machine Tools and Manufacture*, **47**(2), 326-334.

Du, S., Yu, D., Hu, Y., and Huang, Y. (2009). A component model and layered system architecture for reconfigurable CNC systems. *Proceedings of the IEEE International Conference on Information and Automation, ICIA'09,* Zhuhai and Macau, China.

Edelkamp, S., and Schroedl, S. (2011). *Heuristic Search: Theory and Applications*. Waltham, MA: Elsevier, Inc.

Ersal, T., Stein, J. L., and Louca, L. S. (2004). A modular modeling approach for the design of reconfigurable machine tools. *Proceedings of the ASME International Mechanical Engineering Congress and Exposition*, Anaheim, CA, USA.

Feczko, J., Manka, M., Krol, P., Giergiel, M., Uhl, T., and Pietrzyk, A. (2015). Review of the modular self-reconfigurable robotic systems. *Proceedings of the 10th International Workshop on Robot Motion and Control (RoMoCo)*, Poznan, Poland.

Gadalla, M., and Xue, D. (2017a). Recent advances in research on reconfigurable machine tools: a literature review. *International Journal of Production Research*, **55**(5), 1440-1454.

Gadalla, M., and Xue, D. (2017b). Optimal design of a reconfigurable machine tool considering machine configurations and configuration changes. *Proceedings of the 27th CIRP Design Conference – Complex Systems Engineering and Development*, Cranfield, UK.

Gadalla, M., and Xue, D. (2017c). An approach to identify the optimal configurations and reconfiguration processes for design of reconfigurable machine tools. *International Journal of Production Research* (Accepted). doi:10.1080/00207543.2017.1406674.

Gershenson, J. K., Prasad, G. J., and Zhang, Y. (2003). Product modularity: definitions and benefits. *Journal of Engineering Design*, **14**(3), 295-313.

Goyal, K. K., Jain, P. K., and Jain, M. (2012). Optimal configuration selection for reconfigurable manufacturing system using NSGA II and TOPSIS. *International Journal of Production Research*, **50**(15), 4175-4191.

Goyal, K. K., Jain, P. K., and Jain, M. (2013). A novel methodology to measure the responsiveness of RMTs in reconfigurable manufacturing system. *Journal of Manufacturing Systems*, **32**(4), 724-730.

Gu, P., Xue, D., and Nee, A. Y. C. (2009). Adaptable design: concepts, methods and applications. *Proceedings of the Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacture*, **223**(11), 1367-1387.

Haldaman, J., and Parkinson, M. B. (2010). Reconfigurable products and their means of reconfiguration. *Proceedings of the ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference,* Montreal, Canada.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, **4**(2), 100-107.

Ho, W. (2008). Integrated analytic hierarchy process and its applications: a literature review. *European Journal of Operational Research*, **186**(1), 211-228.

Ho, W., Xu, X., and Dey, P. K. (2010). Multi-criteria decision making approaches for supplier evaluation and selection: a literature review. *European Journal of Operational Research*, **202**(1), 16-24.

Hong, G., Dean, P., Yang, W., Tu, Y. L., and Xue, D. (2010). Optimal concurrent product design and process planning based on requirements of individual customers in one-of-a-kind production. *International Journal of Production Research.* **48**(21). 6341-6366.

Hong, G., Hu, L., Xue, D., Tu, Y. L., and Xiong, Y. L. (2008). Identification of the optimal Product configuration and parameters based on individual customer requirements on performance and costs in one-of-a-kind production. *International Journal of Production Research*, **46**(12), 3297-3326.

Hwang, C. L., and Yoon, K. (1981). Methods for multiple attribute decision making. In *Multiple Attribute Decision Making*, Hwang, C. L., and Yoon, K. (Eds.). Berlin, Heidelberg: Springer, Inc., 58-191.

Ic, Y. T., Yurdakul, M., and Eraslan, E. (2012). Development of a component-based machining centre selection model using AHP. *International Journal of Production Research*, **50**(22), 6489-6498.

Ishizaka, A., and Labib, A. (2009). Analytic hierarchy process and expert choice: benefits and limitations. *OR Insight*, **22**(4), 201-220.

Ishizaka, A., and Labib, A. (2011). Review of the main developments in the analytic hierarchy process. *Expert Systems with Applications*, **38**(11), 14336-14345.

Koczkodaj, W. W., and Szybowski, J. (2015). Pairwise comparisons simplified. *Applied Mathematics and Computation*, **253**, 387-394.

Koren, Y., Hu, S. J., Gu, P., and Shpitalni, M. (2013). Open-architecture Products. *CIRP Annals - Manufacturing Technology*, **62**(2), 719-729.

Koren, Y., and Kota, S. (1999). Reconfigurable Machine Tool. US Patent 5,943,750.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA: The MIT Press.

Lawler, E. L. (1976). *Combinatorial Optimization: Networks and Matroids*. Mineola, NY: Holt Rinehart and Winston, Inc.

Liang, S. Y., and Shih, A. J. (2015). Machine tool components. In *Analysis of Machining and Machine Tools*, Liang, S. Y., and Shih, A. J. (Eds.). New York, NY: Springer, Inc., 63-85.

Liu, W., and Liang, M. (2008). Multi-objective design optimization of reconfigurable machine tools: a modified fuzzy-Chebyshev programming approach. *International Journal of Production Research*, **46**(6), 1587-1618.

Lorenzer, T., Weikert, S., Bossoni, S., and Wegener, K. (2007). Modelling and evaluation tool for supporting decisions on the design of reconfigurable machine tools. *Journal of Manufacturing Systems,* **26**(3), 167-177.

Majija, N., Mpofu, K., and Modungwa, D. (2013). Conceptual development of modular machine tools for reconfigurable manufacturing systems. In *Advances in Sustainable and Competitive Manufacturing Systems*, Azevedo, A. (Ed.). Berlin, Heidelberg: Springer, Inc., 467-478.

Marin, F. B., Constantin, I. C., Marinescu, V., Banu, M., and Epureanu, A. (2015). Machine tool configuration and analysis on the base of virtual interactive prototyping. *International Journal on Interactive Design and Manufacturing,* **9**(1), 77-82.

Mehrabi, M. G., Ulsoy, A. G., and Koren, Y. (2000). Reconfigurable manufacturing systems: key to future manufacturing. *Journal of Intelligent Manufacturing*, **11**(4), 403-419.

Moon, Y. M. (2006). Reconfigurable machine tool design. In *Reconfigurable Manufacturing Systems and Transformable Factories*, Westkämper, E. (Ed.). Berlin, Heidelberg: Springer, Inc., 111-140.

Mori, M., and Fujishima, M. (2009). Reconfigurable machine tools for a flexible manufacturing system. In *Changeable and Reconfigurable Manufacturing Systems*, ElMaraghy, H.A. (Ed.). London: Springer, Inc., 101-109.

Mpofu, K., and Tlale, N. S. (2012). Multi-level decision making in reconfigurable machining systems using fuzzy logic. *Journal of Manufacturing Systems*, **31**(2), 103-112.

Padayachee, J., and Bright, G. (2012). Modular machine tools: design and barriers to industrial implementation. *Journal of Manufacturing Systems*, **31**(2), 92-102.

Portman, V. T., Chapsky, V. S., Shneor, Y., and Ayalon, E. (2015). Machine stiffness rating: characterization and evaluation in design stage. *Proceedings of the CIRP 25th Design Conference Innovative Product Creation*, Haifa, Israel.

Pritschow, G., Altintas, Y., Jovane, F., Koren, Y., Mitsuishi, M., Takata, S., van Brussel, H., Weck, M., and Yamazaki, K. (2001). Open controller architecture – past, present and future. *CIRP Annals - Manufacturing Technology*, **50**(2), 463-470.

Riba, R. C., Pérez, R. R., Ahuett, G. H., Jorge, L. S. A., Domínguez, M. D., and Molina, G. A. (2006). Metrics for evaluating design of reconfigurable machine tools. In *Cooperative Design, Visualization, and Engineering*, Luo, Y. (Ed.). Berlin, Heidelberg: Springer, Inc., 234-241.

Saaty, T. L. (1977). A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, **15**(3), 234-281.

Saaty, T. L. (1981). The analytic hierarchy process: scenarios, priorities and cost/benefits for the Sudan transport plan. *Operational Research*, **81**, 161-175.

Saaty, T. L. (1990). How to make a decision: the analytic hierarchy process. *European Journal of Operational Research*, **48**(1), 9-26.

Saaty, T. L. (1999). Fundamentals of the analytic network process. *Proceedings of the 5th International Symposium on the Analytic Hierarchy Process,* Kobe, Japan.

Saaty, T. L., and Takizawa, M. (1986). Dependence and independence: from linear hierarchies to nonlinear networks. *European Journal of Operational Research*, **26**(2), 229-237.

Saaty, T. L., and Vargas, L. (2006). *Decision Making with the Analytic Network Process, Economics, Political, Social and Technological Applications with Benefits, Opportunities, Costs, and Risks*. New York, NY: Springer, Inc.

Sabuncuoglu, I., and Bayiz, M. (1999). Job shop scheduling with beam search. *European Journal of Operational Research*, **118**(2), 390-412.

Samy, S. N., and ElMaraghy, H. (2010). A model for measuring products assembly complexity. *International Journal of Computer Integrated Manufacturing*, **23**(11), 1015-1027.

Shneor, Y., and Portman, V. T. (2010). Stiffness of 5-axis machines with serial, parallel, and hybrid kinematics: evaluation and comparison. *CIRP Annals – Manufacturing Technology*, **59**(1), 409-412.

Siddiqi, A., de Weck, O. L., and Iagnemma, K. (2006). Reconfigurability in planetary surface vehicles: modeling approaches and case study. *Journal of the British Interplanetary Society*, **59**(12), 450-460.

Singh, V., Skiles, S. M., Krager, J. E., Wood, K. L., Jensen, D., and Sierakowski, R. (2009). Innovative in design through transformation: a fundamental study of transformation principles. *Journal of Mechanical Design*, **131**(8).

Slocum, A. H. (1992). System design consideration. In *Precision Machine Design*, Slocum, A. H. (Ed.). Dearborn, MI: Society of Manufacturing Engineers, 304-420.

Sofla, A. Y. N., Meguid, S. A., Tan, K. T., and Yeo, W. K. (2010). Shape morphing of aircraft wing: status and challenges. *Materials & Design*, **31**(3), 1284-1292.

Soh, S. L., Ong, S. K., and Nee, A. Y. (2016). Design for assembly and disassembly for remanufacturing. *Assembly Automation*, **36**(1), 12-24.

Son, H., Choi, H. J., and Park, H. W. (2010). Design and dynamic analysis of an arch-type desktop reconfigurable machine. *International Journal of Machine Tools and Manufacture*, **50**(6), 575-584.

Spicer, P., Koren, Y., Shpitalni, M., and Yip-Hoi, D. (2002). Design principles for machining system configurations. *CIRP Annals - Manufacturing Technology*, **51**(1), 275-280.

Spicer, P., Yip-Hoi, D., and Koren, Y. (2005). Scalable reconfigurable equipment design principles. *International Journal of Production Research*, **43**(22), 4839-4852.

Stanewsky, E. (2001). Adaptive wing and flow control technology. *Progress in Aerospace Sciences*, **37**(7), 583-667.

Walczyk, D. F., Lakshmikanthan, J., and Kirk, D. R. (1998). Development of a reconfigurable tool for forming aircraft body panels. *Journal of Manufacturing Systems*, **17**(4), 287-296.

Weaver, J. M., Wood, K. L., and Jensen, D. (2008). Transformation facilitators: a quantitative analysis of reconfigurable products and their characteristics. *Proceedings of the ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference*, Brooklyn, NY, USA.

Weiss, P. (2003). Wings of change. *Science News*, **164**(23), 359-365.

West, D. B. (2001). Trees and distance. In *Introduction to Graph Theory* (2nd ed.), West, D. B. (Ed.). Upper Saddle River, NJ: Pearson Prentice Hall, Inc., 83-142.

Wilt, C. M., Thayer, J. T., and Ruml, W. (2010). A comparison of greedy search algorithms. *Third Annual Symposium on Combinatorial Search*, Atlanta, GA, USA.

Xie, N., Li, A., and Xue, W. (2012). Cooperative optimization of reconfigurable machine tool configurations and production process plan. *Chinese Journal of Mechanical Engineering*, **25**(5), 982-989.

Xue, D., Cheing, S. Y., and Gu, P. (2006). Configuration design considering the impact of design changes on downstream processes based upon the axiomatic design approach. *Journal of Engineering Design*, **17**(6), 487-508.

Xue, D., Yang, H., and Tu, Y. L. (2006). Modeling of evolutionary design database. *Journal of Computing and Information Science in Engineering*, **6**(1), 22-32.

Yin, Y. H., and Xie, J. Y. (2011). Reconfigurable manufacturing execution system for pipe cutting. *Enterprise Information Systems*, **5**(3), 287-299.

Yin, Y. H., Xie, J. Y., Xu, L. D., and Chen, H. (2012). Imaginal thinking-based human-machine design methodology for the configuration of reconfigurable machine tools. *IEEE Transactions on Industrial Informatics*, **8**(3), 659-668.

Yoon, J. S., Son, S. E., Song, W. J., Kim, J., and Kang, B. S. (2014). Feasibility study on flexibly-reconfigurable roll forming process for multi-curved surface of sheet metal. *International Journal of Precision Engineering and Manufacturing*, **15**(6), 1069-1074.

Zhang, D., and Bi, Z. (2011). Development of reconfigurable parallel kinematic machines using modular design approach. *Proceedings of the Canadian Engineering Education Association*, Toronto, ON, Canada.

Zhang, J., Chen, Y., Xue, D., and Gu, P. (2014). Robust design of configurations and parameters of adaptable products. *Frontiers of Mechanical Engineering*, **9**(1), 1-14.

Zhang, J., Xue, D., and Gu, P. (2015). Adaptable design of open architecture products with robust performance. *Journal of Engineering Design*, **26**(1-3), 1-23.

Zhao, C., Peng, Q., and Gu, P. (2015). Development of a paper-bag-folding machine using open architecture for adaptability. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, **229**(1), 155-169.

Zheng, W., and Zhong, H. (2012). *Reconfigurable Machine Tools Design Methodology*. Master Thesis, Royal Institute of Technology, Stockholm, Sweden.

Publications

Journal Publications

1. Gadalla, M., and Xue, D. An extended analytic network process (EANP) method for optimal design of reconfigurable products considering dependency relations among descriptions of design/process candidates and evaluation criteria. Submitted to *Research in Engineering Design*.
2. Gadalla, M., and Xue, D. An efficient optimization approach for design of reconfigurable Products considering large numbers of design candidates and reconfiguration process candidates. Submitted to *Journal of Mechanical Design*.
3. Gadalla, M., and Xue, D. (2017). An approach to identify the optimal configurations and reconfiguration processes for design of reconfigurable machine tools. *International Journal of Production Research* (Accepted). doi:10.1080/00207543.2017.1406674.
4. Gadalla, M., and Xue, D. (2017). Recent advances in research on reconfigurable machine tools: a literature review. *International Journal of Production Research*, **55**(5), 1440-1454.

Conference Publication

1. Gadalla, M., and Xue, D. (2017). Optimal design of a reconfigurable machine tool considering machine configurations and configuration changes. *Proceedings of the 27th CIRP Design Conference – Complex Systems Engineering and Development*, Cranfield, UK.