PROJECT TASK – MEMBER ASSIGNMENT USING DESIGN STRUCTURE MATRIX AND GENETIC ALGORITHM IN CONCURRENT ENGINEERING PROJECT MANAGEMENT

by

Lukasz Maciej Mazur

A thesis submitted in partial fulfillment of the requirements
of the requirements for the degree

of

Master of Science

in

Industrial and Management Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2005

APPROVAL

of a thesis submitted by

Lukasz Maciej Mazur

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

Chair of Committee:

Dr. Shi - Jie (Gary) Chen

Approved for the Department of Mechanical and Industrial Engineering:

Dr. Jay R. Conant

Approved for the College of Graduate Studies:

Dr. Bruce R. McLeod

STATEMENT OF PERMISSION TO USE

Lukasz Mazur

4/1/2005

ACKNOWLEDGEMENTS

To Dr. Shi - Jie (Gary) Chen for his valuable guidance, support, and continuous encouragement throughout the entire work on this thesis. Without his belief, patience as well as valuable comments and criticism I would not be able to complete this thesis.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In concurrent engineering, project tasks generally require the establishment of multifunctional teams to simultaneously consider various activities throughout the entire product life cycle. Team members from different functional departments of the company interact in every phase of development activities to design products and processes concurrently. This concurrent strategy increases the complexity of product development and design processes and makes teams difficult to organize. Without effective task coordination and team organization, the lack of communication and cooperation among team members in a large group of tasks could seriously delay the project completion. This research provides an integrated solution to overcome these difficulties. This research aims to model both project tasks and team members for the task-member assignments. To accomplish this, we develop an integrated framework that includes three major components: a project task model, a team member model and a task-member assignment model. The project task model optimizes the complex task structure using a Genetic Algorithm (GA), while Design Structure Matrix (DSM) identifies the three major project task types: independent, dependent, and interdependent. The team member model provides a quantitative representation for three important team member characteristics, namely functional knowledge, teamwork capability and working relationship. Analytic Hierarchy Process (AHP) and personality profiling using Myers-Briggs Type Indicator (MBTI) are used to obtain ratings of team member characteristics. According to the project task structure, quantified team member characteristics, and each member's workload schedule, the task-member assignment model accomplishes the ultimate goal of this research – assigning the right team members to the right tasks at the right time. The effectiveness of the developed methodology is demonstrated by an illustrative example.

CHAPTER 1

INTRODUCTION

Background and Motivation

Demanding global marketplace, intense competition forces, and numerous design variables and constraints are only a few critical aspects that a company has to face and to fight for survival in the modern industry. For example, the design of an automobile can involve the coordination of hundreds or even thousands of engineers from different departments at different locations who make more than a million design decisions over months or years (Eppinger *et al*., 1990). The difficulties in designing and planning operations for a complex project do not simply arise from their engineering complexity, but also stem from the organizational sophistication necessary to manage the design process (Chen and Lin, 2002, 2003). Gaining competitive advantage in a global environment means continually reshaping the organization to maximize strengths, address threats, and increase speed. The use of teams has become a common way of accommodating these challenges. The teaming approach can draw talent quickly from different functions, locations, and organizations creating a higher potential for successful project performance. The goal of team formation is to leverage intellectual capital and apply it as quickly as possible. The methods that organizations use to manage the teaming process can mean the difference between success and failure (Duarte and Snyder, 1999). Traditionally, design, manufacturing and operation management activities have been carried out sequentially. Each function is concerned only with meeting specific requirements set for them. No consideration is given to downstream activities in the product life cycle. This "throw-over-the-wall" approach to design and development can result in increased engineering changes, and thus higher cost to the business. To maintain a competitive edge, industry in recent years has embraced the concept of Concurrent Engineering (CE). CE is a systematic approach to integrated design of products and their related processes, including manufacturing and support. Considering the requirements of downstream activities in the product life cycle early in design will reduce the total cost of product development.

Complex products or projects with long development times always involve multiple disciplines and a great deal of effort. Especially in a concurrent engineering environment, the design needs to simultaneously consider various downstream activities throughout the entire product life cycle, in addition to meeting the functional requirements of the products. To accomplish this, tasks in the concurrent engineering environment generally involve multifunctional design teams (Stinson, 1990; Hoedemaker *et al*. 1995; Smith, 1997). Team members from different functional departments of the company interact in every phase of development tasks to design products and processes concurrently. However, the increasing complexity of the product development and design process due to the nature of concurrent strategy makes it difficult to organize the tasks and teamwork. This motivates this study to develop a methodology for assigning the right team members to the right tasks at the right time in the concurrent engineering environment. However, the question is "*how do we assign the right members to the right tasks at the right time in the concurrent engineering environment?*" For example, a project that contains 1000 tasks with independent, dependent, and interdependent relationships among each other and requires the expertise from various functional departments and organizations will be a big challenge to project managers for project task coordination and team organization.

Objectives of the Research

The major objective of this thesis is to develop a methodology for assigning the right team members to the right tasks at the right time with the following three models:

(1) To develop a project task model in order to identify the structure of the entire design process by revealing the independent, dependent, and interdependent tasks and to optimally coordinate the tasks' sequences in the interdependent task groups based on tasks cost, tasks time and tasks relationship strengths. This model is an extension of the project task model developed by Chen and Lin (2002, 2003) and Rohatynski, Kielec and Sasiadek (Rohatynski *et al.*, 2002; Rohatynski and Kielec, 2001, 2002).

(2) To build a team member model to represent member's workload schedule and characteristics that are important for team organization, such as their multifunctional knowledge, teamwork capabilities, and working relationship. This model is an extension of the team member model developed by Chen and Lin (2004).

(3) To formulate a quantitative task-member assignment model based on project task model and team member model for choosing the right team members for the right

tasks at the right time with respect to independent, dependent, and interdependent tasks as well as task time, task cost and/or team performance.

Research Framework

The research framework of this thesis is shown in Figure 1.

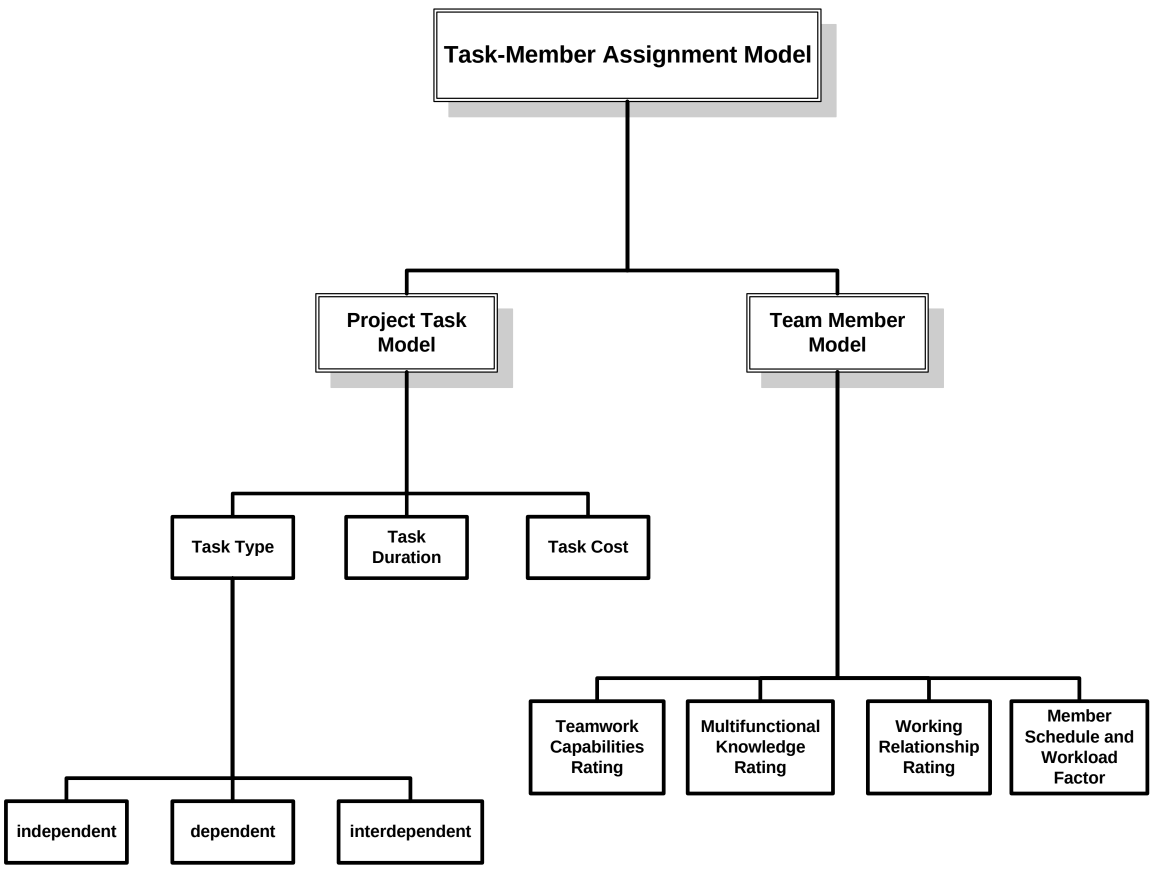

Figure 1. Research Framework

(1) Project Task Model: To understand the complex task structure of the project, the analysis of the sequence and dependency of project tasks is necessary. The project

task model contains three information needed for project task coordination: task type, task duration, and task cost. There are three basic task types in the project: independent, dependent and interdependent tasks. Task Duration is the time required to complete a specific task, and is dependent on the capabilities of the member who is assigned to the task. Task Cost is associated with the execution of the task (not including the cost of the members assigned to the task). Design Structure Matrix (DSM) and Genetic Algorithm (GA) are used for project sequencing and task flow optimization in Project Task Model.

(2) Team Member Model: To assign the right team member to the right task at the right time, this model contains the following four components: Multifunctional Knowledge Rating, Teamwork Capability Rating, Working Relationship Rating, and Member's Workload Schedule. Multifunctional Knowledge Rating is a rating revealing each member's knowledge level to each functional department in the organization. For example, member A can be an expert (rating = 1.00) in Marketing Department, and also 0.5 in Sales Department stating that member A has 1 year of experience in sales. Teamwork Capability Rating is based on: Teamwork Experience, Communication Skills, and Flexibility in Job Assignment. Since the team members work closely, their collegiality (Working Relationship Rating) also needs to be measured. This is done by Myers–Briggs type indicator (MBTI), a popular personality test that has been widely used by both academia and industry. Member's Workload Schedule is to present the utilization of each member's time, expressed as a percentage throughout the entire project cycle, so when a member is available for a task will be known.

Analytic Hierarchy Process (AHP) and Absolute Measurements are used to quantify each member's multifunctional knowledge and teamwork capability ratings in this model.

(3) Task-Member Assignment Model: Based on the understanding of project task structure from Project Task Model and team member characteristics from Team Member Model, this assignment model will accomplish the ultimate research goal – assign the right team members to the right task at a right time. A genetic algorithm will be developed in this model to serve as the task-member assignment engine with respect to independent, dependent, and interdependent tasks as well as task time, task cost and/or team performance.

CHAPTER 2

LITERATURE REVIEW

Task planning and team organization in concurrent engineering are the two main subjects under study for this paper. In the first section of this literature review, concurrent engineering basic concepts as well as the teaming approaches in the concurrent engineering environment are studied as a foundation for this thesis. Second, the decision analysis using Analytic Hierarchy Process (AHP) and absolute measurements are presented as the tools used for quantitative rating of members characteristics. Finally, the literature review on Design Structure Matrix (DSM) and Genetic Algorithms (GA) is provided. DSM and GA are used in this thesis as a data representation tool and search algorithm respectively.

Concurrent Engineering

In the traditional functional organization, people are grouped principally by each functional discipline. In the functional organization, the steps of product development process from design to fabrication are sequential (Orady and Shareef, 1993). This can be viewed as sequential engineering - a series of design processes in which one process occurs only after a particular stage of process has been completed. First, the design engineer takes a conceptual definition and works it into a detailed design, usually without ever consulting the manufacturing engineer. Once a design is tested and verified, it is "thrown over the wall" to the manufacturing department where the design is checked for manufacturability and testability. O'Grady and Young (1991) remarked that companies have realized that such design approaches are unacceptable in an increasingly competitive marketplace and are transforming their serial design strategy into one that stresses the parallel evolution of a product's design beginning with conception. This strategy is called Concurrent Engineering. In recent years many authors (Holt and Barnes, 2002; Barnes, 2000; Barnes *et al*., 2003) have shown that the concepts of CE could improve the productivity and efficiency as well as minimize the total cost and time of the projects. For example, Haas and Sinha (2004) presented a case study on Airbus Concurrent Engineering (ACE) showing that CE approach is a good fit for the project environment in which global competition, short innovation cycles and cost pressure force companies to constantly increase the efficiency of their internal processes.

Prasad (1996) commented that the most commonly referred to definition of concurrent engineering is that of Winner *et al*. (1988):

"Concurrent engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life-cycle from conception through disposal, including quality, cost, schedule, and user requirements."

If a product is designed by concurrent engineering, possible design errors can be foreseen and thus corrected early. Consequently, the total cost and time for developing a product will be significantly reduced. In practice, however, it is an arduous task to consider all aspects of the product manufacturing cycle at the early design stage. This is primarily due to three major sources of difficulty: the complexity of the design process itself, the large volume and large variety of life cycle information, and the separation of life cycle functions (O'Grady *et al*., 1991). To overcome these difficulties, a number of methods have been developed in concurrent engineering, including handbooks, checklists, computerized tools (i.e., Quality Function Deployment (QFD), Design for Manufacturing (DFM), and Design for Assembly (DFA), and multifunctional teams. In order to compromise various design conflicts and integrate process information during the whole product development cycle, building multifunctional design teams is especially important among these approaches.

Just as sequential engineering can be compared to the functional machine layout, concurrent engineering using multifunctional teaming approach can be viewed as a cellular layout in which different functions of machines are grouped together to produce a part from beginning to end within a single machine cell. Thus, a project task to be done by a multifunctional team is similar to finishing a part in a machine cell. A multifunctional team dealing with an arrival task is like a cell processing an incoming part. A company organized into various multifunctional teams is comparable to a cellular layout factory consisting of many machining cells. Because engineers with different backgrounds are invited to a product development team, this prevents iterative redesign work among upstream and downstream areas in sequential engineering. But, as with common problems occurring in the cellular layout, concurrent engineering inherently introduces a more complicated organization to a company which is more difficult to manage. Concurrent engineering aims to incorporate all downstream factors for problem solving during the upstream phase. Simply stated, concurrency is the major force of concurrent engineering (Lee, 1992; Prasad, 1996). However, the concurrent strategy increases the complexity of the design process and makes it more difficult to manage (Kusiak and Wang 1993b). Therefore, developing a way to simplify the project tasks by managing a variety of multifunctional teams in the concurrent engineering environment is critical.

Teaming Approaches in Concurrent Engineering

Complex projects in concurrent engineering environment require a group of highly specialized team members who also have a good understanding of requirements from other design tasks. From a concurrent engineering perspective, concurrency takes two forms (Blackburn *et al*., 1994; Hoedemaker *et al*., 1995):

1) Concurrency in time: This refers to activities that are performed in parallel by different people or groups, such as simultaneous design activities.

2) Concurrency in information: This refers to the integrated or team approach in which all the concerns of the different functions are addressed through shared information.

Concurrent engineering may include many issues, but to most people concurrent engineering is teamwork and how to work together in a more efficient way (Lawson and Karandikar, 1994; Killander, 1995). One of the most important goals in concurrent engineering is assigning (distributing) the right team members (information) to the right tasks (places) at the right time. Here, a team member represents an object holding a certain domain of knowledge and a task is like a place where team members or information are pulled together to work together. To be successful, concurrent engineering requires supporting changes in the organizational environment surrounding a team. Concurrent engineering team formation requires identifying the characteristics and the interactions among tasks, team members, and tasks versus team members. Prasad (1996) illustrates three different types of team formation based on the required skill sets:

1) Multidisciplinary team: Requires a number of teams or team members who specialize in different aspects of product development.

2) Interdisciplinary team: The team possesses some portion of a skill set (partial knowledge) belonging to two or more multidisciplinary areas that are needed for the product development.

3) Cross-disciplinary team: One team has profound knowledge of what the other teams (multidisciplinary or interdisciplinary teams) do.

These three types of team formation indicate some of the basic conditions for organizing the teams and provide some necessary structural characteristics for these teams. But the main intention is still the same; that is, to build up a well-organized design team to tackle various tasks in the same environment.

Chen and Lin (2004) presented a very interesting approach using multifunctional groups composed of members representing many aspects of the product life cycle (i.e., marketing, sales, research and development, manufacturing, purchasing, testing, quality assurance, and service) as a very good fit for the concurrent engineering environment. Their research showed that the main advantage of a multifunctional team is that it brings together the knowledge of various work areas to develop sound solutions in design. When managed effectively, the multifunctional design team should offer improved communication, stronger identification and commitment to the assigned tasks, and a focus on cross-functional problem solving (Clark and Wheelwright, 1992; Chen and Lin, 2004). Multifunctional teams provide a mechanism for bringing together people with different but complementary skills and tying them to designing and manufacturing products. However, as products become more complex, so does the process of designing them. Teams that develop these increasingly complex products are becoming larger and more complex themselves. This will inevitably degrade team performance when the team size becomes too large to handle.

Team Member Characteristics for a Successful Multifunctional Team

Many different criteria for selecting members to multifunctional teams such as functional expertise, personal attributes, communication skills, and motivation through job assignment have been reported (Blackwell 1986). For example, Chung and Guinan (1994) suggest that teams with more experienced members perform better than those with less experienced members. They concluded that in small teams that consist of experienced members, participation is strongly associated with team performance. Allen (1986) concluded that many studies have shown a positive relationship between project performance and communication within each project team. Therefore, not only the expert knowledge from each functional member needs to be considered, but the communication skills of each team member should also be taken into account. Safoutin and Thurston (1993) pointed out the importance of communication in a design team by emphasizing that most design failures are caused by communication errors at key decision points. In addition, Hiliard (1993) showed that team members with good communication skills will facilitate agreement among team members and hence improve the team performance. Griffin and Hauser (1992) concluded that recent scientific evidence suggests that new-product development teams are more successful if their members communicate with one another. McKee (1992), Nicholas (1994), Prasad (1998) concluded that to build a successful multifunctional team, the barriers in the subject-matter, the process, as well as cultural have to be avoided. These barriers seriously impede product improvement due to a lack of expertise, burden of excessive information processing and deficiency of adaptation to efficient workflows. In addition, there is growing evidence that team members must possess both individual task skills and teamwork skills in order to perform together successfully (Campion, 1993; Converse, 1991; Hague *et al.,* 2000; Logan, 1993; Nicholas, 1994; Pawar and Sharifi, 1997; Sharifi and Pawar, 1993). Chen and Lin (2004), based on their very comprehensive literature review, proposed five team member characteristics for organizing a successful multifunctional team in which the right team members are selected:

1) Functional expertise (Blackwell, 1986; Taylor, 1986; Converse *et al*., 1991; Clark and Wheelwright, 1992; Hirshhorn and Gilmore, 1992; Mckee, 1992).
2) Teamwork experience (MacDonald, 1986; Sundstrom *et al*., 1990; Chung and Guinan 1994).

3) Communication skills (Allen, 1986; Blackwell, 1986; MacDonald, 1986; Petrie, 1986; Taylor, 1986; Sundstrom *et al*., 1990; Griffin and Hauser, 1992; Hirshhorn and Gilmore, 1992; Mckee, 1992; Prince *et al*., 1992; Hiliard, 1993; Safoutin and Thurston, 1993).
4) Flexibility to job assignment (Blackwell, 1986; Taylor, 1986; Stinson, 1990; Sundstrom *et al*., 1990; Clark and Wheelwright, 1992; Mckee, 1992; Prince *et al*., 1992).
5) Personality traits (Blackwell, 1986; MacDonald, 1986; Petrie, 1986; Taylor, 1986; Sundstrom *et al*., 1990; Prince *et al*., 1992; Trower and Moore, 1996).

To quantify personality traits, Chen's and Lin (2004) used the Myers–Briggs type indicator (MBTI), a personality test that has been widely used by both academia and industry (Baran, 1992; Bayne, 1990; Chance, 1989; Church and Waclawski, 1996; Coe, 1992; Cofsky, 1992; Edgley, 1992; Hildebrand, 1995; Jaffe, 1985; Jung, 1953; Levasseur, 1992; Lyman and Richter, 1995; Maxon, 1986; McClure and Werther, 1993; Moore and Woods, 1987; Weldon, 1995; Zemke, 1992). The MBTI test suggests that knowing one's own preferences and learning about the preferences of others helps individuals to identify their special strengths, to determine the kinds of work they will enjoy, and to work more productively with their colleagues. MBTI has been and continues to be a popular tool for researchers, and has been applied in numerous personal and organizational settings (Myers and McCaulley, 1985). It is believed that differences in personality can significantly affect individual behavior, and subsequently individual and group performance, in an organization. The literature review provided by Chen and

Lin (2004) showed that using the right personality style can help reduce conflict and create improved communication between team members. When people become aware of which types they and their teammates are, they will improve their communication and productivity (McClure and Werther, 1993; Moore and Woods, 1987; Tomal, 1992). As a result, many companies are using personality tests to help evaluate how well prospective employees will fit into their organization (Sunseri and Kostera, 1991). In general MBTI works by characterizing member's personality into four groups with each group consisting of two opposite preferences.

1) Focus of Attention:

- Extrovert (E): Those who relate best to the outer world. They are comfortable in talking and sharing with others. They gain their energy from working with groups.
- Introvert (I): Those who relate best to their inner self. They are comfortable in working quietly alone. They drain their energy from interactions in a group.

2) Seeking Information:

- Sensing (S): Those who rely on facts and reality. They focus on the details. When asked to review a document, they like to find typographical errors and misspellings.
- iNtuitive (N): Those who use intuition, speculation, possibilities and imagination. They focus on the big picture. When asked to review a document, they like to identify problems in how the topic in the document was developed.

3) Decision-Making:

- Thinking (T): Decisions are made by using sound principles, laws, policy and criteria. Thinkers are analytical, logical, and objective.

- Feeling (F): Decisions are made by values, devotion, sympathy, and harmony. Feelers will take the emotions and opinions of others into consideration when making a decision. They have a strong need to maintain harmony within a group.

4) Relationships with the World:

- Judging (J): They are outcome-oriented, regulated, and decisive. They make decisions quickly. Judging members like to get thing settled or come to a closure.
- Perceiving (P): They are process-oriented, flexible, and open-minded. They make decisions slowly. Perceiving members like to get additional information or consider a new possibility.

Every individual owns one preference from each of the above four groups in any combination that then forms the basis for the 16 personality types. For example, a person may be identified as INTP, ENTJ, ESFP, INFJ, or one of the 12 other possible combinations. The most effective teams should have a good combination of personality types. When the identified personality types can complement each other in cooperative work, the team can enjoy a balance to becoming a successful team (Lyman and Richter, 1993). According to Myers (Myers, 1980a), too much opposition makes it hard for people to work together well. She claimed that the best teamwork is usually done by people who differ on one or two preferences to complement each other and have two or three preferences in common to help them understand and communicate with each other. She recommended Sensing and iNtuitive types can be useful to each other, as well as Thinking and Feeling types. This is especially true in teamwork in which a mixed group of SN and TF members is needed for problem solving and decision making. Moreover, Myers (1980b) also suggested that people who differ on Judging and Perceiving preferences would be hard-pressed to understand and predict each other. On the other hand, with the same type of Judging or Perceiving, people will tend to have common interests since they share the same kind of perception, and to consider the same things important since they share similar judgment. Therefore, when selecting team members, the above factors need to be considered. MBTI has been a common battery for personality profiling. This is evidenced by the fact that over three million people a year complete the MBTI test; and nearly 40% of such tests are applied to team building and management development in major corporations (Gardner and Martinko, 1996). It is the most widely used personality test in the U.S., with the corporate arena being the biggest user (Moore and Woods, 1987). Since it is the instrument used to explain the effects that personal preferences have on decision making and problem solving (Guthrie, 1993), the MBTI will continue to be a popular personality profiling instrument for teamwork.

Analytic Hierarch Process (AHP)

The Analytic Hierarchy Process (AHP) method was developed during the early 1970's, and holds both the advantages of the eigen-function evaluation and prioritization of qualitative information (Saaty, 1980). It has been demonstrated to be a suitable method for the selection of functionally most appropriate components of technical systems (Zavbi and Duhovnik, 1996). AHP uses the pairwise comparison technique in a decomposed hierarchy to resolve the complex multi-criteria decision problems quantitatively. The basic steps for using AHP are: 1) Principles of decomposition, 2) Comparative judgments, and 3) Synthesis of priorities. The first step is to disintegrate the unstructured decision problem into components and arrange them into a hierarchy. The hierarchy usually contains three major levels: the overall objective, criteria or sub-goals, and decision alternatives. The decision-maker then begins the comparative judgment procedure to determine the relative importance of the elements in each level. Pairwise comparisons are made on a single criterion without concerning other criteria or elements (Saaty, 1990). Relative priorities for decision alternatives can be evaluated under each specific criterion. The process is carried out at each level by calculating a set of eigen-vectors (or priority-vectors) to determine the overall rankings of competing elements. The larger ranking value, the higher the priority for the element.

AHP has been applied to various areas of multi-criteria decision-making and conflict-solving problems. Partovi and Hopton (1994) introduced the AHP method to inventory management problems for integrating qualitative and quantitative criteria in decision making. An evaluation model developed by Lin and Yang (1996) employed AHP for machine selection to achieve required financial efficiency. Chen and Lin (2002) used AHP to quantify the multifunctional knowledge and teamwork capability ratings of the members. Many other applications such as those presented by Partovi and Burton (1992), Basak (2002), Satty (2004a,b,c,d,e,f), Jyoti and Sudhakar (2001), Sudhakar (2001) have also illustrated the merits of AHP on the ease of use and the effectiveness in dealing with multi-criteria decision problems.

Another method besides AHP is absolute measurements. Forman and Selly (1998) present a very comprehensive review on absolute measurements. They showed that when a model is created using all relative measurements (AHP) the priorities of the objectives, sub-objectives and alternatives are computed by comparing the elements one to another. In contrast, absolute measurements are used to gauge the alternatives against an established scale and not against each other. In everyday life, we often use such absolute measurements. For example, when measuring distance (e.g. kilometers), volume (e.g. liters), and temperature (e.g. degrees Celsius). Therefore, it is natural to create such absolute measurements and derive new scales even where they don't exist. Expert Choice, the software proposed by Satty presents the absolute measurements in the example to establish categories into which letters of recommendation for students fall. First of all, the categories for the scale have to be established. For example: EXCELLENT (E), VERY GOOD (VG), GOOD (G), AVERAGE (A), BELOW AVERAGE (BA), POOR (P), and VERY POOR (VP). Second they have to be prioritized by the user defining how much (for example) EXCELLENT is better than POOR on a scale from 1 to 9. After prioritizing, the categories become the standards or scale for measuring letters of recommendation. Overall, in an absolute measurement approach the objectives are compared against one another as usual using a pre-established scale instead of being compared to one another (so, in this example the categories are compared pairwise against one another as usual, to pre-establish a scale for the letters of recommendation so that they don't have to be later compared one to another). While some scales such as miles or liters are well established and widely recognized, new scales can be often created and customized to fit particular evaluation.

Absolute measurements approach combines the power of the hierarchy and the pairwise comparison process with the capability to rate hundreds or thousands of alternatives. It is used in a wide variety of applications such as college admission decisions, personnel evaluation, and resource allocation. The hierarchy for a ratings model differs from that of a traditional AHP model in that intensities appear at the lowest level of the hierarchy instead of the alternatives. For example, instead of comparing all members to each other at the department, first define multifunctional knowledge rating intensities, such as E, VG, A, BA, P, and VP and then compare them for every department and then use the already scaled intensities as an actual descriptor of members multifunctional knowledge rating.

Forman and Selly (1998) also shows in their book that absolute measurements approach is mostly used for models with a large number of alternatives. However, absolute measurements are also often used in small problems when the evaluation methodology is more familiar to people using it, or when there may are or regulations that prohibit comparing one alternative against another.

Design Structure Matrix (DSM)

There are three basic types of relationships among project tasks: independent, dependent, and interdependent (see Figure 2). To identify these three basic task types inherent in the whole complex process system, Design Structure Matrix (DSM) proposed by Steward is a useful tool (1981a, 1981b).

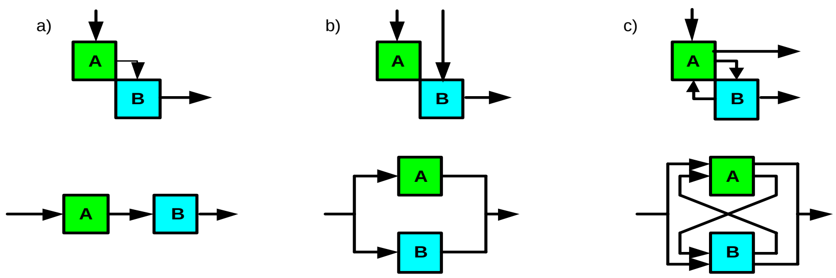

Figure 2. Task Relations: a) dependent, b) independent, c) interdependent.

Much research has demonstrated the effectiveness of DSM in the past (Steward, 1981a; Goldberg, 1989; Rogers, 1989a, 1989b, 1994; Eppinger *et al*., 1990, 1994, 2001a, 2001b; Syswerda, 1990; Gebala and Eppinger, 1991; Kusiak and Wang, 1993a, 1993b, 1993c; Kusiak *et al*., 1994, 1995; McCulley and Bloebaum, 1994, 1996, 1998; Rogers and Bloebaum, 1994; Thomas and Eppinger, 1994; Guo *et al*., 1995; Kusiak and Larson, 1995; Krishnan, 1996; Krishnan *et al*., 1997; McCulley *et al.,* 1997; Yassine *et al.,* 2000, 2001; Ahmadi *at el.,* 2001; Browning, 2001; Cho, 2001; Dong and Whitney 2001; Eppinger, 2001; Eppinger and Salminen, 2001; Chen and Lin 2002, 2003). DSM divides the design project into n individual tasks in an $n \times n$ matrix. Similar to an adjacency matrix, DSM is a square matrix with n rows, n columns, and m non-zero elements, where *n* is the number of nodes and *m* is the number of edges. If there exists an edge from node *i* to node *j*, the value of element *ij* is a unity or a marked sign in the matrix. Otherwise, the value of the element is zero or empty. Information links among individual tasks are revealed clearly by the systematic mapping, regardless of the number of tasks. The advantages of using DSM are: 1) DSM overcomes the size and complexity limitations of digraphs in visualizing the entire project task structure; 2) DSM is easy to understand and able to handle the processes in their entirety; 3) The matrix format is suitable to program and calculate using a computer; and 4) Not only can the entire sequences among project tasks be identified, but also the underlying structure of the project can be analyzed by rearranging the task sequences. Figure 3 shows a 17-task DSM example with 31 task relations/connections including 20 progressive connections (above the diagonal) and 11 feedback connections (below the diagonal).

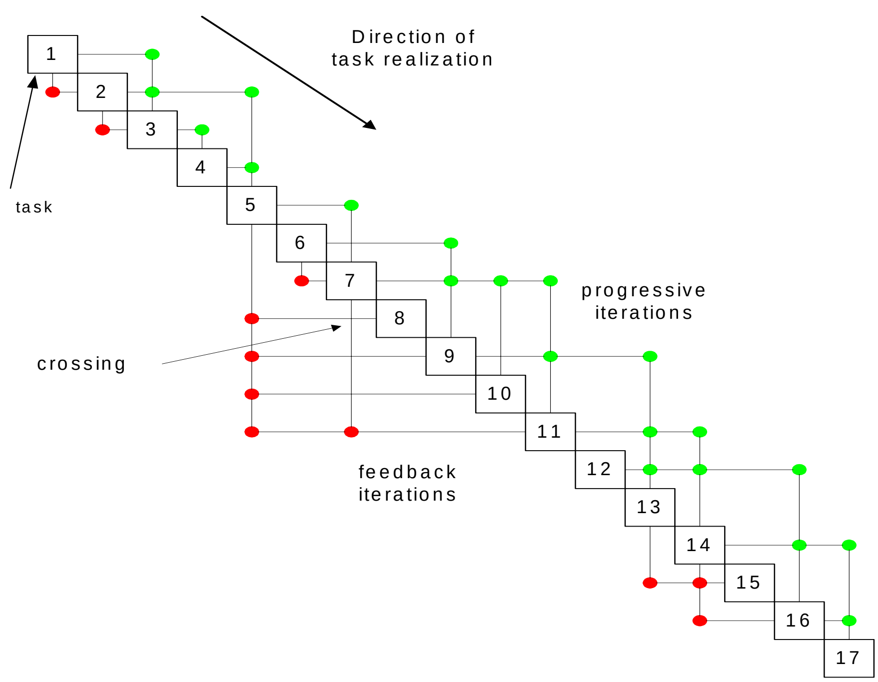

Figure 3. Design Structure Matrix

In recent years, DSM has been used as a management aid as well as an engineering tool to guide the organizational structure of design projects (Sobieszczanski-Sobieski, 1989, 1993; Rogers, 1989; Eppinger, *et al*., 1990, 1994; Kusiak and Wang, 1993a, 1993b; Guo *et al*., 1995; Smith and Eppinger, 1997a, 1997b; Yassine *et al.*, 2000, 2001; Dong and Whitney 2001; Eppinger, 2001; Rohatynski and Kielec, 2001, 2002; Kielec, 2002; Rohatynski *at el*., 2002; Chen and Lin, 2002, 2003). According to Steward's partitioning algorithm, Rogers (1989) developed an expert system tool called DeMAID (Design Manager's Aide for Intelligent Decomposition) that can perform DSM analysis in determining the task sequences and their relationships. DeMAID has been claimed to be a very useful tool in assisting task sequencing (Rogers and Bloebaum, 1994; McCulley and Bloebaum, 1994, 1996, 1998). Sobieszczanski-Sobieski (1989, 1993) combined DSM with a mathematical model of the physical problem to form a hierarchical, constrained nonlinear optimization problem. Eppinger *et al*. (1990, 1994) analyzed task interactions to create design task groupings using DSM in order to find alternative sequence and structure of a project. Kusiak and Wang (1993a, 1993b) decomposed tasks in design using DSM and group technology to identify how tasks should be divided into groups. Guo *et al*. (1995) developed an algorithm for DSM to carry out the planning tasks. Smith and Eppinger (1997a, 1997b) used DSM to represent each task with deterministic duration and probabilistic repetition in order to identify controlling features of engineering design iterations. Chen and Lin (2002, 2003) used DSM for interdependent task group decomposition and project task coordination model for team organization in concurrent engineering.

Genetic Algorithms (GA)

Invented by Holland in 1975 and later developed by Goldberg (1989), genetic algorithm (GA) is a programming technique that mimics biological evolution as a problem-solving strategy. GA has proved to be very effective in optimizing very complex systems (Goldberg, 1989; Rogers, 1989, 1996, 1997; Eppinger *et al*., 1990, 1994; Syswerda, 1990; Gebala and Eppinger, 1991; Kusiak and Wang, 1993a, 1993b, 1993c; McCulley and Bloebaum, 1994, 1996, 1998; Rogers and Bloebaum, 1994; Thomas and Eppinger, 1994; Kusiak and Larson, 1995; Yassine *et al.,* 2000, 2001; Rohatynski and Kielec, 2001, 2001; Kielec, 2002; Rohatynski *at el*., 2002).

The Work of GA

The concept of GA is simple. First, an initial population of chromosomes is randomly generated and serves as a group of candidates or potential solutions. Next, GA selects/eliminates those candidates based on a user-defined selection method with a designed fitness function for evaluating the chromosomes. Crossing and mutation processes are used to create multiple variations of the selected chromosomes, which are later subjected to the next fitness evaluation and selection. Only the best chromosomes survive. The expectation is that the fitness average of the population will improve each round, and therefore, by repeating this process for hundreds or thousands of generations, an optimal solution or some near optimal solutions to the problem will be obtained. Figure 4 shows the classical GA process (Goldberg, 1989).

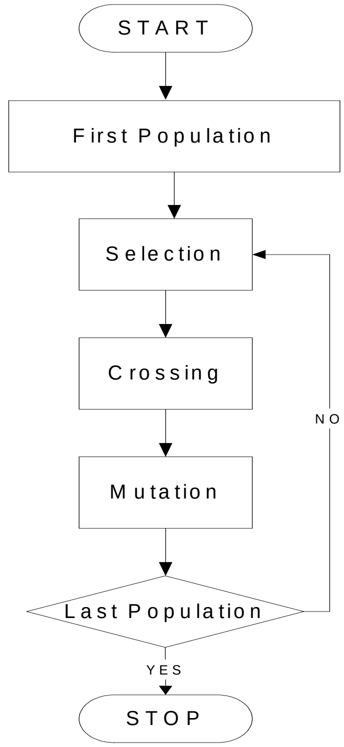

Figure 4. Classical Genetic Algorithm

GA Code Representation

Before a genetic algorithm can be put to work on any problem, a method is needed to encode potential solutions to that problem in a form that a computer can process. One common approach is to encode solutions as binary strings: sequences of 0's and 1's, where the digit at each position represents the value of some aspect of the solution. Another, similar approach is to encode solutions as arrays of integers or decimal numbers, with each position again representing some particular aspect of the solution. This approach allows for greater precision and complexity than the comparatively restricted method of using binary numbers only and often "is intuitively closer to the problem space" (Fleming and Purshouse, 2002). Virtually every combination of process solution can be represented by genetic code as a binary string called chromosome. Figure 5 a) represents a dependency matrix containing five tasks; Figure 5 b) shows the representation of the same process in the genetic and binary form.

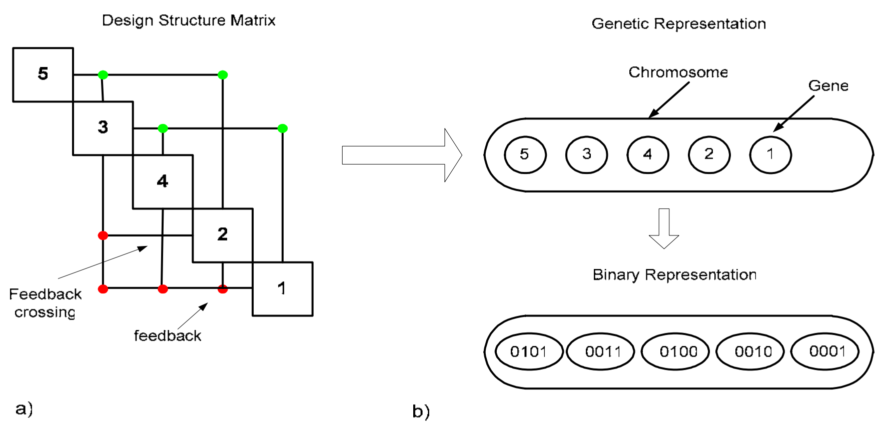

Figure 5. Genetic and Binary Representation of Dependency Matrix.

<u>Selection in GA</u>

In nature, a life form that is more suitable to its environment has a better chance of survival. If it does, its genetic information is passed down to the next generation, otherwise, there is a high probability that it will die before having a chance to reproduce. This is the most important way the nature ensures that subsequent generations have the best possible genetic heritage and they are well suited to the environment. In the similar fashion, the selection methods in GA with a user-defined fitness function control the probability that a given chromosome in a population will reproduce, favoring those chromosomes with higher fitness. The following is a list of selection methods commonly used in GA:

1) Roulette Selection – In this method, each chromosome has a place on the wheel with a slice proportional to its fitness. The wheel is spun enough times to fill up the required population size. The larger the slice area (i.e., the higher fitness value), the better chance the chromosome will be selected. This method has a potential for obtaining the optimal solution very fast. However, if only those chromosomes with high fitness values are selected in the beginning, this method often leads to a premature convergence and results in a poor solution.
2) Tournament Selection – This is the most often-used selection method in GA. First, chromosomes in the population are randomly divided into subgroups with n chromosomes each. The chromosome with the highest fitness in each subgroup is then selected and copied to the next new population. The advantage of this method is that it prevents the algorithm from a premature convergence due to random generation of subgroups where the chromosomes with lower fitness still have a chance of survival. In addition, it should be noted that the size and the number of subgroups have to be carefully selected by the user when using the method.
3) Ranking Selection – At first, this method orders the chromosomes in accordance with their fitness values. Each chromosome is then assigned a number to indicate its ranking in the population. The number of copies of each chromosome selected depends on its ranking. Therefore, much care should be given when developing the fitness function for the method. Similar to tournament selection, this method is also able to prevent the algorithm from a premature convergence.

4) Generational Selection – In this method, the chromosomes selected from each previous generation are not included the next generation, so that no chromosomes are retained between generations. The outcome of using this method is very case-dependent. If the method is applied to an unsuitable case, for example, it can lead to a long computational time before obtaining the optimal solution. This is due to not retaining the chromosomes between generations.
5) Steady-State Selection – In contrast with generational selection, this method retains some chromosomes between generations. This is done by replacing some of the less-fit chromosomes in the new generation with the better ones from the previous generation. Even though this method can quickly lead to an optimal solution, it can also lead to a local optimum due to premature convergence.
6) Hierarchical Selection – In this method, chromosomes go through multiple rounds of selections in each generation. Generally, there are two levels in the selection: lower-level selection (faster but less discriminating) and higher-level selection (slower but more discriminating). This method is able to reduce the overall computational time by using the lower-level selection to quickly remove most chromosomes that show little or no promise, and only subjecting those chromosomes survived from the initial selection to more discriminating and more computationally expensive fitness evaluation in the higher-level selection. However, hierarchical selection is difficult to implement because the method involves n multiple rounds that have to work efficiently in a sequential manner in order to reach the optimal solution. In addition,

the interactions between these multiple rounds will have a significant effect on the selection, so more care should be given when designing the multiple rounds.

7) Scaling Selection – In this method, if the average fitness of the population increases, the strength of the selective pressure also increases and the fitness function becomes more discriminating. This method can be helpful to the later stage of selections when all chromosomes have relatively high fitness with very small differences in fitness among them.

Crossing and Mutation in GA

Crossing (or crossover) in GA is equivalent to genetic recombination in nature. When two parent organisms produce an offspring, the offspring is not identical to either parent, but represents a genetic mix of both. The potential benefit of the mix is a chance that the fitter chromosome will be created, however, the chance of creating a weaker chromosome is also possible. Among all selected chromosomes to reproduce, pairs are randomly selected to execute the crossing. There are two methods for crossing: single-point and multi-point crossing (Syswerda, 1990). In single-point crossing, a point of exchange is set at a random location in the two individual chromosomes, and one chromosome contributes all its code from before the point of exchange where the other contributes all its code from after the point of exchange to produce an offspring. In multi-point crossing, the value at any given location in the offspring's chromosome is either the value of one parent's chromosome at that location or the value of the other parent's chromosome at that location, chosen with 50/50 probability.

Occasionally in nature, in the process of producing offspring, a gene mutates. In most cases, mutation results in organism unsuited to live in the environment. However, sometimes mutation creates useful genetic information that enhances its fitness. In GA, the mutation operation randomly changes the gene's position within the chromosome. Mutations are processed usually with small probability (about 0.1% chance). They contribute to diversity within the population and prevent from homogenous population that can lead to premature convergence in the evolution process.

Strengths of GA

There are many strength of using GA for complex systems optimization. Some of the most important are presented below:

1) Multiple Direction Search: Most other algorithms can only explore the solution space to a problem in one direction at a time, and if the solution they discover turns out to be a local optimum, there is nothing to do but abandon all work previously completed and start over. GA on the other hand can explore the solution space in multiple directions at once. If one direction turns out to be a dead end, it can easily be eliminated and GA can continue work on more promising paths, giving a greater of finding the optimal solution.
2) Efficient Performance in Complex Problems: Genetic algorithms are particularly well-suited for solving any complex problems where the space of all potential solutions is truly huge - too vast to search exhaustively in any reasonable amount of time. For example, a genetic algorithm developed jointly by engineers from General

Electric and Rensselaer Polytechnic Institute produced a high-performance jet engine turbine design that was three times better than a human-designed configuration and 50% better than a configuration designed by an expert system by successfully navigating a solution space containing more than 10^{387} possibilities. Conventional methods for designing such turbines are a central part of engineering projects that can take up to five years and cost over $2 billion; the genetic algorithm discovered this solution after two days on a typical engineering desktop workstation (Holland, 1992).

3) Effective at Escaping the Local Optima: Many search algorithms can become trapped by local optima. The problem of finding the global optimum in a space with many local optima is also known as the dilemma of exploration vs. exploitation, "a classic problem for all systems that can adapt and learn" (Holland, 1992). For example, if they reach the top of a hill on the fitness landscape, they will discover that no better solutions exist nearby and conclude that they have reached the best one, even though higher peaks exist elsewhere on the map. GA, on the other hand, has proven to be effective at escaping local optima and discovering the global optimum in even a very rugged and complex fitness landscape. It should be noted that in reality, there is usually no way to tell whether a given solution to a problem is the global optimum or just a very high local optimum. However, even if a GA does not always deliver a perfect solution to a problem, it can almost always deliver at least a very good solution.

4) Ability to Manipulate Many Parameters Simultaneously: GAs are very good at solving problems expressed in terms of multiple objectives: in particular, their use of

parallelism enables them to produce multiple equally good solutions to the same problem, possibly with one candidate solution optimizing one parameter and another candidate optimizing a different one (Haupt and Haupt, 1998), and a human overseer can then select one of these candidates to use. If a particular solution to a multiobjective problem optimizes one parameter to a degree such that this parameter cannot be further improved without causing a corresponding decrease in the quality of some other parameters, this solution is called *Pareto optimal* or *non-dominated* (Coello, 2000).

5) No Prior Knowledge Requirement: GA knows anything about the problems they are deployed to solve. Instead of using previously known domain-specific information to guide each step and making changes with a specific eye towards improvement, as human designers do, they are "blind watchmakers" (Dawkins, 1996); they make random changes to their candidate solutions and then use the fitness function to determine whether those changes produce an improvement. The virtue of this technique (Marczyk, 2004) is that it allows genetic algorithms to start out with an open mind, so to speak. Since its decisions are based on randomness, all possible search pathways are theoretically open to a GA; by contrast, any problem-solving strategy that relies on prior knowledge must inevitably begin by ruling out many pathways a priori, therefore missing any novel solutions that may exist there (Koza *et al*., 1999). Lacking preconceptions based on established beliefs of “how things should be done” or “what couldn't possibly work”, GAs do not have this problem. Similarly, any technique that relies on prior knowledge will break down when such knowledge

is not available, but again, GAs are not adversely affected by ignorance (Goldberg, 1989). Through their components of parallelism (searching in many directions at the same time), crossover, and mutation, they can range widely over the fitness landscape, exploring regions which intelligently produced algorithms might have overlooked, and potentially uncovering solutions of startling and unexpected creativity that might never have occurred to human designers. One vivid illustration of this is the rediscovery, by genetic programming, of the concept of negative feedback - a principle crucial to many important electronic components today, but one that, when it was first discovered, was denied a patent for nine years because the concept was so contrary to established beliefs (Koza *et al*., 2003). Genetic algorithms, of course, are neither aware nor concerned whether a solution runs counter to established beliefs - only whether it works.

Problems and Limitations of GA

1) Code Representation: The code used to specify a solution (chromosome) has to be able to tolerate random changes such that fatal errors or nonsense do not consistently result. This issue of representing chromosome in a robust way does not arise in nature but it does in GAs. Therefore, some changes (usually mutations) to an individual's genes could produce an unintelligible or even impossible result. There are two main ways of avoiding this (Marczyk, 2004). The first, which is used by most genetic algorithms, is to define individuals as lists of numbers – binary-valued, integer-valued, or real-valued – where each number represents some aspect of a candidate

solution. In second method, genetic programming, the actual program code changes. GA represents individuals as executable trees of code that can be mutated by changing or swapping subtrees. Both of these methods produce representations that are robust against mutation and can represent many different kinds of problems.

2) Fitness Function Design: The problem of how to write the fitness function must be carefully considered so that higher fitness is attainable and actually does equate to a better solution for the given problem. If the fitness function is chosen poorly or defined imprecisely, the genetic algorithm may be unable to find a solution to the problem, or may end up solving the wrong problem (Marczyk, 2004). In addition to making a good choice of fitness function, the other parameters of a GA (size of the population, the rate of mutation and crossover, the type and strength of selection) must be also chosen with care. If the population size is too small, the genetic algorithm may not explore enough of the solution space to consistently find good solutions. If the rate of genetic change is too high or the selection scheme is chosen poorly, convergence might never be obtained.

3) Size of the Population: if a population size is too low, mutation rates tend to be too high, selection pressure is too strong then solution might never be found (Marczyk, 2004). It is crucial to establish the size of the population that will lead to an optimal solution to the particular problem. In general, the more complex the problem is the larger the population size should be chosen. It is usually recommended to test the newly design GA with its fitness function to make conclusions about the optimal population size that should be used.

4) Premature Convergence: If an individual that is more fit than most of its competitors emerges early on in the course of the run, it may reproduce so abundantly that it drives down the population's diversity too soon, leading the algorithm to converge on the local optimum that this individual represents rather than searching the fitness landscape thoroughly enough to find the global optimum (Forrest, 1993; Mitchell, 1996). This is an especially common problem in small populations, where specific selection method can cause one particular chromosome to become dominant over others. This problem can be avoided by controlling the strength of selection, so as not to give excessively fit individuals too much of an advantage. Rank, scaling and tournament selection, discussed earlier, are three major means for accomplishing this (Marczyk, 2004).
5) Computational Effort: Several researchers (Holland, 1992; Forrest 1993; Haupt and Haupt, 1998) advise against using genetic algorithms on analytically solvable problems. It is not that genetic algorithms cannot find good solutions to such problems; it is merely that traditional analytic methods take much less time and computational effort than GAs and, unlike GAs, are usually mathematically guaranteed to deliver the one exact solution.

Genetic Algorithm vs. Other Problem-Solving Methods

With the rise of artificial life computing and the development of heuristic methods, other computerized problem-solving techniques have emerged that are in some ways similar to genetic algorithms. This section explains some of these techniques, in what ways they resemble GAs and in what ways they differ.

Calculus-based Search Methods

Calculus-based methods subdivide into two main classes: indirect and direct. Indirect methods seek local extrema by solving the usually non-linear set of equations resulting from setting the gradient of the objective function equal to zero, whereas direct methods seek local optima by hopping on the function and moving in a direction related to the local gradient. The main disadvantage of Calculus-based Search is the tendency for the search to get trapped at local maxima even though a better solution may exist. Secondly, the application of such searches depends on the existence of derivatives, for example the gradient of the solution space.

Enumerative Methods

Enumerative methods are search algorithms that start looking within a finite search space, or a discrete infinite search space, at objective function values at every point in the space, one at a time. Although the algorithm looks attractive due to its simplicity, many practical spaces are simply too large to search one at a time and still have a chance of using the information to some practical end.

Random Methods

Random search algorithms walk along randomly chosen points within the search space and simply choose the point with the best value of the objective function. They have achieved increasing popularity as they are easier to program than calculus-based methods. However, in the long run, random searches can be expected to do no better than enumerative schemes. Monte Carlo methods (random search for a limited period) are well-known random search algorithms.

Hill-Climbing Method

Similarly to genetic algorithms, hill-climbing technique is a combination of random and gradient search. It is more systematic and less random than genetic algorithm. Hill-climbing algorithm begins with one initial solution to the problem at hand, usually chosen at random. The string is then mutated, and if the mutation results in higher fitness for the new solution than for the previous one, the new solution is kept; otherwise, the current solution is retained. The algorithm is then repeated until no mutation can be found that causes an increase in the current solution's fitness and this solution is returned as the result (Koza *et al*., 2003). Once one peak has been located, the hill-climb is started again, but with another randomly chosen starting point. However, since each random trial is performed in isolation, no overall idea of the shape of the domain is obtained; also trials are randomly allocated over the entire search space and as many points in regions of low fitness will be evaluated as points in high fitness regions.

A genetic algorithm, however, starts with an initial random population, and allocated more trials to regions of the search space found to have high fitness.

Simulated Annealing

This technique was invented by Kirkpatrick in 1982, it is essentially a modified version of hill-climbing. The idea borrows its name from the industrial process of annealing in which a material is heated to above its melting point, then gradually cooled in order to erase defects in its crystalline structure, producing a more stable and regular lattice arrangement of atoms (Haupt and Haupt, 1998). In simulated annealing, as in genetic algorithms, there is a fitness function that defines a fitness landscape. However, simulated annealing deals with only one candidate at a time, rather than a population of candidates as in GAs, so, like random search, it does not build an overall picture of the solution space, and no information from previous moves is used to guide the selection of new moves. Simulated annealing also adds the concept of "temperature", a global numerical quantity which gradually decreases over time (Marczyk, 2004). At each step of the algorithm, the solution mutates (which is equivalent to moving to an adjacent point of the fitness landscape). The fitness of the new solution is then compared to the fitness of the previous solution; if it is higher, the new solution is kept. Otherwise, the algorithm makes a decision whether to keep or discard it based on temperature. If the temperature is high, as it is initially, even changes that cause significant decreases in fitness may be kept and used as the basis for the next round of the algorithm, but as temperature decreases, the algorithm becomes more and more inclined to only accept fitness-increasing changes.

Finally, the temperature reaches zero and the system "freezes", whatever configuration it is in at that point becomes the solution. Simulated annealing is often used for engineering design applications such as determining the physical layout of components on a computer chip (Kirkpatrick *at el.,* 1983).

Neural Network Approach

Neural Network is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It consists of layers of processing units called nodes joined by directional links: one input layer, one output layer, and zero or more hidden layers in between. An initial pattern of input is presented to the input layer of the neural network, and nodes that are stimulated then transmit a signal to the nodes of the next layer to which they are connected. If the sum of all the inputs entering one of these virtual neurons is higher than that neuron's so-called activation threshold, that neuron itself activates, and passes on its own signal to neurons in the next layer. The pattern of activation therefore spreads forward until it reaches the output layer and is there returned as a solution to the presented input (Marczyk, 2004). Neural Networks, like people, learn by example. They cannot be programmed to perform a specific task, so, the examples from which neural network needs to learn must be selected carefully otherwise useful time is wasted or even worse the network might function incorrectly. Different genetic algorithms have been used to build and to train neural networks.

Tabu Search

Tabu search (Golver and Laguna 1997) is an iterative heuristic technique that is capable of solving combinatorial optimization problems. The main difference between Tabu search and GA (and other methods) is that it forbids or penalizes the moves which take the solution, in the next iteration, to points in the solution space previously visited. In general, the resulting tendency of Tabu search to deviate from a charted course might be regretted as a source of error but can also prove to be source of gain. The Tabu method operates in this way with on exception: new courses are not chosen randomly but are preceded accordingly to the supposition that there is no point in accepting a new solution unless it is to avoid a path already investigated. This insures new regions of a problems solution space will be investigated with the goal of avoiding local minima and ultimately finding the desired solution. The Tabu search begins by marching to a local minimum. To avoid retracing steps, the method records recent moves in one or more Tabu lists. The original intent of the list was not to prevent a previous move from being repeated, but rather to insure it was not reversed. The Tabu lists are historical in nature and form the Tabu search memory. The role of the memory can change as the algorithm proceeds. At initialization the goal is to make a coarse examination of the solution space, known as diversification, but as candidate locations are identified the search is more focused to produce local optimal solutions in a process of intensification. In many cases the differences between the various implementations of the Tabu method have to do with the size, variability, and adaptability of the Tabu memory to a particular problem domain.

Summary of Literature Review

Sound project management must assign the right team members to the right project tasks at the right time. After a careful literature review, the existing and critical research gap in project management within the concurrent engineering environment is identified. Clearly, an integrated methodology for efficient task-member assignment model in complex engineering projects is needed.

To effectively manage large-scale development projects in concurrent engineering, the complex task structure of the project must be well understood first. To optimize such complex task structure the Design Structure Matrix (DSM) and Genetic Algorithm (GA) were identified as a suitable technique to solve this problem. Second, to assign members to tasks a methodology for rating member's characteristics in a quantitative way is required. The Chen and Lin (2004) team member model using Analytic Hierarchy Process (AHP) have been found as an effective technique to deal with this problem.

In recognizing of the important industry need in concurrent engineering project management and a research gap, an integrated methodological framework for assigning the right team members to the right tasks at the right time is proposed.

CHAPTER 3

METHODOLOGY

The goal of this chapter is to develop a methodology for assigning the right team member to the right task at the right time in the concurrent engineering environment. The entire research framework consists of three models: Project Task Model, Team Member Model and Task-Member Assignment Model.

Project Task Model optimizes the task flow structure using DSM and GA. Team Member Model quantifies member's characteristics (i.e., multifunctional knowledge, teamwork capability, working relationship) using AHP and absolute measurements as well as to provide member's workload schedule. Task-Member Assignment Model accomplishes the ultimate goal of this research: assign the right team member to the right task at the right time.

Project Task Model

To understand the complex task structure of the project, the analysis of the sequence and dependency of project tasks is necessary. The genetic algoritm developed in this model for optimizing project task structure is shown in Figure 6. Steps 1 to 7 in the following highlight some important details of the GA optimization process:

1) Initial Population: This is to randomly generate the first population of chromosomes (solutions).

2) Fitness Function Evaluation: The fitness function (Rohatynski and Kielec, 2001, 2001; Kielec, 2002; Rohatynski *at el.*, 2002) shown below evaluates the fitness (or quality) of every chromosome in the population by calculating the fitness value, W_t . The chromosomes with higher fitness values are the better chromosomes.

$$W_t = (w_f \cdot F + w_x \cdot X + w_t \cdot T + w_c \cdot C)^{-4}$$

where:

F = number of feedback connections

X = number of crossings of feedback connections

T = task time

C = task cost

w_f, w_x, w_t, w_c = weights (from 0 to 1)

3) Selection of Chromosomes: The algorithm uses the tournament selection method. The major advantage of tournament selection method is ability to protect the algorithm from premature convergence. This is done by random chromosome selection for tournaments in which chromosomes with low fitness function have greater probability for survival. In every reproduction step, two chromosomes are selected from the new generated population ($ch_i, ch_j \subset P_t$) for the tournament selection. The winner of the tournament is the chromosome with a higher fitness value, W_j, which is then copied to the next population. The tournament selection continues until the population meets a user-defined size. The following example shows two chromosomes (ch_1 and ch_5) are selected among six chromosomes in a population for the tournament selection where

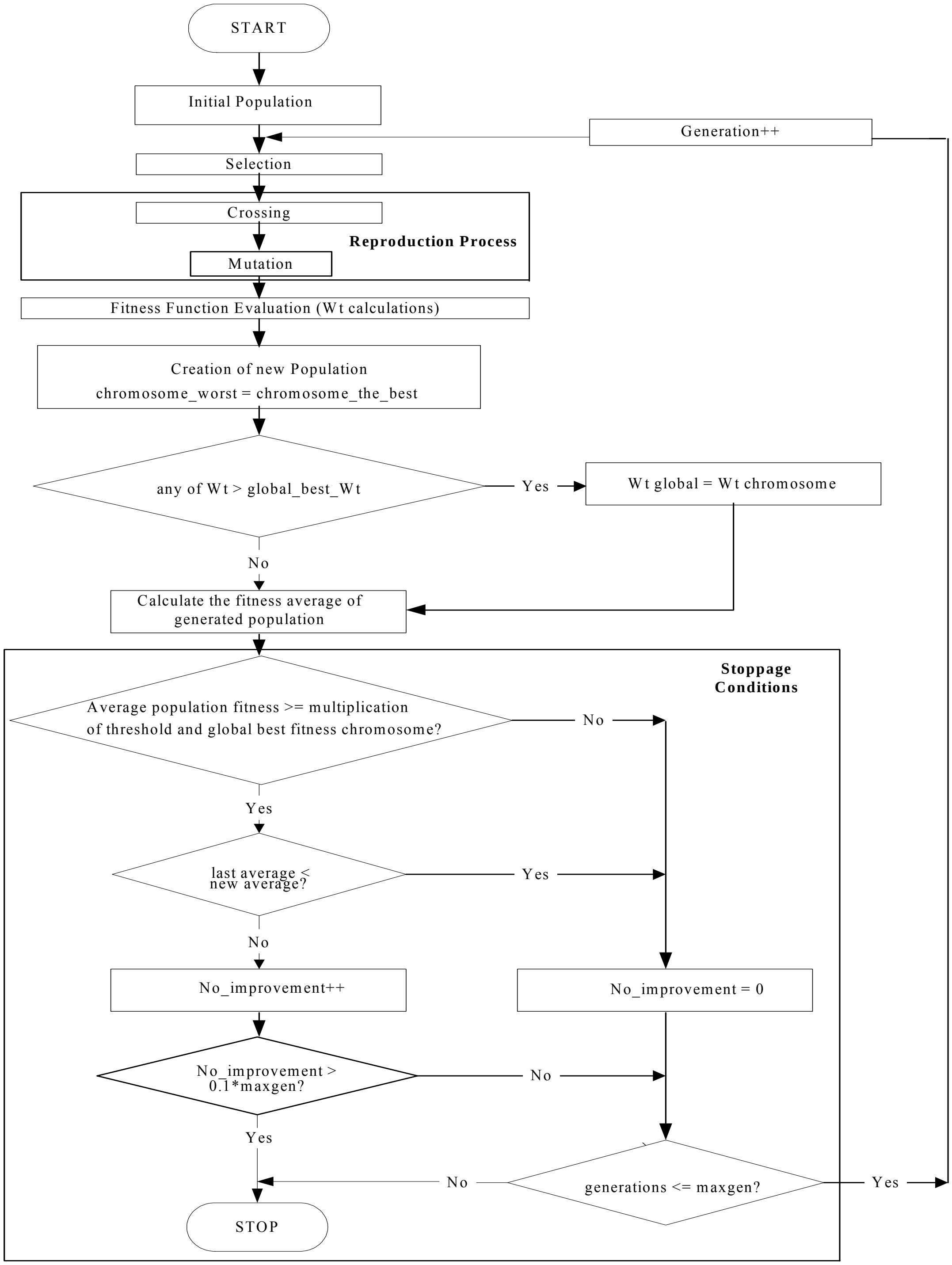

Figure 6. Genetic Algorithm for Project Task Structure Optimization

ch_5 with a higher fitness value wins the tournament and will advance to the next population.

$ch_1 = [1\ 2\ 3\ 4\ 5]$ $ch_2 = [3\ 1\ 2\ 4\ 5]$ $ch_3 = [5\ 3\ 4\ 2\ 1]$
$ch_4 = [2\ 1\ 3\ 4\ 5]$ $ch_5 = [4\ 1\ 3\ 2\ 5]$ $ch_6 = [4\ 5\ 3\ 2\ 1]$

$W_t\ (ch_1) = 0.4672$ $W_t\ (ch_5) = 0.8452$

4) Reproduction Process by Crossing and Mutation: Multi-points crossing is used in the algorithm. Multi-points crossing introduces more variability in new chromosome creation than single-point crossing, as is desired for optimization in very complex project task structures where numerous local optimum solutions may exist. For each randomly selected pair of chromosomes in the population, the positions of the genes to cross are selected according to the following conditions:

$$L_p > 0.5 \rightarrow Stay$$
$$L_p \leq 0.5 \rightarrow Cross$$

where: L_p = generated random number for crossing probability

For example, a new chromosome is generated as shown in Figure 7 where tasks (or "genes") 5 and 2 in chromosome 1 stay in the same position because they are the only two tasks whose L_p values are higher than 0.5, while tasks 4, 3, and 1 in chromosome 2 are crossing to the new positions in sequence.

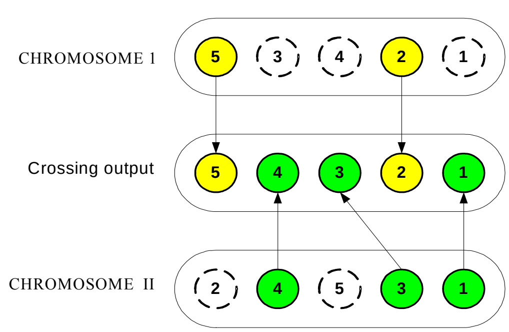

Figure 7. Crossing Process

The probability of mutations, that is a pair of genes within the chromosome switching their positions, usually is very low in the GA optimization process (i.e., about 0.1%). Figure 8 shows the mutation of tasks 4 and 2.

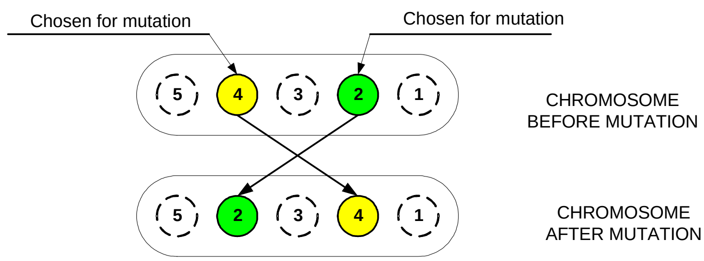

Figure 8. Mutation Process

5) Creation of the New Population: The chromosomes generated by the reproduction process will join the new population and each of them will be re-evaluated by the fitness function. In general, the best chromosome from the old population replaces the chromosome with the lowest fitness value in the new population. And then the rest of the chromosomes in the new population replace the chromosomes in the old population.

6) Check for the Stoppage Conditions: There are three stoppage conditions in the algorithm. First, the stoppage occurs when the fitness average of the current population is equal or greater than the global best chromosome times a user-defined threshold, Z_1 (1 by default), which is expressed mathematically as follows:

$$\sum_{i=1}^{n_{pop}} W_t / n_{pop} \geq global\ W_t \bullet Z_1$$

where: W_t = the fitness value,

Z_1 = the user-defined threshold

Second, the stoppage also occurs if there is no further improvement in the fitness function for a certain number of generations (e.g., 10% of the maximum number of generations pre-defined by the users). Third, the algorithm stops after the pre-defined maximum number of generations.

7) Selection of the Best Chromosome (or Solution): If one of the stoppage conditions is met, the GA optimization process is stopped and the optimal solution is the chromosome with the highest fitness value.

Team Member Model

(1) Multifunctional Knowledge Rating

A member who does not work in a certain functional department may still have a certain level of knowledge of this department. This will increase the flexibility when a key functional member is needed during the team organization. Using AHP and absolute measurements, the goal of this section is to model such multifunctional knowledge ratings for each member. Criteria, elements of comparison, pairwise comparison matrices, and the multifunctional knowledge rating vectors are explained in the following:

- Criteria: Each functional department D_k
- Elements of Comparison: Verbal intensities.

- Pairwise Comparison Matrices: N $m \times m$ pairwise comparison matrices are formed corresponding to n departments in the database. The general expression of the comparison matrix is formulated as follows:

D_k	E	VG	G	A	BA	P	VP
E	1	$\cdots$	k_{1i}	$\cdots$	k_{1j}	$\cdots$	k_{1m}
VP	$\vdots$	$\ddots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
G	k_{i1}	$\vdots$	1	$\cdots$	k_{ij}	$\cdots$	k_{im}
A	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\cdots$	$\cdots$	$\cdots$
BA	k_{j1}	$\vdots$	k_{ji}	$\vdots$	1	$\cdots$	k_{jm}
P	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\cdots$
VP	k_{m1}	$\vdots$	k_{mi}	$\vdots$	k_{mj}	$\vdots$	1

Where:

D_k = current selected functional department in the selected organization as criterion for the comparison

S_{ij} = eigen-vectors for verbal intensities (E, VG, G, A, BA, P, VP)

k_{ij} = numerical scale for S_i compare to $S_j \in \{1, 3, 5, 7, 9, 2, 4, 6, 8\}$

$k_{ji} = 1 / k_{ij}$ (reciprocal rule)

$k = 1, 2, \cdots, n$ (n departments) $\quad \forall\, i, j = 1, 2, \cdots\cdots, m$

- Eigen Vectors Calculations: N comparison matrices each with seven verbal intensities are obtained with respect to n functional departments.
- Multifunctional Knowledge Rating:

(i) For each member in the k = 1 assign the obtained eigen-vector *(E, VG, G, A, BA, P, VP)* using absolute measurements.

(ii) Increase k by 1.

(iii) Repeat steps (i) through step (ii) until $k = n$.

A multifunctional knowledge rating database is then obtained:

	D_1	$\cdots$	D_k	$\cdots$	D_n
M_1	K_{11}	$\cdots$	K_{1k}	$\cdots$	K_{1n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
M_i	K_{i1}	$\cdots$	K_{ik}	$\cdots$	K_{in}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
M_p	K_{p1}	$\cdots$	K_{pk}	$\cdots$	K_{pn}

Where: M_i = the i^{th} member in the company, $i = 1, 2, \cdots, p$

D_k = the k^{th} functional department in the company, $k = 1, 2, \cdots, n$

K_{ik} = multifunctional knowledge rating for the i^{th} member in the k^{th} department, $0 \le K_{ik} \le 1$ ($\forall\ i = 1, 2, \cdots\cdots, p$ and $k = 1, 2, \cdots\cdots, n$)

(2) Teamwork Capability Rating

In addition to functional knowledge, team members must also possess teamwork skills in order to work together successfully. Therefore, teamwork capability of each member is important for organizing a successful multifunctional team. From our review, three team member characteristics that relate to each member's teamwork capability are used in modeling the teamwork capability rating of team members:

1) Teamwork Experience: Since teams with more experienced members perform better than those with less experienced members, teamwork experience of each member should be taken into account during the team member selection.

2) Communication Skills: Team members with good communication skills will put forth agreement among team members and hence improve the team performance. Thus, selecting team members who have better communication skills is essential.

3) Flexibility in Job Assignment: Since the flexibility and involvement of teamwork also help ensure the quality of team performance, we should select team members who are motivated to engage in the task and have adequate time to devote to the job assignment in teamwork.

Thus, quantifying the ratings of each member's teamwork capability is a multi-criteria decision problem. However, defining the explicit measurement units for some of these team member characteristics is difficult, such as communication skills and flexibility in job assignment. From our review, as a multi-criteria decision method, Analytic Hierarchy Process (AHP) has been found effective in transferring qualitative decision measures of competing decision scenarios into quantitative values via pairwise comparison technique. The detail procedures for acquiring teamwork capability rating for each member in the company are explained in the following:

- Top Goal: Teamwork capability (for two criteria being compared, which criterion has a higher contribution to teamwork capability).
- Criteria: Three criteria are chosen, teamwork experience, communication skills, and flexibility in job assignment.

- Elements of Comparison: Criteria and verbal intensities.
- Pairwise Comparison Matrices: Two different sets of pairwise comparison matrices are considered. First, a 3×3 comparison matrix is formed for the three criteria to obtain their relative importance with respect to teamwork capability. The general expression of this comparison matrix is formulated as follows:

Top goal	C_1	C_2	C_3
C_1	1	t_{12}	t_{13}
C_2	t_{21}	1	t_{23}
C_3	t_{31}	t_{32}	1

Where:

Top goal = teamwork capability as a primary criterion to compare the three criteria:

C_1 = teamwork experience

C_2 = communication skills

C_3 = flexibility in job assignment

t_{ij} = comparison scale for C_i compare to $C_j \in \{1, 3, 5, 7, 9, 2, 4, 6, 8\}$ with respect to their teamwork capability

$t_{ji} = 1 / t_{ij}$ (reciprocal rule) $\quad \forall\ i, j = 1, 2, 3$

Second, three 7×7 comparison matrices, one for each C_i , are formed for verbal intensities to obtain the corresponding eigen-vectors. The general expression of these three comparison matrices is formulated as follows:

$$
\begin{array}{c|ccccccc}
C_k & E & VG & G & A & BA & P & VP \\
\hline
E & 1 & \cdots & c_{1i} & \cdots & c_{1j} & \cdots & c_{1p} \\
VG & \vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \cdots \\
G & c_{i1} & \vdots & 1 & \cdots & c_{ij} & \cdots & c_{ip} \\
A & \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \cdots \\
BA & c_{j1} & \vdots & c_{ji} & \vdots & 1 & \cdots & c_{jp} \\
P & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\
VP & c_{p1} & \vdots & c_{pi} & \vdots & c_{pj} & \vdots & 1
\end{array}
$$

Where: C_k = current selected criterion, $k = 1, 2, 3$

C_1 = teamwork experience

C_2 = communication skills

C_3 = flexibility in job assignment

S_{ij} = eigen-vectors for verbal intensities (*E, VG, G, A, BA, P, VP*)

c_{ij} = numerical scale for S_i compare to $S_j \in \{1, 3, 5, 7, 9, 2, 4, 6, 8\}$ with respect to each criterion k

$c_{ji} = 1 / c_{ij}$ (reciprocal rule) $\quad \forall\ i, j = 1, 2, \cdots\cdots, p$

- Teamwork Capability Rating Vector:

(i) First, a 1×3 eigen-vector, $W = \{w_1, w_2, w_3\}$, showing the relative importance of the three criteria is obtained in terms of pairwise comparisons. Next, three $p \times 1$ vectors are composed using absolute measurements from the previously generated eigen-vectors for all p members in the company with respect to each of the three criteria: teamwork experience, communication skills, and flexibility in job assignment. These three vectors are then combined into a $p\times 3$ matrix C. Finally, a $p\times 1$ teamwork capability vector T is obtained by multiplying matrix C with vector W.

$$T = C \times W = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ \vdots & \vdots & \vdots \\ c_{i1} & c_{i2} & c_{i3} \\ \vdots & \vdots & \vdots \\ c_{p1} & c_{p2} & c_{p3} \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} t_1 \\ \vdots \\ t_i \\ \vdots \\ t_p \end{bmatrix}$$

Where: c_{i1} = eigen-vector with respect to the criterion C_1

c_{i2} = eigen-vector with respect to the criterion C_2

c_{i3} = eigen-vector with respect to the criterion C_3

w_1 = the weight of criterion C_1

w_2 = the weight of criterion C_2

w_3 = the weight of criterion C_3

t_i = teamwork capability rating for member i in the company

$0 \leq t_i \leq 1 \qquad \forall\ i = 1, 2, \cdots\cdots, p$

(ii) Normalize the T eigen-vector to ensure all vector indices is between 0 to 1.000.

(3) Working Relationship Rating

In a multifunctional team, even with expert members who possess good teamwork capabilities, if they cannot work along well with one another, team performance will still be seriously degraded. Thus, it is important to understand working relationships between team members. Personality profiling using Myers-Briggs Type Indicator (MBTI), as a proven research, can be used as a reliable data source in characterizing the working relationship between team members. According to our review, MBTI has been a common battery for personality profiling that is the most widely used personality test in the US (Moore and Woods 1987, Guthrie 1993, Gardner and Martinko 1996). It is the instrument used to explain the effects that personal preferences have on decision making and problem solving that are essential to the teamwork. In MBTI, there are four groups of personality pairs: 1) Extrovert (E) - Introvert (I); 2) Sensing (S) - iNtuitive (N); 3) Thinking(T) - Feeling (F); and 4) Judging(J) - Perceiving (P). Every individual owns one preference from each of these four groups in any combination, which then forms the basis for the 16 personality types. For example, a person may be identified as INTP, ENTJ, ESFP, INFJ, or one of the 12 other possible combinations. According to our literature survey, we make the following findings for different personality types regarding teamwork (Myers 1980a, 1980b, Myers and McCaulley 1985, Edgley 1992, Crossley 1994):

1) An Extrovert person feels more comfortable working with a group, while an Introvert person prefers working alone quietly.
2) Since Sensing and iNtuitive types can be useful to each other as well as Thinking and Feeling types, a combination of S-N and T-F members is desirable for problem solving and decision making in teamwork.

3) People differ on Judging and Perceiving preferences will be hard to understand and predict to one another while people with the same type of Judging will consider the same things important and people with the same type of Perceiving will tend to have similar interests that enhance the team performance. Figure 9 shows four 2×2 tables indicating the positive, neutral, and negative relations for different combinations of personality types. By assigning numerical scales of +9, +3, and -3 to positive, neutral, and negative relations accordingly, a 16×16 working relationship scaling matrix can be formed with respect to 16 personality types (see Figure 10). For example, the working relationship scale of an ESTJ member vs. an ENTJ member is (9 + 9 + 3 + 9) = 30. Likewise, the working relationship scale of an ISTJ member vs. an ISTP member is (-3 + 3 + 3 - 3) = 0. For any two types, the higher value in the matrix indicates a better working relationship between these two types, and vice versa. These values can be further normalized into a range between 0 to 1 by dividing 36 into each value in the matrix, since 36 is the maximum value in the matrix. When completing the MBTI test, each member will know which one of these 16 types his or her personality profile belongs to. Consequently, the working relationship scaling matrix shows the degrees of how easy or how hard the two members can work together.

Extrovert (E) ↔ Introvert (I)

	E	I
E	+	O
I	O	–

Sensing (S) ↔ iNtuitive (N)

	S	N
S	O	+
N	+	O

Thinking (T) ↔ Feeling (F)

	T	F
T	O	+
F	+	O

Judging (J) ↔ Perceiving (P)

	J	P
J	+	–
P	–	+

Figure 9. Relations between Different Combinations of Personality Types

	ESTJ	ESTP	ESFJ	ESFP	ENTJ	ENTP	ENFJ	ENFP	ISTJ	ISTP	ISFJ	ISFP	INTJ	INTP	INFJ	INFP
ESTJ	24															
ESTP	12	24														
ESFJ	30	18	24													
ESFP	18	30	12	24												
ENTJ	30	18	36	24	24											
ENTP	18	30	24	36	12	24										
ENFJ	36	24	30	18	30	18	24									
ENFP	24	36	18	30	18	30	12	24								
ISTJ	18	6	24	12	24	12	30	18	12							
ISTP	6	18	12	24	12	24	18	30	0	12						
ISFJ	24	12	18	6	30	18	24	12	18	6	12					
ISFP	12	24	6	18	18	30	12	24	6	18	0	12				
INTJ	24	12	30	18	18	6	24	12	18	6	24	12	12			
INTP	12	24	18	30	6	18	12	24	6	18	12	24	0	12		
INFJ	30	18	24	12	24	12	18	6	24	12	18	6	18	6	12	
INFP	18	30	12	24	12	24	6	18	12	24	6	18	6	18	0	12

Figure 10. Working Relationship Scaling Matrix for Teamwork

(4) Member's Workload Schedule - is to enable assigning multiple tasks to one member. It presents the utilization of member's time, expressed as a percentage, throughout the entire project length. For example, a 75% workload in the member's workload schedule indicates that a particular member is to be utilized in 75% of his/her capacity during that time, leaving remaining 25% open for other tasks to be potentially

assigned. In such case, only a task with a 25% workload factor (T_{WF}) can be assigned to such member during that particular time. Overall, member's workload schedule is to be established so that feasible solution can be found. Members that are already occupied up to their full time capacity should not be considered for the task assignments, so that time, cost, and unnecessary effort corresponding to such efforts can be saved.

Task - Member Assignment Model

To assign the right team member to the right task at the right time a genetic algorithm with a newly designed fitness function was formulated. Figure 11 presents the algorithm.

Steps 1-8 below highlight some important details of the GA optimization process for task-member assignments:

(1) Initial Population: Randomly generate the first population of chromosomes (solutions).

(2) Chromosomes Feasibility Check: Every chromosome (solution) is checked for feasibility according to the each member's schedule and workloads. For example, if member 1 has already a workload factor set to 100% during the entire project duration, any task assignment for him/ her will result in not feasible solution.

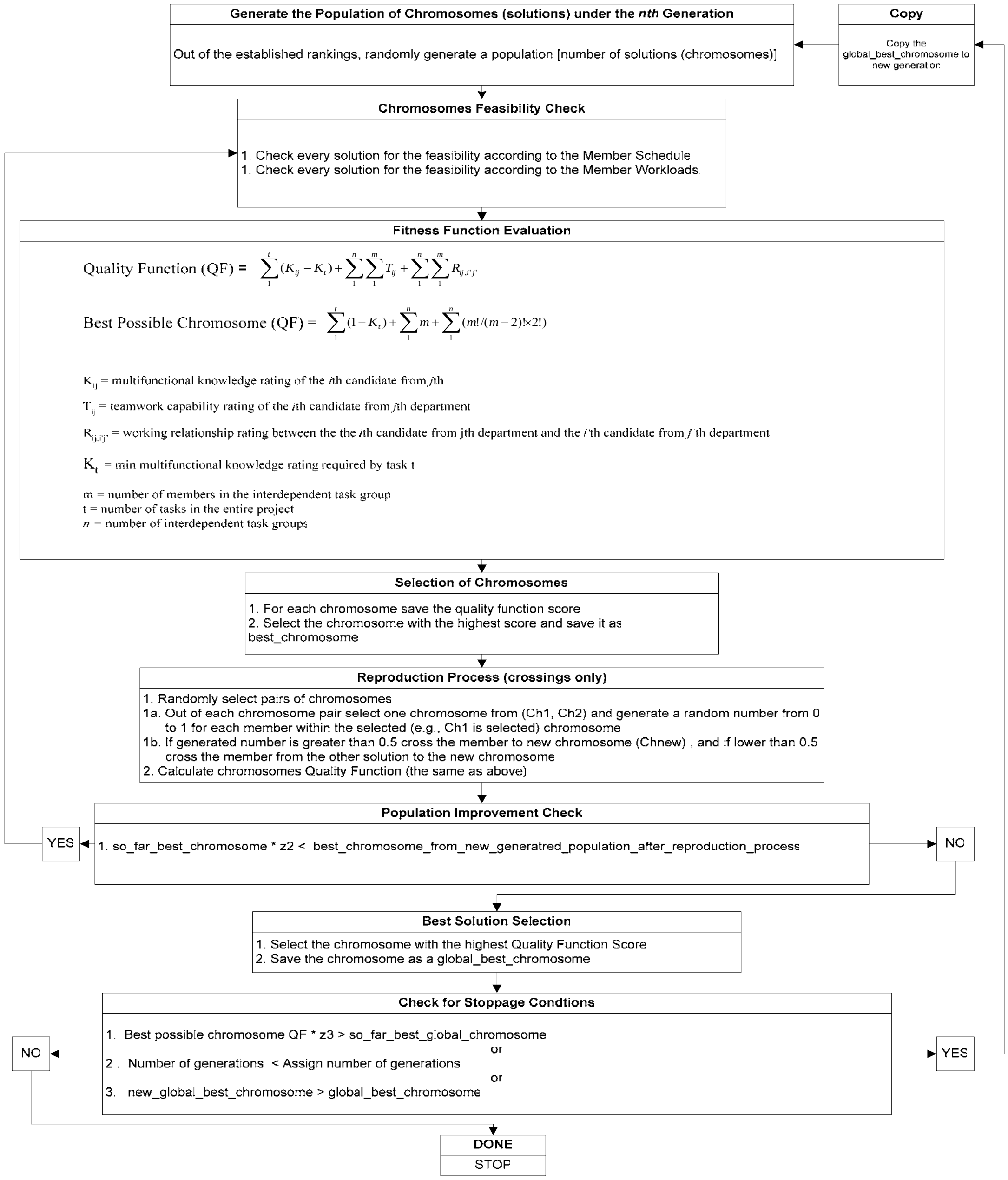

Figure 11. Genetic Algorithm for Task-Member Assignment Model

(3) Fitness Function Evaluation: The fitness function $W_{t\text{-}m}$ shown below is used to evaluate the fitness of every chromosome in the population. Function W_b represents

the best possible score for the chromosome. The chromosomes with closer values to W_b are the better chromosomes.

$$W_{t\text{-}m} = \sum_{1}^{t}(K_{ij} - K_t) + \sum_{1}^{n}\sum_{1}^{m} T_{ij} + \sum_{1}^{n}\sum_{1}^{m} R_{ij,i'j'}$$

$$W_b = \sum_{1}^{t}(1 - K_t) + \sum_{1}^{m} m + \sum_{1}^{n}(m!/((m-2)!\times 2!))$$

where:

K_{ij} = multifunctional knowledge rating of the *i*th candidate from *j*th department

T_{ij} = teamwork capability rating of the *i*th candidate from *j*th department

$R_{ij,i'j'}$ = working relationship rating between the the *i*th candidate from *j*th department and the *i'*th candidate from *j'*th department

K_t = multifunctional knowledge rating threshold required by task *t*

m = number of members in the interdependent task group

n = number of interdependent task groups

t = number of tasks in the entire project

(4) Selection of Chromosomes: Task-Member Assignment GA uses exactly the same tournament selection as project task model (see p. 45).

(5) Reproduction Process (crossing only): Task-Member Assignment GA uses the same crossing method as project task model (see p.45).

(6) Population Improvement Check: To check for the chromosome (solution) improvement within the population, the so far best chromosome is multiple by threshold Z_2 (user defined number from 0 to 1) to potentially lower (if Z_2 less than 1) the value of the so far best chromosome preventing the algorithm from founding the

local optima. For example, assigning 0.7 to Z_2 means that within any new iteration any chromosome scoring above the value equal 70% of the so far best chromosome will cause the algorithm to go to the next iteration within the same population to search for the better chromosome. If there is no chromosome generated with the score of above 70%, the algorithm will proceed to the next generation.

$$W_{t-m^{**}} * Z_2 > W_{t-m^*}$$

Where:

$W_{t\text{-}m^{**}}$ = so far best chromosome

$W_{t\text{-}m^{*}}$ = best chromosome from the new iteration after reproduction process

Z_2 = threshold (from 0 to 1)

(7) Best Solution Selection: In order to find the optimum solution the chromosome with the highest quality function score is selected and saved as a global best chromosome.

(i) Check of the Stoppage Condition: Similarly to the project task model there are three stoppage conditions in the algorithm. First, the stoppage occurs if the so far best global chromosome is less than best possible chromosome W_b times the threshold Z_3.

$$W_b * Z_3 > W_{t-m^{***}}$$

Where:

W_b = best possible chromosome

$W_{t\text{-}m^{***}}$ = global best chromosome

Z_3 = threshold (from 0 to 1)

Threshold Z_3 potentially lowers (if less than 1) the value of the best possible chromosome which is used in the algorithm as a minimum requirement for finding the solution. For example, assigning 0.8 to Z_3 means that any chromosome with fitness function score higher than 80% of the best possible chromosome is accepted as a solution. Second, the algorithm will stop if the number of generations already generated by the algorithm is equal the user defined number of generations. However, the stoppage will not occur if there is improvement in the fitness function from last generation.

CHAPTER 4

IllUSTRATIVE EXAMPLE

To demonstrate the effectiveness of the research model, the illustrative example is taken from a company that designs, manufactures and distributes different types of industrial and commercial furnaces. The ultimate goal of the example is to assign the right team member to the right task at the right time.

First, the entire project is decomposed into 27 tasks. Second, the relationships among them are established. It should be noted that the project decomposition and task relationship establishment provide critical information for the study that have to be handled by the experienced managers and engineers with care if satisfactory results are desired. Next, the time and cost of each individual task is estimated. Finally, departmental belonging is established for each task. Table 1 below presents the project task information including number, name, relationship, time, cost, and the responsible department of each task. For example, task 1 – Customer Order/Price Negotiations – has a relationship with task 5 – Project Initiation – meaning that task 1 provides information input to task 5. The time and cost for task 1 are estimated to be 150 and 40 units respectively. Lastly, task 1 is assigned to be executed in Sales Department (SD) (9).

Table 1. Project Task Information

Task Number	Task Description	Relationship	Time	Cost	Department(s) Responsible for Task Realization
1	Customer Order/Price Negotiations	5	150	40	Sales Dept. (SD) (9)
2	Order Specification Analysis for Production Feasibility	5	16	80	Design Testing Dept. (DTD) (3) Design Dept. (DD) (2) Services and Installation Dept. (SID) (10)
3	Technical Concept Proposition	5	60	40	Sales Dept. (SD) (9)
4	Price Evaluation	5	40	40	Sales Dept. (SD) (9)
5	Project Initiation	6, 7	30	40	Procurement Dept. (PrcD) (7) Technology Dept. (TD) (12) *Planning Coordinator (pc) (6) Design Testing Dept. (DTD) (3)
6	Project Design Documentation	8, 9, 10	1619	40	Design Dept. (DD) (2)
7	Project Components Documentation	11	16	40	Design Dept. (DD) (2)
8	Project Specification Check	9, 10, 26	308	40	Design Dept. (DD) (2)
9	User Manual Documentation	24	86	40	Design Dept. (DD) (2)
10	Past Experience Technology Consultation	11	45	28	Archival Office (AO) (1)
11	Technology Documentation	8, 12, 16, 17,18, 19, 20, 21, 22	324	40	Technology Dept. (TD) (12)
12	Evaluation/Selection of Procurement Partners	13, 15	42	35	Selling Dept. (SD) (9)
13	Material Procurement Ordering	14	100	35	Selling Dept. (SD) (9)
14	Material Procurement Receiving	15, 17, 18, 20, 21	100	28	Inventory Dept. (ID) (5)
15	Project Material Evaluation/ Specification Changes	8	5	35	Design Testing Dept. (DTD) (3)
16	Evaluation/Selection of Supply Chain Management Partners	17	452	40	Supply Chain Management Dept. (SCMD) (11)
17	Detail Analysis of Supply Chain Management Value Proposition	19, 22	1892	28	Supply Chain Management Dept. (SCMD) (11)
18	Detail Analysis of Firms Value Proposition	19, 22	270	28	Production Dept. (PD) (8)
19	Assembly of Furnace Subparts	20, 21, 22	1622	28	Production Dept. (PD) (8)
20	Assembly of Main Furnace Part	21, 22	2387	28	Production Dept. (PD) (8)
21	Final Assembly	8, 22	537	28	Production Dept. (PD) (8)
22	Final Testing	8, 23	250	32	Service and Installation Dept. (SID) (10)
23	Disassembly and Painting	24	347	28	Production Dept. (PD) (8)
24	Transportation	25	288	28	Distribution Dept. (DistD) (4)
25	Assembly over the Customer Site	26	353	84	Service and Installation Dept. (SID) (10)
26	Furnace Assembly Inspection	8, 27	259	32	Service and Installation Dept. (SID) (10)
27	Final Meeting	-	12	40	*Technical Director (td) (13)

Project Task Model

Project Task Flow before the GA Optimization Process

According to the task input/output relationships in Table 1, Figure 12 shows the entire project task flow using DSM before the GA optimization process (with 43 progressive and 8 feedback connections).

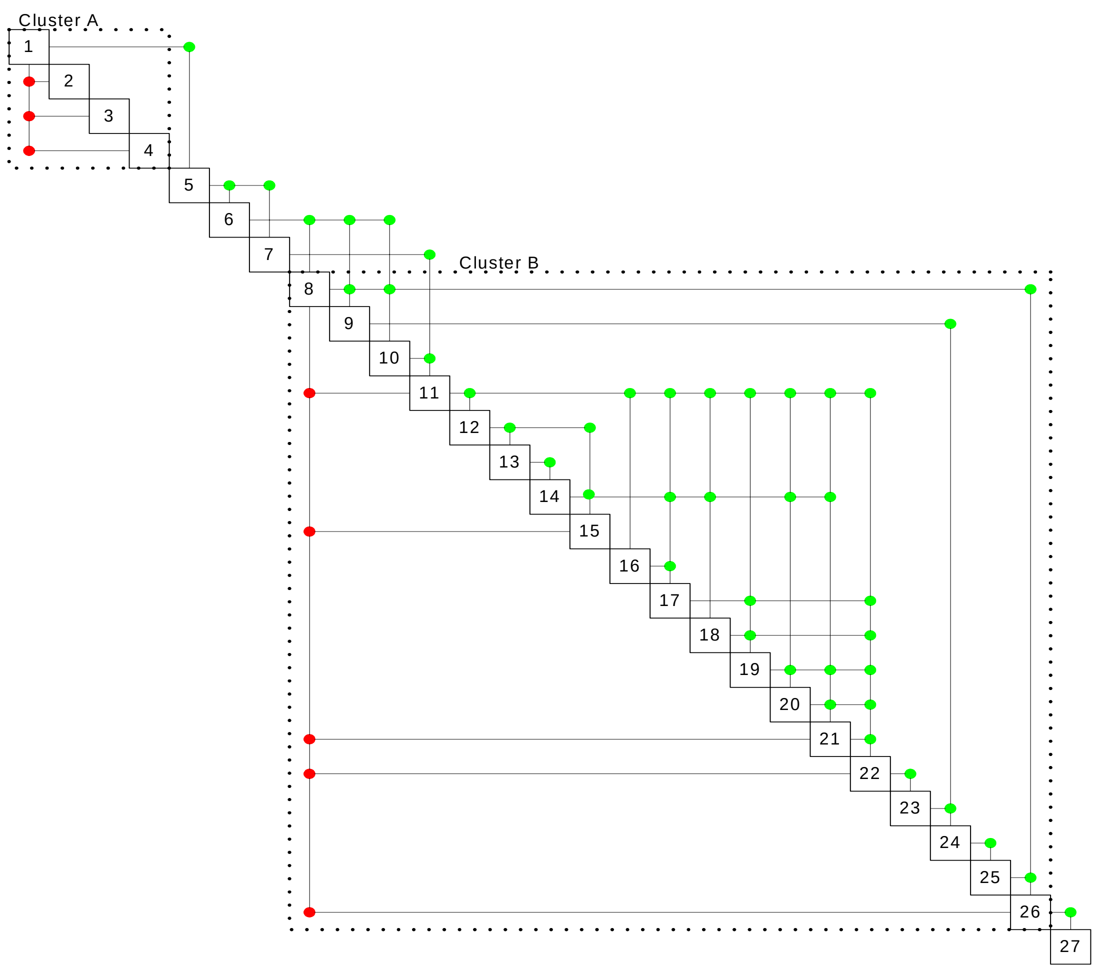

Figure 12. Project Task Flow before GA Optimization

The total time and cost of the entire project can be calculated by following the task flows revealed in DSM where the progressive and feedback flows are considered along with the concurrency among some tasks. For example, assuming that each feedback connection will cause only one rework (or iteration), the time units required to complete the tasks in cluster A and in the feedback loop between tasks 8 and 11 can be calculated as follows, respectively:

$$T_{clusterA} = T_{longest(1,2,3,4)} + T_1 = 150 + 150 = 300$$

$$T_{feedback(8-11)} = 2 \times \left(T_8 + T_{longest(9,10)} + T_{11}\right) = 2 \times \left(308 + 86 + 324\right) = 2 \times 718 = 1436$$

Calculating the cost is much simpler because there is no need to consider the concurrency among tasks. For example, the total cost units for the project are calculated in the following by adding up the costs of 27 tasks and the sums of five feedback loops:

$$C_{total} = \sum_{t=1}^{27} C_t + \sum_{t=8}^{11} C_t + \sum_{t=8}^{15} C_t + \sum_{t=8}^{21} C_t + \sum_{t=8}^{22} C_t + \sum_{t=8}^{26} C_t = 3273$$

It should be noted that more feedback connections entail more potential rework. Additionally, the more tasks in the feedback loop the more expensive and lengthy rework will be needed, which usually result in higher cost and longer time for a project to complete. Table 2 shows the rework time and cost needed for each feedback loop as well as the total time and cost for the entire project before the GA optimization process.

Table 2. Project Task Time and Cost before GA Optimization

Project Time and Cost before Optimization																		
Tasks	time	cost	time (1-27)	cost (1-27)	time (1-4)	cost (1-4)	time (8-11)	cost (8-11)	time (8-15)	cost (8-15)	time (8-21)	cost (8-21)	time (8-22)	cost (8-22)	time (8-26)	cost (8-26)	Total Time	Total cost
1	150	40																
2	16	80																
3	60	40																
4	40	40			150	200												
5	30	40																
6	1619	40																
7	16	40																
8	308	40																
9	86	40																
10	45	28																
11	324	40					718	148										
12	42	35																
13	100	35																
14	100	28																
15	5	35							965	281								
16	452	40																
17	1892	28																
18	270	28																
19	1622	28																
20	2387	28																
21	537	28									7608	461						
22	250	32											7558	493				
23	347	28																
24	288	28																
25	353	84																
26	259	32													9105	665		
27	12	40	11156	1025													37110	3273

<u>Project Task Flow after the GA Optimization Process</u>

The main goal in this section is to establish the best DSM task structure by the GA optimization process based on the information of task time and cost. Figure 13 shows the optimal project task flow after optimization, which becomes very different from the original structure. The optimized DSM contains only two feedback connections comparing to eight feedbacks connections before optimization.

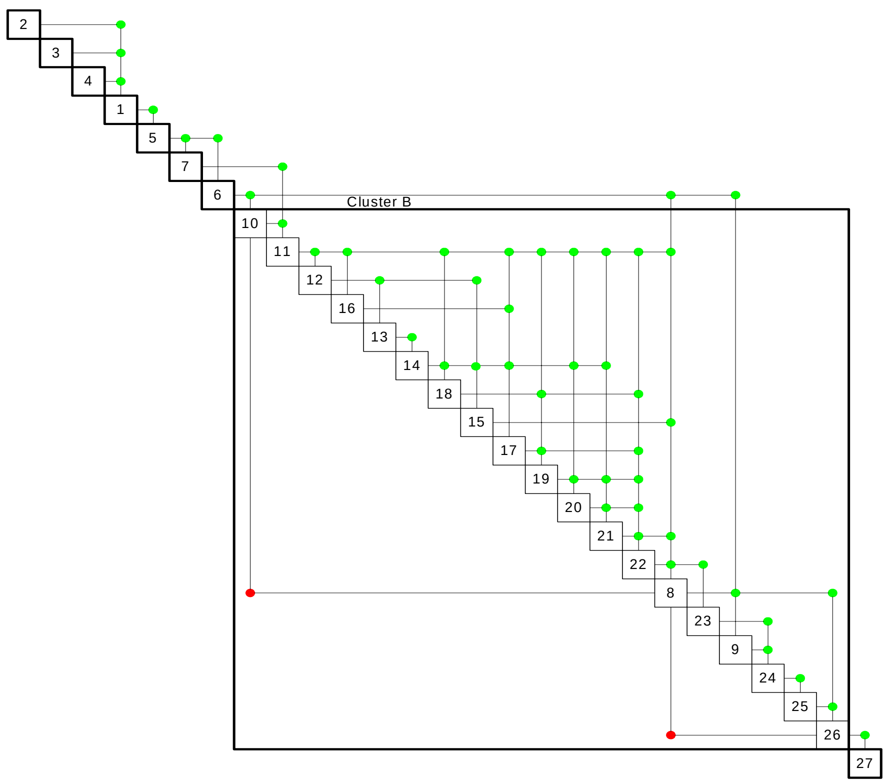

Figure 13. Project Task Flow after GA Optimization

Table 3 shows the rework time and cost needed for each feedback loop as well as the total time and cost for the entire project after the GA optimization process. It should be noted that the second feedback loop between tasks 8 and 26 in Figure 13 contains only six tasks comparing to 19 tasks in the original loop before optimization, thus the extra rework time and cost required for this new feedback loop are significantly reduced after optimization (time units decreased from 9105 to 1294 and cost units decreased from 665 to 252). Finally after the optimization process, we are able to reduce the total project time from 37110 to 19824 time units (46.58 % time reduction) and the cost from 3273 to 1730 cost units (47.14% cost saving).

Table 3. Project Task Time and Cost after GA Optimization

Project Time and Cost after Optimization										
Task Number	time	cost	time (1-27)	cost (1-27)	time (10-8)	cost (10-8)	time (8-26)	cost (8-26)	Total time	Total cost
2	16	80								
3	60	40								
4	40	40								
1	150	40								
5	30	40								
7	16	40								
6	1619	40								
10	45	28								
11	324	40								
12	42	35								
16	452	40								
13	100	35								
14	100	28								
18	270	28								
15	5	35								
17	1892	28								
19	1622	28								
20	2387	28								
21	537	28								
22	250	32								
8	308	40			7817	453				
23	347	28								
9	86	40								
24	288	28								
25	353	84								
26	259	32					1294	252		
27	12	40	10713	1025					19824	1730

Team Member Model

To assign the right team member to the right task at the right time, this model contains the following four member databases: Multifunctional Knowledge Rating, Teamwork Capability Rating, Working Relationship Rating, and Member's Workload Schedule. Multifunctional Knowledge Rating is a rating revealing each member's knowledge level to each functional department in the organization. Teamwork Capability Rating reveals each member's teamwork capability based on: Teamwork Experience, Communication Skills, and Flexibility in Job Assignment. Since the team members work closely, their collegiality (Working Relationship Rating) also needs to be measured. This is done by Myers–Briggs type indicator (MBTI), a popular personality test that has been widely used by both academia and industry. Member's Workload Schedule is to present the utilization of each member's time, expressed as a percentage, throughout the entire project cycle. Analytic Hierarchy Process (AHP) and absolute measurements are used to quantify each member's multifunctional knowledge and teamwork capability ratings in this model.

Multifunctional Knowledge Rating

A member who does not work in a certain functional department may still have a certain level of knowledge of this department. Using AHP and absolute measurement the goal of this section is to model such multifunctional knowledge rating for each member in the database. Appendix A presents the multifunctional knowledge database with ratings assigned for each member to a particular department. The weights for each eigen-vector are calculated using the same numerical values k_{ij} for all departments. The example of a pairwise comparison matrix for each department is shown below.

D_k	E	VG	G	A	BA	P	VP
E	1	2	3	5	6	7	9
VG	$\vdots$	1	2	3	5	6	7
G	k_{i1}	$\vdots$	1	2	4	5	6
A	$\vdots$	$\vdots$	$\vdots$	1	2	3	4
BA	k_{j1}	$\vdots$	k_{ji}	$\vdots$	1	2	3
P	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	1	2
VP	k_{p1}	$\vdots$	k_{pi}	$\vdots$	k_{pj}	$\vdots$	1

Where: D_k = current selected functional department in the selected organization as criterion for the comparison

$$k_{ji} = 1 / c_{ij} \text{ (reciprocal rule)} \quad \forall\, i, j = 1, 2, \cdots\cdots, p$$

The calculation from the pairwise comparison matrix using AHP results in the following eigen-vector's weights for the verbal intensities:

	S_{ij}
E	1
VP	0.654
G	0.448
A	0.257
BA	0.162
P	0.109
VP	0.074

Where: S_{ij} = eigen-vectors for verbal intensities.

These weights are assigned to members using absolute measurements. For example, (please refer to Appendix A) member 1 holds rating E to department 1 and VG to department 11. This results in the multifunctional knowledge rating 1.00 and 0.654 to department 1 and department 11 respectively.

Teamwork Capability Rating

In addition to functional knowledge, team members must also possess teamwork skills in order to work together successfully. Therefore, teamwork capability of each member needs to be calculated. Appendix B presents the teamwork capability rating (and normalized teamwork capability rating) for all the members. The ratings are calculated by multiplying the 1×3 eigen-vector $W = \{w_1, w_2, w_3\}$ showing the importance of the three criteria by p×3 matrix C representing teamwork capability ratings for all p members in the company with respect to each of the three criteria: teamwork experience, communication skills, and flexibility in job assignment.

To obtain vector W, a 3×3 comparison matrix with respect to 3 criteria (teamwork capability, communication skills, and flexibility in job assignment) is used.

Top goal	C_1	C_2	C_3
C_1	1	*1*	*5*
C_2	*1*	1	*3*
C_3	*0.2*	*0.33*	1

Where:

Top *goal* = teamwork capability as a primary criterion to compare the three criteria:

C_1 = teamwork experience

C_2 = communication skills

C_3 = flexibility in job assignment

To obtain vector C, three 7×7 comparison matrices are used with respect to each C_i .

C_1	E	VG	G	A	BA	P	VP
E	1	2	3	5	6	7	9
VG	⋮	1	2	3	5	6	7
G	c_{i1}	⋮	1	2	4	5	6
A	⋮	⋮	⋮	1	2	3	4
BA	c_{j1}	⋮	c_{ji}	⋮	1	2	3
P	⋮	⋮	⋮	⋮	⋮	1	2
VP	c_{p1}	⋮	c_{pi}	⋮	c_{pj}	⋮	1

C_2	E	VG	G	A	BA	P	VP
E	1	2	3	3	6	7	9
VG	⋮	1	2	4	5	6	8
G	c_{i1}	⋮	1	2	3	5	6
A	⋮	⋮	⋮	1	2	3	4
BA	c_{j1}	⋮	c_{ji}	⋮	1	2	3
P	⋮	⋮	⋮	⋮	⋮	1	2
VP	c_{p1}	⋮	c_{pi}	⋮	c_{pj}	⋮	1

C_3	E	VG	G	A	BA	P	VP
E	1	2	5	5	7	7	9
VG	⋮	1	3	3	7	7	7
G	c_{i1}	⋮	1	1	3	3	6
A	⋮	⋮	⋮	1	2	4	5
BA	c_{j1}	⋮	c_{ji}	⋮	1	1	3
P	⋮	⋮	⋮	⋮	⋮	1	2
VP	c_{p1}	⋮	c_{pi}	⋮	c_{pj}	⋮	1

Where:

C_1 = teamwork experience

C_2 = communication skills

C_3 = flexibility in job assignment

$c_{ji} = 1/c_{ij}$ (reciprocal rule) $\quad \forall\ i, j = 1, 2, \cdots\cdots, p$

Finally, the $p\times1$ rating vector T is obtained from matrix W and C multiplication and then normalized to obtained normalized teamwork capability ratings. For example, in our teamwork capability database presented in Appendix B, member 3, holds *E, E,* and *E* for each C_i, respectively. This results in 0.3665 for teamwork capability rating and 1.000 for normalized teamwork capability rating.

Working Relationship Rating

In a multifunctional team, even with expert members who possess good teamwork capabilities, if they cannot work along well with one another, team performance will still be seriously degraded. Thus, it is important to calculate working relationships between team members. Personality profiling using Myers-Briggs Type Indicator (MBTI), as a proven research, is used as a reliable data source in characterizing the working relationship between team members. Appendix C presents the working relationship ratings for all the members. Numerical scales of +9, +3, and -3 to positive, neutral, and negative relations accordingly are used. For example, the working relationship rating between members 10 and 19 is 0.500.

Task-Member Assignment Model

Task-Member Assignment Model combines both previous models (project task model and team member model), so that the ultimate research goal of this study can be accomplished – assign the right team members to the right tasks at the right time in the concurrent engineering environment. Appendix D presents the database including member's workload schedule. For example, (please refer to Appendix D) member 2 is not available in project periods 5, 8 and 10 because he/she indicates 100% workload of his/her capacity during those periods and only partially available, 50% in project period 18 and 25% in project period 19.

Task-Member Assignment Model Input

In our model to assign the right team member to the right task at the right time a user input regarding task workload factor (T_{WF}) and multifunctional knowledge rating threshold (K_t) are required. Table 4 presents the task-member assignment input for our illustrative example. Please note that only 1 organization is considered in this example (organization 1). Also, the departments in which tasks are to be executed are already determined by the user. For example, task 5 (Project Initiation) is to be executed in organization 1 under the following 4 departments: Procurement Department (7), Technology Department (12), *Planning Coordinator (6), and Design Testing Department (3). Moreover, task 5 has a workload factor set to 400% meaning that it will require 4 members committed each at 100% capacity. Because, there is only one planning coordinator the member assignment for department 6 is already known – member 19. Finally, the multifunctional knowledge rating threshold (K_t) for task 5 is set for 1 (E) allowing only expert member to be assigned to this task.

Table 4. Task-Member Assignment Model Input

Start	Name		Time	Cost	Information In	T_{WF}	Organization				Department				Member				K_t
1	T1	Business Offer Proposition	150	40	5	100%	1				9								0.654(VG)
	T2	Order Specification Analysis for Production Feasibility	16	80	1	300%	1				3	2	10						0.654(VG)
	T3	Technical Concept Proposition	60	40	1	100%	1				9								0.654(VG)
	T4	Price Evaluation	40	40	1	100%	1				9								0.654(VG)
	T5	Project Initiation	30	40	6,7	400%	1				7	12	6	3			19		1(E)
	T6	Project Design Documentation	1619	40	8,9,10	100%	1				2								1(E)
	T7	Project Components Documentation	16	40	11	100%	1				2								0.654(VG)
	T8	Project Specification Check/ Changes	308	40	9,10,26	200%	1				2	2							0.449(G)
	T9	User Manual Documentation	86	40	24	200%	1				2	2							0.449(G)
	T10	Past Experience Technology Consultation	45	28	11	200%	1				1	1							1(E)
	T11	Technology Documentation	324	40	8,12,16,17,18,19,20,21,22	200%	1				12	12							0.654(VG)
	T12	Evaluation/Selection of procurement partners	42	35	13,15	200%	1				9	9							0.654(VG)
	T13	Material Procurement Ordering	100	35	14	100%	1				9								0.654(VG)
	T14	Material Procurement Receiving	100	28	15,17,18,20,21	200%	1				5	5							0.654(VG)
	T15	Project Material Evaluation/ Specification Changes	5	35	8	100%	1				3								0.654(VG)
	T16	Evaluation/Selection of supply chain management partners	452	40	17	100%	1				11								1(E)
	T17	Detail Analysis of Supply Chain Management Value Proposition	1892	28	19,22	100%	1				11								0.654(VG)
	T18	Detail Analysis of Firms Value Proposition	270	28	19,22	100%	1				8								1(E)
	T19	Assembly of furnace subparts	1622	28	20,21,22	400%	1				8	8	8	8					1(E)
	T20	Assembly of main furnace part	2387	28	21,22	400%	1				8	8	8	8					1(E)
	T21	Final Assembly	537	28	8,22	200%	1				8	10							1(E)
	T22	Final Testing	250	32	8,23	200%	1				10	10							0.654(VG)
	T23	Disassembly and painting	347	28	24	300%	1				8	8	8						1(E)
	T24	Transportation	288	28	25	200%	1				4	4							1(E)
	T25	Assembly over the customer site	353	84	26	400%	1				10	10	10	10					0.654(VG)
	T26	Furnace Assembly Inspection	259	32	8, 27	100%	1				10								1(E)
	T27	Final Meeting	12	40	-	100%	1				13				42				1(E)

Task-Member Assignment Solution before and after Task Structure Optimization

According to the model input from Table 4, the task-member assignment for the before and after task flow structure optimization is accomplished. Tables 5 and 6 below present the before and after solutions respectively. Please note that the task-member assignments for both solutions do not differ significantly. However, in after task structure optimization solution task 2, 3, and 4 were assigned members 6, 7, 35 (task 2), 32 (task 3), and 31 (task 4), where in before task structure optimization members 6, 7, 31 were assigned to task 2, 32 to task 3, and 36 to task 4. Analyzing the solutions carefully it can be seen that not all the requirements from the input model have been fulfilled in case of before task structure optimization. For example, member 36 assigned to task 4 has only multifunctional knowledge rating 0.449(G), where the multifunctional knowledge rating threshold required for this task indicates 0.654(VG). Please note that a similar situation happens in case of task 12 and 20. Task 12 with required 0.654(VG) accepted member 36 0.449(G), where task 20 with required 1(E) accepted members 27 and 28 both with 0.449(G). This is due to the member's schedule availability conflict, so that the only other members that can still be assigned to such tasks are the ones with a lower than required multifunctional knowledge rating.

Overall, the ultimate research goal is achieved. The right team members are assigned to the right tasks at the right time in the concurrent engineering environment.

Table 5. Task-Member Assignment Solution before GA Task Optimization

Start	Name		Time (hr)	Cost	Information In	T_{WF}	Organization				Department				Member				K_t
1	T1	Business Offer Proposition	150	40	5	100%	1				9				32				0.654(VG)
	T2	Order Specification Analysis for Production Feasibility	16	80	1	300%	1				3	2	10		6	7	31		0.654(VG)
	T3	Technical Concept Proposition	60	40	1	100%	1				9				32				0.654(VG)
	T4	Price Evaluation	40	40	1	100%	1				9				36				*0.449(G)
	T5	Project Initiation	30	40	6,7	400%	1				7	12	6	3	20	40	19	11	1(E)
	T6	Project Design Documentation	1619	40	8,9,10	100%	1				2				7				1(E)
	T7	Project Components Documentation	16	40	11	100%	1				2				3				0.654(VG)
	T8	Project Specification Check/ Changes	308	40	9,10,26	200%	1				2	2			4	11			0.449(G)
	T9	User Manual Documentation	86	40	24	200%	1				2	2			3	4			0.449(G)
	T10	Past Experience Technology Consultation	45	28	11	200%	1				1	1			37	38			1(E)
	T11	Technology Documentation	324	40	8,12,16,17,18,19,20,21,22	200%	1				12	12			39	40			0.654(VG)
	T12	Evaluation/Selection of procurement partners	42	35	13,15	200%	1				9	9			31	36			*0.449(G)
	T13	Material Procurement Ordering	100	35	14	100%	1				9				31				0.654(VG)
	T14	Material Procurement Receiving	100	28	15,17,18,20,21	200%	1				5	5			14	15			0.654(VG)
	T15	Project Material Evaluation/ Specification Changes	5	35	8	100%	1				3				11				0.654(VG)
	T16	Evaluation/Selection of supply chain management partners	452	40	17	100%	1				11				38				1(E)
	T17	Detail Analysis of Supply Chain Management Value Proposition	1892	28	19,22	100%	1				11				1				0.654(VG)
	T18	Detail Analysis of Firms Value Proposition	270	28	19,22	100%	1				8				22				1(E)
	T19	Assembly of furnace subparts	1622	28	20,21,22	400%	1				8	8	8	8	22	23	24	29	1(E)
	T20	Assembly of main furnace part	2387	28	21,22	400%	1				8	8	8	8	22	23	27	28	*0.449(G)
	T21	Final Assembly	537	28	8,22	200%	1				8	10			22	23			1(E)
	T22	Final Testing	250	32	8,23	200%	1				10	10			31	34			0.654(VG)
	T23	Disassembly and painting	347	28	24	300%	1				8	8	8		22	23	24		1(E)
	T24	Transportation	288	28	25	200%	1				4	4			12	15			1(E)
	T25	Assembly over the customer site	353	84	26	400%	1				10	10	10	10	31	34	35	36	0.654(VG)
	T26	Furnace Assembly Inspection	259	32	8, 27	100%	1				10				36				1(E)
	T27	Final Meeting	12	40	-	100%	1				13				42				1(E)

** - indicates a change in T_{MKR} from the original input*

Table 6. Task-Member Assignment Solution after GA Task Optimization

Start	Name		Time (hr)	Cost	Information In	W	Organization				Department				Member				MKR threshold
1	T2	Order Specification Analysis for Production Feasibility	16	80	1	300%	1				3	2	10		6	7	35		0.654(VG)
	T3	Technical Concept Proposition	60	40	1	100%	1				9				32				0.654(VG)
	T4	Price Evaluation	40	40	1	100%	1				9				31				0.654(VG)
	T1	Business Offer Proposition	150	40	5	100%	1				9				32				0.654(VG)
	T5	Project Initiation	30	40	6,7	400%	1				7	12	6	3	20	40	19	11	1(E)
	T7	Project Components Documentation	16	40	11	100%	1				2				3				0.654(VG)
	T6	Project Design Documentation	1619	40	8,9,10	100%	1				2				7				1(E)
	T10	Past Experience Technology Consultation	45	28	11	200%	1				1	1			37	38			1(E)
	T11	Technology Documentation	324	40	8,12,16,17,18,19,20,21,22	200%	1				12	12			39	40			0.654(VG)
	T12	Evaluation/Selection of procurement partners	42	35	13,15	200%	1				9	9			33	34			0.654(VG)
	T16	Evaluation/Selection of supply chain management partners	452	40	17	100%	1				11				38				1(E)
	T13	Material Procurement Ordering	100	35	14	100%	1				9				31				0.654(VG)
	T14	Material Procurement Receiving	100	28	15,17,18,20,21	200%	1				5	5			15	16			0.654(VG)
	T18	Detail Analysis of Firms Value Proposition	270	28	19,22	100%	1				8				22				1(E)
	T15	Project Material Evaluation/ Specification Changes	5	35	8	100%	1				3				11				0.654(VG)
	T17	Detail Analysis of Supply Chain Management Value Proposition	1892	28	19,22	100%	1				11				1				0.654(VG)
	T19	Assembly of furnace subparts	1622	28	20,21,22	400%	1				8	8	8	8	22	23	24	29	1(E)
	T20	Assembly of main furnace part	2387	28	21,22	400%	1				8	8	8	8	22	23	24	29	1(E)
	T21	Final Assembly	537	28	8,22	200%	1				8	8			22	23			1(E)
	T22	Final Testing	250	32	8,23	200%	1				10	10			31	34			0.654(VG)
	T8	Project Specification Check/ Changes	308	40	9,10,26	200%	1				2	2			4	11			0.449(G)
	T23	Disassembly and painting	347	28	24	300%	1				8	8	8		22	23	24		1(E)
	T9	User Manual Documentation	86	40	24	200%	1				2	2			3	4			0.449(G)
	T24	Transportation	288	28	25	200%	1				4	4			12	15			1(E)
	T25	Assembly over the customer site	353	84	26	200%	1				10	10	10	10	31	34	35	36	0.654(VG)
	T26	Furnace Assembly Inspection	259	32	8, 27	100%	1				10				36				1(E)
	T27	Final Meeting	12	40	-	100%	1				13				42				1(E)

Solution Analysis

Departmental Flow Analysis (DFA)

In order to estimate the project completion time and cost more accurately, project managers have to consider the "weight" of each feedback connection in the project, such as the probability of a feedback to happen and/or the number of loops (or iterations) that a feedback connection should incur. A higher feedback probability with more feedback iterations means a stronger weight should be given to the feedback connection, which will significantly increase the project time and cost.

Figures 14 and 15 on the next 2 pages show the task flows among departments before and after the GA optimization. The task numbers from 1 to 27 are within circles. Member assigned to the task are presented as plain numbers next to the task. The abbreviations within squares stand for the names of the departments where the tasks are performed. The solid and dashed lines represent the progressive and feedback connections, respectively. Such departmental flow diagram will be useful to help the managers evaluate the weights of feedback connections. In general, the feedback connections involving the tasks/members that belong to the same department tend to have less iteration than those feedbacks with the tasks/members from different departments. Moreover, the higher number of members involved in the feedback the more costly and lengthy the feedback can be expected. In addition, the departments with members that are willing to share and cooperate with each other in a feedback loop are also likely to have less iteration than the departments/ members that lock themselves up from cooperation.

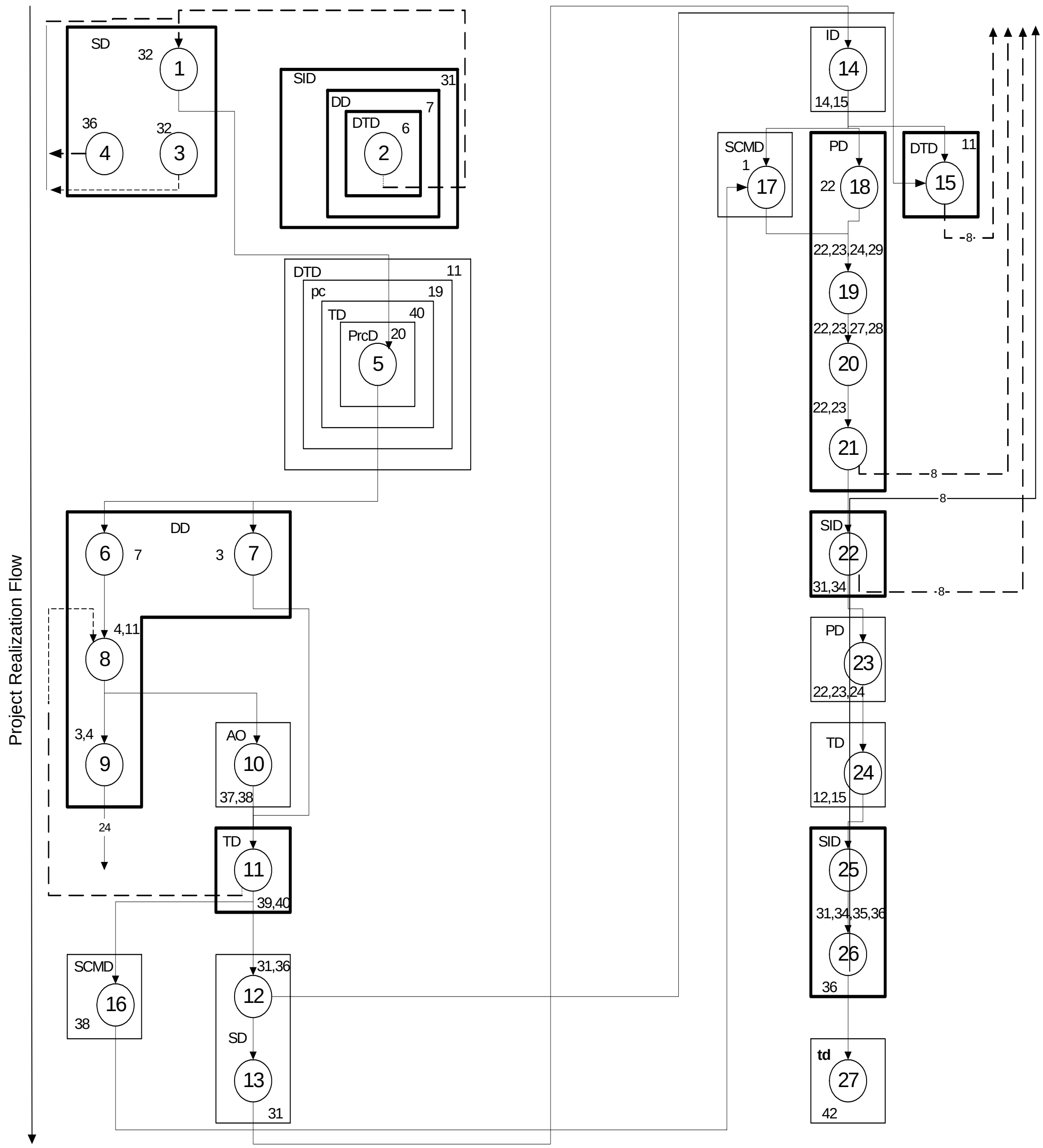

Figure 14. Project Departmental Flow before Optimization

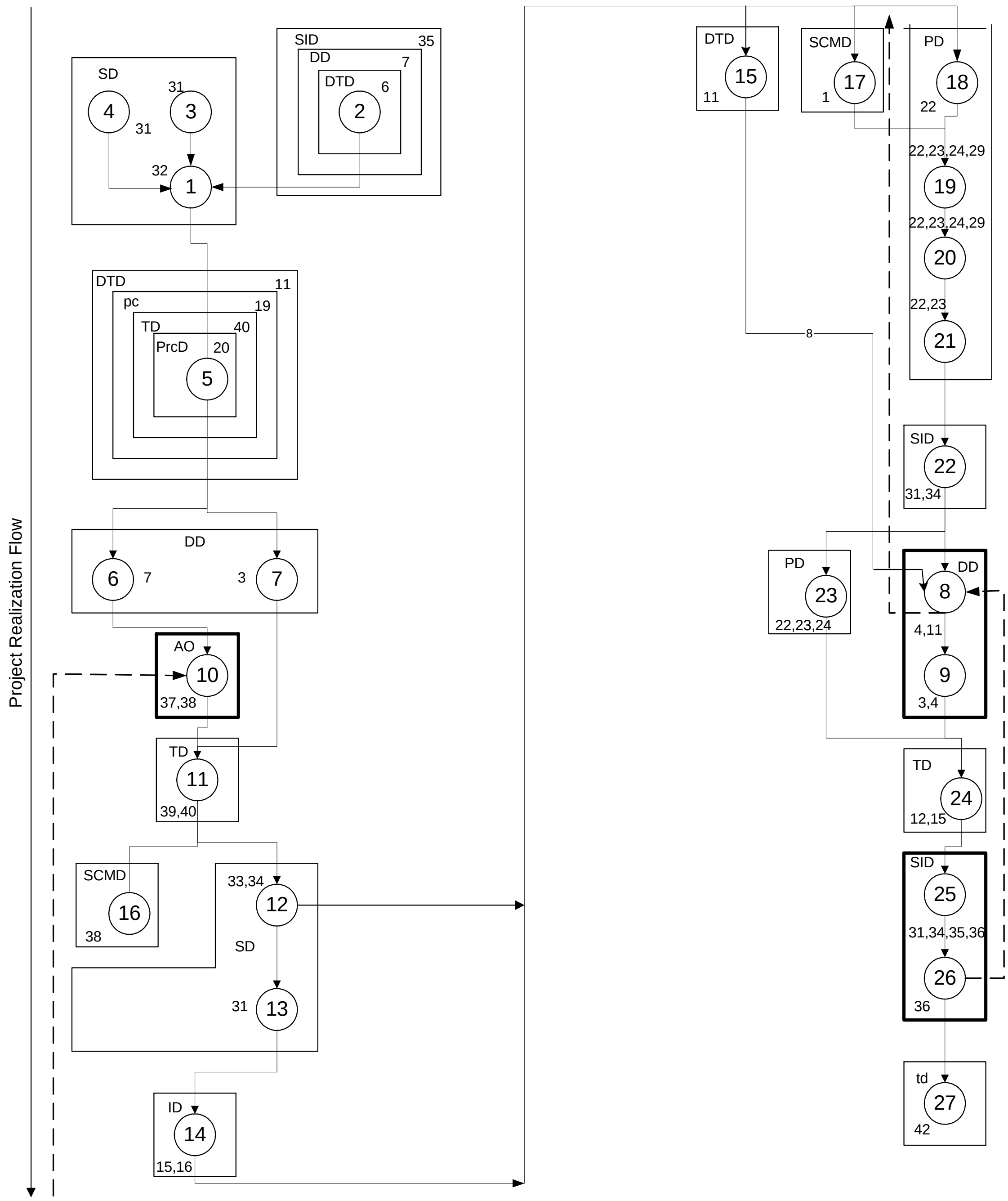

Figure 15. Project Departmental Flow after Optimization

In Figure 14 and 15 above, task 3 and task 4 both having a feedback connection to task 1 are expected to receive better coordination and cooperation with task 1 because these three tasks are to be executed by the members that belong to the same department (SD). However, in Figure 14 (before project structure optimization) task 4 was assigned a member (36) with a lower than required multifunctional knowledge rating. This can potentially lengthen the project execution or even increase the overall project cost. Handling the feedback connection from task 2 to task 1 is more difficult because task 2 involves three different departments (DTD, DD, and SID) that all have to deal with the department (SD) of task 1. Therefore, the feedback connection of task 2 to task 1 is given a stronger weight than the feedbacks of task 3 to task 1 and task 4 to task 1.

Compared the departmental flow of Figure 14 (before optimization) with Figure 15 (after optimization), it can be seen that the level of cooperation among departments is much improved after optimization. The feedback loops in Figure 14 involve six functional departments and total of 13 members (DTD (6, 11), DD (4, 7, 11), SID (31, 34), SD (32, 36), TD (39, 40), and PD (22, 23)) for cooperation, which will be difficult for managers to coordinate: DTD (6) , DD (7) , and SID (31) with SD (32, 36) (between tasks 1 and 2); TD (39, 40) with DD (4, 11) (between tasks 8 and 11); DTD (11) with DD (4, 11) (between tasks 8 and 15); PD (22, 23) with DD (4, 11) (between tasks 8 and 21); SID (31, 34) with DD (4,11) (between tasks 8 and 22). In contrast, the two feedback loops in Figure 15 need only three functional departments and total of 5 members (DD (4, 11), AO (37, 38) and SID (36) to cooperate with each other during the project execution: DD (4, 11) with AO (37, 38) (between tasks 8 and 10) and SID (36) with DD (4, 11) (between tasks 8 and 26).

A few important recommendations should be made. For example, in after task structure optimization solution, in task 10, the members 37 and 38 from the AO department, are expected to maintain close communication with the members 4, and 11 from the DD department where task 8 belongs in order to avoid the unnecessary rework. Moreover, task 6 undertaken by member 7, which provides information input to task 10, also belongs to the DD department. Therefore, managers should give more emphasis to the level of cooperation and communication between members from the departments of AO and DD. The above example shows that the proposed model is able to improve the project task flow, select the right team members, and then assigned them to the right tasks at the right time resulting in less bureaucratic processes, better coordination for the information exchange, and thus reduces the total time and cost of the project.

CHAPTER 5

SUMMARY, CONCLUSIONS, AND FUTURE RESEARCH

This research tackles one very important yet largely ignored question in concurrent engineering "*how do we assign the right members to the right tasks at the right time in the concurrent engineering environment?*" Complex problems with large number of independent, dependent, and interdependent tasks usually require the expertise from various functional departments to be accomplished successfully. To accomplish the ultimate goal of assigning the right people to work on the right tasks in the right time academia and industry should pay more attention to ways of integrating the process task optimization with member's technical, teamwork, as well as working relationship capabilities.

This research provides an integrated methodological framework for task-member assignment, so that complex concurrent engineering projects can be planned for optimal project task flow with improved management of the valuable member's capabilities.

Summary

In this research, an integrated methodological framework that integrates the project tasks and team members has been developed. It includes three components: a project task model, a team member model, and a task-member assignment model. The effectiveness of the developed methodology has been successfully demonstrated on an illustrative example throughout the three models, in which a project with 27 tasks and 1 organization with 42 members and 13 departments are used.

In the project task model, we have presented the coordination of independent, dependent, and interdependent tasks of a project in a concurrent engineering environment using DSM and GA. DSM helps identify the relationships among project tasks (i.e., independent, dependent, and interdependent task relations). GA is used to help coordinate/optimize the project task structure in terms of task cost, task time and their coupling strength. This model is an extension of the project task model developed by Chen and Lin (2002, 2003, 2004) and Rohatynski, Kielec and Sasiadek (Kielec, 2002; Rohatynski *et al.,* 2002; Rohatynski and Kielec, 2001, 2002).

In the team member model, we have developed a quantitative representation scheme of three team member characteristics, namely multifunctional knowledge, teamwork capability and working relationship. A pairwise comparison using AHP and absolute measurement are applied to extract each member's multifunctional knowledge rating and teamwork capability rating. Additionally, Myers-Briggs Type Indicator (MBTI), as a proven research in personality profiling, is used in assessing the working relationship rating between team members. This model is an extension of the team member model developed by Chen and Lin (2004).

In the task-member assignment model, we have developed a genetic algorithm with a newly designed fitness function including all members' characteristics from the team member model as well as the function that allow us to calculate the best possible solution representing the desired (optimal) solution. Having the optimized task structure from the project task model, right members from the team member model, task-member assignment model with the additional help from member's workload schedule database can achieve the ultimate research goal of this study - assign the right team members to the right tasks at the right time in the concurrent engineering environment.

Conclusions

While tasks and teams are usually treated separately in both academia and industry, this research integrates both task coordination and project team organization in concurrent engineering. In conclusion, this research has the following major contributions:

1) The project task model provides a GA optimization approach with a special designed fitness function so that an optimal project task flow can be established based on individual task time and/or cost, as well relationships among them.
2) The quantitative representation of three important team member characteristics makes it possible to establish a successful multifunctional team considering the quantitative ratings of individual member expertise, teamwork capability and their collegiality. As a result, we can organize teams with members who not only have expertise in their functional area, but also are capable of teamwork skills and enjoy a good working relationship.
3) The methodological framework integrates task coordination and team organization in the task-member assignment with no limitation on complexity of the project task

structure or the number of members to be considered for assignment. Consequently, task-member assignment in complex concurrent engineering project can be successfully planned and executed.

Recommendations for Future Research

While this research provides a methodology for the task-member assignment in concurrent engineering, it does not fully consider the possibility of multi-project/multi-organization environment. In reality, teams are often required to cooperate over space and time, where members are drawn from different organizations and even different locations. For example, when two or more projects overlap in time and there is a need to recruit members from different organization that could all possibly work on the particular tasks, what would be the decision procedure to assign a specific member to a specific task in project A and not in project B? The model should be extended for such cases. To accomplish this, team member's characteristics corresponding to cooperation over space and time as well as information input for project importance level should be taken into account. Thus, assigning the right member to the right task at the right time in the multi-project/multi-organization environment can be realized.

The future work will also concentrate on the working relationship rating, an important element of this thesis, which proved to be especially difficult to quantify. Please note that if two members who are not recommended to work with each other due to a poor working relationship rating (quantified by MBTI) between them, they may not be necessary counter-productive to the team performance and may bring creativities to the team.

This research can also be extended towards the resource assignment and allocation. This will accommodate the needs of wider variety of projects that require numerous recourses to be undertaken. For example, a construction-engineering project that requires not only the human resources but also some heavy machinery like cranes, cutting tools or excavators. In such situation, it is clear to us that the need to extend this research to multiple resources assignment and allocation exists.

Finally, the future research will compare the other search methods with GA to see if comparable or better results can be achieved in shorter/longer computational time. This part of the future work will specifically focus on the multivariable mathematical models, Tabu Search and Neural Networks. An example of a quadratic assignment formulation for task-member assignment model, considering only one-time period, can be seen in Appendix F.

REFERENCES

Ahmadi, R., Roemer, T., and Wang, R., "Structuring Product Development Processes," *European Journal of Operational Research*, Vol. 130, pp. 539-558, 2001

Allen, T., "Organizational Structure, Information Technology, and R&D Productivity," *IEEE Transactions on Engineering Management*, Vol. EM-33, No. 4, pp. 212-217, 1986

Baran, J., "CIGNA Begins Its Quality Transformation," *Quality Progress*, Vol. 25, No. 1, pp. 42-47, 1992

Barnes, C.J, "A Methodology For The Concurrent Design of Products And Their Assembly Sequences," Phd Thesis , 2000

Barnes, C.J., Jared, G.E.M., Swift, K.G., "A Pragmatic Approach to Interactive Assembly Sequence Evaluation," Proceedings of the Institution of Mechanical Engineers. Part B, Engineering Manufacture, Vol. 217, No.4, pp. 541-550, 2003

Basak, I., "On the Use of Information Criteria in Analytic Hierarchy Process," *European Journal of Operational Research,* Vol. 141, pp. 200-216, 2002

Bayne, R., "A New Direction for the Myers-Briggs Type Indicator," *Personnel Management*, Vol. 22, No. 3, pp. 48-51, 1990

Blackburn, J.D., Hoedemaker, G.M., and Van Wassenhove, L.N., "Concurrent Software Engineering: Prospects and Pitfalls," *IEEE Transactions on Engineering Management*, Vol. 43, No. 2, 1996

Blackwell, G.W., "Multidisciplinary Team Research," *Interdisciplinary Analysis and Research,* Chubin, D.E. *et al*. Eds., Lomond, pp. 103-114, 1986

Browning, T.R., "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions," *IEEE Transactions on Engineering Management,* Vol. 48, No. 3, pp. 292-306, 2001.

Boudreau, M., Loch, K.D., Robey, D., and Straud, D., "Going Global: Using Information Technology to Advance the Competitiveness of Virtual Transnational Organization," *The Academy of Management Executive*, Vol. 12, No. 4. p.120, 1998

Campion, M.A., Medsker, G.J. and Higgs, A.C., “Relations between Work Group Characteristics and Effectiveness: Implications for Designing Effective Work Groups,” *Pers. Psychology*, Vol. 46, pp. 823–850, 1993

Chance, P., "Great Experiments in Team Chemistry," *Across the Board*, Vol. 26, No. 5, pp. 18-25, 1989

Chen, S.-J. and Lin, L., "A Project Task Coordination Model for Team Organization in Concurrent Engineering," Concurrent Engineering: Research and Applications, Vol. 10, No. 3, pp. 187-203, 2002

Chen, S.-J. and Lin, L., "Decomposition of Interdependent Task Group for Concurrent Engineering," Computers and Industrial Engineering, Vol. 44, No. 3, pp. 435-459, 2003

Chen, S.-J. and Lin, L., "Modeling Team Member Characteristics for the Formation of a Multifunctional Team in Concurrent Engineering," *IEEE Transactions on Engineering Management*, Vol. 51, No. 2, pp. 111-124, 2004

Cho, S-H., “An Integrated Method for Managing Complex Engineering Projects Using the Design Structure Matrix and Advanced Simulation,” MIT MS Thesis, Mech. Eng., 2001

Chung, W.Y. and Guinan, P.J., “Effects of Participative Management,” Proceedings of the ACM SIGCPR Conference, pp. 252-260, 1994

Church, A.H. and Waclawski, J., "The Effects of Personality Orientation and Executive Behavior on Subordinate Perceptions of Workgroup Enablement," *The International Journal of Organizational Analysis*, Vol. 4, No. 1, pp. 20-51, 1996

Clark, K.B. and Wheelwright, S.C., “Organizing and Leading “Heavyweight” Development Teams,” *California Management Review*, Vol. 34, No. 3, pp. 9-28, 1992
Glover F., and Laguna M., “Tabu Search in Modern Heuristic Techniques for Combinatorial Problems,” C.R. Reeves, editor, John Wiley & Sons, Inc, 1993

Coe, C.K., "The MBTI: Potential Uses and Misuses in Personnel Administration," *Public Personnel Management*, Vol. 21, No. 4, pp. 511-522, 1992

Coello, C., "An Updated Survey of GA-based Multiobjective Optimization Techniques," *ACM Computing Surveys*, Vol.32, No.2, pp. 109-143, 2000

Cofsky, B., "A Journey toward Excellence," *Management Accounting*, Vol. 77, No. 7, pp. 14, 1996

Converse, S.A., Cannon-Bowers, J.A., and Salas, E., “Team Member Shared Mental Models: A Theory and Some Methodological Issues,” in Proc. Human Factors Society 35th Ann. Meeting, Vol. 2, pp. 1417–1421, 1991

Dawkins, R., “The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design,” W.W. Norton, 1996

Dong, Q. and Whitney, D., "Designing a Requirement Driven Product Development Process," Proceedings of the 13th International Conference on Design Theory and Methodology (DTM 2001), Pittsburgh, Pennsylvania, USA, 2001

Duarte, D.L. and Snyder, N.T., ”Mastering Virtual Teams,” San Francisco: Jossey-Bass, 1999

Edgley, G.J., "Type and Temperament," *Association Management*, Vol. 44, No. 10, pp. 83-92, 1992

Eppinger, S.D., Whitney, D.E., Smith, R.P., and Gebala, D.A., “Organizing the Tasks in Complex Design Projects,” *ASME Design Theory and Methodology*, Vol. 27, pp. 39-46, 1990

Eppinger, S.D., Whitney, D.E., Smith, R.P. and Gebala, D.A., “A Model-Based Method for Organizing Tasks in Product Development,” *Research in Engineering Design*, Vol. 6, No. 1, pp. 1-13, 1994

Eppinger, S.D., "Innovation at the Speed of Information," *Harvard Business Review*, Vol. 79, No. 1, pp. 149-158, 2001

Eppinger, S.D. and Salminen, V., "Patterns of Product Development Interactions," International Conference on Engineering Design, Glasgow, Scotland, 2001

Fleming, P. and Purshouse, R.C., "Evolutionary Algorithms in Control Systems Engineering: A Survey," *Control Engineering Practice*, Vol.10, pp. 1223-1241, 2002

Forman, E., Selly, M. A., “Decission by Objectivs,” 1998

Forrest, S., "Genetic Algorithms: Principles of Natural Selection Applied to Computation." *Science*, Vol. 261, pp. 872-878, 1993

Gardner, W.L. and Martinko, M.J., "Using the Myers-Briggs Type Indicator to Study Managers: A Literature Review and Research Agenda," *Journal of Management*, Vol. 22, No. 1, pp. 45-83, 1996

Gebala, D.A. and Eppinger, S.D., "Methods for Analyzing Design Procedures," *ASME Design Theory and Methodology*, DE-Vol. 31, pp. 227-233, 1991

Girffin, A and Hauser, J.R., "Pattern of Communication among Marketing, Engineering and Manufacturing: A Comparison Between Two New Products," *Management Science*, Vol. 38, No. 3, pp. 360-373, 1992

Goldberg, D., "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley Publishing Co., New York, 1989.

Guthrie, J., "The Myers-Briggs Type Indicator at the Australian Management College Mt Eliza," *Practicing Manager*, Vol. 13, No. 2, pp. 11-18, 1993

Guo, W., Cha, J., Fang, Y. and Woo, T.C., "Concurrent Design: An Algebraic Perspective," *International Journal of Information Technology*, Vol. 1, No. 1, pp. 17-32, 1995

Haas, R. and Sinha, M., "Concurrent Engineering at Airbus", *International Journal of Manufacturing Technology and Management (IJMTM),* Vol. 6, No. 3/4, 2004

Haque, B., Pawar, K. S., and Barson, R. J., "Analyzing Organizational Issues in Concurrent New Product Development," *International Journal of Production Economics*, Vol. 67, pp. 169–182, 2000

Haupt, R. and Haupt, S. E., "Practical Genetic Algorithms," John Wiley & Sons, 1998

Hildebrand, C., "Staying Ahead of the Class," CIO Perspectives Supplement, pp. 2-3, 1995

Hilliard, D.K., "The Communicator's Role in Project Teams," *IEEE International Professional Communication Conference*, pp. 315-320, 1993

Hirshhorn, L. and Gilmore, T., "The New Boundaries of the "Boundaryless" Company," *Harvard Business Review*, Vol. 70, No. 3, pp. 104-115, 1992

Hoedemaker, G.M., Blackburn, J.D., and Van Wassenhove, L.N., "Limits to Concurrency," Working Paper in INSEAD, Fontainebleau, France, January 1995

Holland, J., "Genetic algorithms." *Scientific American*, p. 66-72, 1992

Holt, R.J., Barnes, C.J., "A Framework For Integrating Product and Process Design," 4th International Conference On Integrated Design and Manufacturing In Mechanical Engineering, 2002

Jaffe, J.M., "Of Different Minds," *Association Management*, Vol. 37, No. 10, pp. 120-124, 1985

Jarvenpaa, S. L., Ives, B., “The Global Network Organization of the Future: Information Management Opportunities and Challenges,” *Journal of Management Information Systems*, Vol. 10, No. 4, pp. 25-57, 1994.

Jarvenpaa, S. L., and Leider, D. E., “Communication and Trust in Global Virtual Teams,” *Journal of Computer Mediated Communication*, Vol. 3, No. 4, 1998

Jung, C., “The Integration of Personality,” Farrar & Ruchart, New York, NY, 1953

Jyoti K. Parikh and Sudhakar Y., “Techno-Economic Assessment of Alternative Options in Urban Transport Sector for Delhi and Mumbai,” Under the Project of Analysis of Technological Alternatives for Mitigation of GHG Emissions from Urban Transport Sector in Selected Asian Cities (SIDA-AIT sponsored project), 2001

Kielec, R., ”O wyższości algorytmów ewolucyjnych nad dotychczasowymi metodami optymalizacyjnymi”, Zaklad Podstaw Konstrukcji Maszyn i Projektowania Wspolbieznego na Uniwesytecie Zielonogorskim, 2002

Killander, A.J., “Concurrent Development Requires Uncoupled Concepts and Projects,” ISPE International Conference on Concurrent Engineering, CE95, McLean, Virginia, pp. 22-25, 1995

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., "Optimization by Simulated Annealing," *Science*, Vol.220, pp. 671-678, 1983

Koza, J., Bennett, F., Andre, D., and Keane, M., “Genetic Programming III: Darwinian Invention and Problem Solving,” Morgan Kaufmann Publishers, 1999

Koza, J., Keane, M., Streeter, M., Mydlowec, W., Yu, J., and Lanza, G., “Genetic Programming IV: Routine Human-Competitive Machine Intelligence,” Kluwer Academic Publishers, 2003.

Krishnan, V., Eppinger, S.D., and Whitney, D.E., “A Model-Based Framework to Overlap Product Development Activities,” *Management Science*, Vol. 43, pp. 437-451, 1997

Kusiak, A. and Wang, J., “Decomposition in Concurrent Design,” *Concurrent Engineering: Automation, Tools, and Techniques*, John Wiley & Sons, New York, NY, pp. 481-507, 1993a

Kusiak, A. and Wang, J., “Efficient Organizing of Design Activities,” *International Journal of Production Research*, Vol. 31, No. 4, pp. 753-769, 1993b

Kusiak, A. and Wang, J., “Decomposition of the Design Process,” *Journal of Mechanical Design*, Vol. 115, No. 4, pp. 687-695, 1993c

Kusiak, A., Larson, T.N. and Wang, J., “Reengineering of Design and Manufacturing Processes,” *Computer Industry Engineering*, Vol. 26, No. 3, pp. 521-536, 1994

Kusiak, A. and Larson, T.N., “Decomposition and Representation Methods in Mechanical Design,” *Journal of Mechanical Design*, Vol. 117, No. 3, pp. 17-24, 1995

Kusiak, A., Wang, J., He, D.W., and Feng, C., “A Structured Approach for Analysis of Design Processes,” *IEEE Transactions on Components, Packaging, and Manufacturing Technology-Part A*, Vol. 18, No. 3, pp. 664-673, 1995

Lawson, M. and Karandikar, H.M., “A Survey of Concurrent Engineering,” *Concurrent Engineering: Research and Applications*, Vol. 2, No. 1, pp. 1-6, 1994

Lee, D.M.S., “Management of Concurrent Engineering: Organizational Concepts and A Framework of Analysis,” *Engineering Management Journal*, Vol. 4, No. 2, June, pp. 15-25, 1992

Levasseur, R.E., "People Skills: What Every Professional Should Know About Designing and Managing Meetings," *Interfaces*, Vol. 22, No. 2, pp. 11-14, 1992

Lin, Z. and Yang, C., "Evaluation of machine selection by the AHP method," *Journal of Materials Processing Technology*, Vol. 57, pp. 253-258, 1996

Lyman, D. and Richter, K., "QFD and Personality Type: The Key to Team Energy and Effectiveness," *Industrial Engineering*, Vol. 27, No. 2, pp. 57-61, 1995

Logan, L. R., “Team Members Identify Key Ingredients for Team-Building Success,” *National Productivity Review*, pp. 209-223, 1993

MacDonald, W.R., "Characteristics of Interdisciplinary Research Teams," *Interdisciplinary Analysis and Research,* Chubin, D.E. *et al*. Eds., Lomond, pp. 395-406, 1986

Marczyk, A., “Genetic Algorithm and Evolutionary Computation,” The Talk. Origins Archive, 2004 (http://www.talkorigins.org/faqs/genalg/genalg.html)

Maxon, J., "Innovators, Entrepreneurs, and Intrapreneurs: Food for Thought," *Industrial & Commercial Training*, Vol. 18, No. 5, pp. 13-16, 1986

McClure, L. and Werther, W.B. Jr., "Personality Variables in Management Development Interventions," *Journal of Management Development*, Vol. 12, No. 3, pp. 39-47, 1993

McCulley, C. and Bloebaum, C.L., "Optimal Sequencing for Complex Engineering Systems Using Genetic Algorithms," *American Institute of Aeronautics and Astronautics*, AIAA-94-4327-CP, pp. 718-730, 1994

McCulley, C. and Bloebaum, C.L., "A Genetic Tool for Optimal Design Sequencing in Complex Engineering Systems," *Structural Optimization*, Vol. 12, pp. 186-201, 1996

McCulley, C. and Bloebaum, C.L., "Comparison of Heuristic Convergence Strategies for Multidisciplinary Analysis," *American Institute of Aeronautics and Astronautics*, AIAA-98-4943, pp. 1-11, 1998

McCulley, C., Hulme, K. and Bloebaum, C.L., "Simulation-Based Development of Heuristic Strategies for Complex System Convergence," *Applied Mechanics Review*, Vol. 50, No. 11, pp. 117-124, 1997

McKee, B., “Turn your workers into a team,” *Nation’s Bussines*, pp. 36–38,July 1992

Mitchell, M., “An Introduction to Genetic Algorithms,” MIT Press, 1996

Moore, T. and Woods, W., "Personality Tests Are Back," *Fortune*, Vol. 115, No. 7, pp. 74-82, 1987

Myers, I.B., “Introduction to Type,” Consulting Psychologists Press, Palo Alto, CA, 1980a

Myers, I.B., “Gifts Differing,” Consulting Psychologists Press, Palo Alto, CA, 1980b

Myers, I.B. and McCaulley, M.H., “Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator,” Consulting Psychologists Press, Palo Alto, CA, 1985

Nicholas, J.M., “Concurrent Engineering: Overcoming Obstacles to Teamwork,” *Production & Inventory Management Journal*, Vol. 35, No. 3, pp. 18–22, 1994

O’Grady, P. and Young, R.E., “Issues in Concurrent Engineering Systems,” *Journal of Design and Manufacturing*, Vol. 1, pp. 27-34, 1991

O'Grady, P., Young, R.E., Greef, A., and Smith, L., “An Advice System for Concurrent Engineering,” *International Journal of Computer Integrated Manufacturing*, Vol. 4, No. 2, pp. 63-70, 1991

Orady, E.A. and Shareef, I., “Expert System Design Philosophy for Application of Simultaneous Engineering in Industry,” *ASME Design for Manufacturability*, DE-Vol. 52, pp. 151-157, 1993

Partovi, F.Y. and Burton, J., "An Analytical Hierarchy Approach to Facility Layout", *Computers in Industrial Engineering*, Vol. 22, No. 4, pp. 447-457, 1992

Partovi, F.Y. and Hopton, W.E., "The analytic hierarchy process as applied to two types of inventory problems", *Production and Inventory Management Journal*, Vol. 6, pp. 13-19, 1994

Pawar, K. S. and Sharifi, S., “Physical or virtual team collocation: Does it matter?,” *International Journal of Production Economics*, Vol. 52, pp. 283-290, 1997

Petrie, H.G., "Do You See What I See? The Epistemology of Interdisciplinary Inquiry," *Interdisciplinary Analysis and Research,* Chubin, D.E. *et al*. Eds., Lomond, pp. 115-130, 1986

Prasad, B., *Concurrent Engineering Fundamentals: Integrated Product and Process Organization*, Prentice Hall International Series, New Jersey, NJ, 1996

Prasad, B., “Decentralized Cooperation: A Distributed Approach to Team Design in a Concurrent Engineering Organization,” *Team Performance Management*, Vol. 4, No. 4, pp. 138-165, 1998

Price, R.C., "TQM in the R&D function," *Quality Progress*, Vol. 28, No. 7, pp. 109-111, 1995

Prince, A., Brannick, M.T., Prince, C. and Salas, E., "Team Process Measurement and Implications for Training," Proceedings of the Human Factors Society 36th Annual Meeting, San Francisco, CA, Vol. 2, pp. 1351-1355, 1992

Robinson, S., "Versatility in Managers Beats Extremes into Teams," *Rydge's*, Vol. 58 No. 10, pp. 128-130, 1985

Rogers, J.L., “DeMaid: A Design Manager’s Aide for Intelligent Decomposition User’s Guide,” NASA Technical Memorandum 101575, March, 1989a

Rogers, J.L., “A Knowledge-Based Tool for Multilevel Decomposition of a Complex Design Problem,” NASA Technical Paper 2903, May, 1989b

Rogers, J.L. and Bloebaum, C.L., “Ordering Design Tasks Based on Coupling Strengths,” Proc. 5th AIAA Symp. on Multidisciplinary Analysis and Optimization, pp. 708-717, 1994

Rohatyński, R. and Kielec, R., “Artificial Evolution in Design Process Optimization”, Proceedings of the International Conference ‘Computer Integrated Manufacturing’, CIM, Vol. 2, pp. 139-150, 2001

Rohatynski, R. and Kielec, R., ”Artificial Evolution in Design Process Optimization,” Zaklad Podstaw Konstrukcji Maszyn i Projektowania Wspolbieznego na Uniwesytecie Zielonogorskim, 2002

Rohatyński, R., Kielec, R., Sąsiadek, M., “Implementation of Matrix Method and Evolution Algorithm for Reorganization of Design Processes,” Third International Seminar and Workshop on Engineering Design in Integrated Product Development (EDIProD’2002), Zielona Góra – Łagów, Wydawnictwa Naukowo-Techniczne UZ, Zielona Góra, pp. 157-158, 2002

Saaty, T.L., “The Analytic Hierarchy Process,” McGraw-Hill, New York, 1980

Saaty, T.L., “How to Make A Decision: The Analytic Hierarchy Process,” *European Journal of Operational Research*, Vol. 48, pp. 9-26, 1990

Satty, T.L., "Decision Making - The Analytic Hierarchy and Network Processes (AHP/ANP)," *Journal of Systems Science and Systems Engineering*, published at Tsinghua University, Beijing, Vol. 13, No. 1, pp 1-35, 2004a

Satty, T.L., "Fundamentals of the Analytic Network Process: Dependence and Feedback in Decision-Making with a Single Network," *Journal of Systems Science and Systems Engineering*, published at Tsinghua University, Beijing (to appear), Vol. 13, No. 2, 2004b

Satty, T.L., "Fundamentals of the Analytic Network Process: Multiple Networks with Benefits, Costs, Opportunities and Risks," *Journal of Systems Science and Systems Engineering*, published at Tsinghua University, Beijing (to appear), Vol. 13, No. 3, 2004c

Satty, T.L., "Automatic Decision-Making: Neural Firing and Response," *Journal of Systems Science and Systems Engineering*, published at Tsinghua University, Beijing, (to appear), Vol. 13, No. 4, 2004d

Satty, T.L., "The Analytic Hierarchy and Analytic Network Processes for the Measurement of Intangibles and for Decision-Making," 67-page chapter in *Multiple Criteria Decision Analysis: The State of the Art Surveys*. Kluwer Academic Publishers, Dordrecht, edited by Figueira, J., Greco, S. and Ehrgott, M., 2004e

Satty, T.L., "Rank from Comparisons and from Ratings in the Analytic Hierarchy/ Network Processes" (forthcoming), *European Journal of Operations Research*, 2004f

Safoutin, M.J. and Thurston, D.L., "A Communications-Based Technique for Interdisciplinary Design Team Management," *IEEE Transactions on Engineering Management*, Vol. 40, No. 4, pp. 360-372, 1993

Sharifi, S. and Pawar, K. S., “Virtually Co-Located Product Design Teams: Sharing Teaming Experiences After the Event?,” *Int. J. Oper. Prod.Manage.*, Vol. 22, No. 6, pp. 656–679, 2002.

Smith, R.P., “The Historical Roots of Concurrent Engineering Fundamentals,” *IEEE Transactions on Engineering Management*, Vol. 44, No. 1, pp. 67-78, 1997

Smith, R.P. and Eppinger, S.D., “Identifying Controlling Features of Engineering Design Iteration,” *Management Science*, Vol. 43, No. 3, pp. 276-293, 1997a

Smith, R.P. and Eppinger, S.D., “A Predictive Model of Sequential Iteration in Engineering Design,” *Management Science*, Vol. 43, No. 8, pp. 1104-1120, 1997b

Sobieszczanski-Sobieski, J., “Multidisciplinary Optimization for Engineering Systems: Achievements and Potential,” *NASA Technical Memorandum 101566*, March, 1989

Sobieszczanski-Sobieski, J., “Multidisciplinary Design Optimization: An Emerging New Engineering Dicipline,” *NASA Technical Memorandum 107761*, May, 1993

Steward, D.V., *System Analysis and Management: Structure, Strategy and Design*, Petrocelli Books, Princeton, NJ, 1981a

Steward, D.V., “The Design Structure System: A Method for Managing the Design of Complex Systems,” *IEEE Transactions on Engineering Management*, Vol. 28, No. 3, pp. 71-74, 1981b

Stinson, T., “Teamwork in Real Engineering,” *Machine Design*, Vol. 22, March, pp. 99-104, 1990

Sudhakar Y., “Application of Analytic Hierarchy Process for the Selection of Alternative Options for Urban Transportation in India – Analysis of Group Aggregation Methodology Adopted in AHP,” A Report Submitted to Asian Institute of Technology (AIT), Bangkok, Thailand, 2001

Sundstrom, E., DeMeuse, K.P., and Futrell, D., “Work Teams, Applications and Effcetiveness,” *American Psychologist*, Vol. 45, No. 2, pp. 120-133, 1990

Sunseri, A.J. and Kosteva, D.B., "Psychological Testing Enters the Job Market," *Healthcare Financial Management*, Vol. 45, No. 10, pp. 120, 1991

Syswerda, G., ‘’Schedule Optimization Using Genetic Algorithms’’, Handbook of Genetic Algorithms, Van Nostran Reinhold, New York, 1990.

Thomas, U.P. and Eppinger, S.D., “Integration Analysis of Product Decompositions,” *ASME Design Theory and Methodology*, DE-Vol. 68, pp. 343-351, 1994

Taylor, J.B., "Building an Interdisciplinary Team," *Interdisciplinary Analysis and Research,* Chubin, D.E. *et al*. Eds., Lomond, pp. 141-154, 1986

Tomal, D.R., "Using the Right Personality Style," *Supervision*, Vol. 53, No. 10, pp. 12-13, 1992

Weldon, D., "A Mutual Understanding," *Computerworld*, Vol. 29, No. 18, pp. 103-111, 1995

Winner, R., Pennell, J., Bertrand, H., and Slusarczuk, M., “The Role of Concurrent Engineering in Weapons System Acquisition,” IDA Report R-338, Alexandria, VA: Institute for Defense Analyses, 1988

Zavbi, R. and Duhovnik, J., “The Analytic Hierarchy Process and Functional Appropriateness of Components of Technical Systems,” *Journal of Engineering Design.* Vol. 7, No. 3, pp. 313-327, 1996

Zemke, R., "Second Thoughts About the MBTI," *Training*, Vol. 29, No. 4, pp. 43-47, 1992

Yassine, A., Whitney, D., Lavine, J., and Zambito, T., "DO-IT-RIGHT-FIRST-TIME (DRFT) Approach to Design Structure Matrix (DSM) Restructuring," Proceedings of the 12th International Conference on Design Theory and Methodology (DTM 2000), September 10-13, Baltimore, Maryland, USA, 2000

Yassine, A., Whitney, D., and Zambito, T., "Assessment of Rework Probabilities for Design Structure Matrix (DSM) Simulation in Product Development Management," Proceedings of the 13th International Conference on Design Theory and Methodology (DTM 2001), September 9-12, Pittsburgh, Pennsylvania, USA, 2001

APPENDICES

APPENDIX A

MULTIFUNCTIONAL KNOWLEDGE RATING TABLE

Multifunctional Knowledge Rating Table

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13
M1	E										VG		
M2	VG										VG		
M3		E	VG										
M4		G											
M5		VG											
M6		VG	VG							P			
M7		E	E							VP			
M8		VG	E										
M9			VG							A			
M10			G										
M11		VG	E							A			
M12				E	VG		A		P				
M13				G									
M14				VG									
M15				E	VG		VP		VP				
M16				VG	E		P		P				
M17					VG		A		A				
M18					VG		VP		A				
M19						E							
M20					VP		E						
M21					A		VG						
M22								E					
M23								E					
M24								E					
M25								VG					
M26								VG					
M27								G					
M28								G					
M29								E					
M30								VG					
M31			P		P				VG	VG			
M32					VP				VG				
M33					P				VG				
M34			A		A				VG	E			
M35										VG			
M36					A				G	E			
M37	E										E		
M38	E										E		
M39												VG	
M40												E	
M41												G	
M42													E

APPENDIX B

TEAMWORK CAPABILITY RATING TABLE

Teamwork Capability Rating Table

	C1	C2	C3	TCR	TCR (normalized)
M1	E	VG	G	0.2299	0.8074
M2	G	VG	E	0.2299	0.6273
M3	E	E	E	0.3665	1
M4	VG	VG	VG	0.2517	0.6868
M5	A	G	A	0.1227	0.3348
M6	BA	A	A	0.0801	0.2186
M7	G	E	E	0.2687	0.7332
M8	E	E	E	0.3665	1
M9	VG	VG	G	0.2345	0.6398
M10	BA	G	E	0.1383	0.3774
M11	E	E	VG	0.3519	0.9602
M12	E	E	E	0.3665	1
M13	E	VG	VG	0.3131	0.8543
M14	BA	G	BA	0.0991	0.2704
M15	VG	VG	VG	0.2517	0.6868
M16	E	E	E	0.3665	1
M17	A	A	A	0.097	0.2647
M18	E	VG	E	0.3277	0.8941
M19	E	E	E	0.3665	1
M20	VG	VG	E	0.2663	0.7266
M21	VG	E	VG	0.2905	0.7926
M22	G	VG	G	0.198	0.5402
M23	G	A	VG	0.1487	0.4057
M24	A	E	E	0.2347	0.6404
M25	G	VG	E	0.2299	0.6273
M26	G	VG	E	0.2299	0.6273
M27	G	VG	A	0.1975	0.5389
M28	VG	VG	VG	0.2517	0.6868
M29	E	VG	E	0.3277	0.8941
M30	E	VG	E	0.3277	0.8941
M31	E	VG	VG	0.3131	0.8543
M32	E	G	VG	0.2723	0.743
M33	G	G	VG	0.1745	0.4761
M34	G	E	E	0.2687	0.7332
M35	VG	E	E	0.3051	0.8325
M36	A	E	VG	0.2201	0.6005
M37	E	E	E	0.3665	1
M38	VG	E	VG	0.2905	0.7926
M39	VG	VG	VG	0.2517	0.6868
M40	E	E	VG	0.3519	0.9602
M41	VG	E	VG	0.2905	0.7926
M42	E	E	E	0.3665	1

APPENDIX C

WORKING RELATIONSHIP RATING TABLE

Working Relationship Rating Table

		E S T J	E S T P	E S F J	E S F P	E N T J	E N T P	E N F J	E N F P	I S T J	I S T P	I S F J	I S F P	I N T J	I N T P	I N F J	I N F P
M10, M20	ESTJ	0.67															
M2, M11,M21,M31	ESTP	0.33	0.67														
M13, M33	ESFJ	0.83	0.5	0.67													
M4, M14, M24, M34	ESFP	0.5	0.83	0.33	0.67												
M5, M15, M25, M35	ENTJ	0.83	0.5	1	0.67	0.67											
M6, M26	ENTP	0.5	0.83	0.67	1	0.33	0.67										
M7, M17, M27, M37	ENFJ	1	0.67	0.83	0.5	0.83	0.5	0.67									
M8, M18	ENFP	0.67	1	0.5	0.83	0.5	0.83	0.33	0.67								
M9, M29, M39	ISTJ	0.5	0.17	0.67	0.33	0.67	0.33	0.83	0.5	0.33							
M32, M40	ISTP	0.17	0.5	0.33	0.67	0.33	0.67	0.5	0.83	0	0.33						
M1, M30	ISFJ	0.67	0.33	0.5	0.17	0.83	0.5	0.67	0.33	0.5	0.17	0.33					
M12, M17	ISFP	0.33	0.67	0.17	0.5	0.5	0.83	0.33	0.67	0.17	0.5	0	0.33				
M3, M23	INTJ	0.67	0.33	0.83	0.5	0.5	0.17	0.67	0.33	0.5	0.17	0.67	0.33	0.33			
M16, M36	INTP	0.33	0.67	0.5	0.83	0.17	0.5	0.33	0.67	0.17	0.5	0.33	0.67	0	0.33		
M10,M22,M28 M38	INFJ	0.83	0.5	0.67	0.33	0.67	0.33	0.5	0.17	0.67	0.33	0.5	0.17	0.5	0.17	0.33	
M19, M41, M42	INFP	0.5	0.83	0.33	0.67	0.33	0.67	0.17	0.5	0.33	0.67	0.17	0.5	0.17	0.5	0	0.33

APPENDIX D

MEMBER’S WORKLOAD SCHEDULE

Member's Workload Schedule

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M1	25%	75%			100%															
M2					100%			100%		100%								50%	75%	
M3	100%		100%				100%			100%					100%					
M4	100%			100%			100%													
M5				100%											100%	100%				
M6			100%	100%						100%					100%	100%		50%	50%	
M7			100%							100%					100%	100%				
M8	100%		100%	100%		75%	75%			100%					100%	100%				
M9	100%		100%							100%										
M10	100%		100%							100%										
M11	100%			100%												100%				
M12									100%											
M13																	100%			
M14	45%			50%	50%							50%	50%				100%			
M15												50%	25%							
M16		35%															100%			
M17	10%								100%			75%	75%							
M18	75%								100%											
M19		50%		50%																
M20																				
M21	25%																			
M22			75%	75%	75%												65%			
M23										100%							35%			
M24										100%			100%							
M25	80%									100%	100%	100%	100%		100%					
M26					25%	25%				100%	100%	100%	100%		100%					
M27										100%	100%	100%			100%					
M28		80%								100%	100%	100%			100%					
M29										100%			100%		100%					
M30										100%	100%	100%	100%		100%					
M31				100%			100%													
M32							100%	100%											100%	
M33	100%	100%						100%												
M34	100%	100%						100%												
M35		100%												100%					100%	
M36	100%						100%			50%	50%			100%						
M37							100%										65%			
M38										75%	75%									
M39			100%													50%				
M40																	60%			
M41			100%			100%														
M42								75%	75%	50%	50%									

APPENDIX E

TASK-MEMBER ASSIGNMENT GA C++ CODE

Task-Member Assignment GA C++ Code

```
// User Defined
#define MAX_POP_SIZE 100
#define MAX_TASK_NUMBER 27
#define MAX_GN 2
#define MAX_MEMBER_QUANTITY_PER_TASK 4
#define MAX_MEMBER_QUANTITY 42
#define MAX_MEMBER_PER_TIME_UNIT 40
#define MIN_INTERDEPENDENT_TASK_NR 8
#define MAX_INTERDEPENDENT_TASK_NR 26
#define NR_OF_MEM_IN_INTERDEPENDENT_TASKS 37
#define Z2 0,9
#define Z3 0,3

// Start
{
    int i, j, current_best, global_best = 0;
    short flag = 0;
    time_t t;
    unsigned srd = (unsigned) time(&t);
    srand(srd);

    // Generation of Population
    for (j = 0; j < MAX_GN; j++)
    {
        for (int z = 0; z < 100; z++)
         {
            generate_pop(global_best, flag);
            i = 0;
            while (!pop[i].feasible && i < MAX_POP_SIZE) i++;
            if (i != MAX_POP_SIZE) break;
         };

        flag = 1;
        do
        {
            feasibility_check ();
            pop_fitness();
            find_best();
            show_population();
            reproduction();
            feasibility_check ();
            pop_fitness();

            //finds new best chromosome
            i = 1;
            while (!pop[i].feasible && i < MAX_POP_SIZE) i++;
            if ( i != MAX_POP_SIZE)
            {
                current_best = i;
                while (i < MAX_POP_SIZE)
                {
                 if (pop[i].quality >= pop[current_best].quality && pop[i].feasible) current_best = i;
                 i++;
                }
            }
            else current_best = (rand() % MAX_POP_SIZE);
            show_population2();
```

```
 if (pop[best_chrom].quality == pop[current_best].quality) break;
      }
 while (pop[best_chrom].quality * Z2 < pop[current_best].quality);

     if (pop[best_chrom].quality > pop[current_best].quality) global_best = best_chrom;
     else global_best = current_best;

    if (best_possible() * Z3 <= pop[global_best].quality || best_possible() == pop[global_best].quality) break;
    if (pop[global_best].quality <= pop[best_chrom].quality) break;
}

if (pop[global_best].feasible == 0)
{
    F1Book1->Sheet = 2;
    F1Book1->TextRC [1][4] = "Didn't find any feasible chromosome";
}
else show_best (global_best);


//generation of population
void TForm1::generate_pop(int global_best, short flag)
{
    int i, j, k, n, pass;
    int department;
    short time_unit;
    double min_mkr, mkr;

    if (flag)
    {
        for(j = 0; j < MAX_TASK_NUMBER; j++)
        {
            F1Book1->Sheet = 4;
            n = get_quantity(j+2);
            for(k = 0; k < n; k++)
            {
                pop[0].chrom[j][k] = pop[global_best].chrom[j][k];
            }
        }

    }

    for(i = flag; i < MAX_POP_SIZE; i++)
    {
        pop[i].quality = 0;
        for(j = 0; j < MAX_TASK_NUMBER; j++)
        {
            F1Book1->Sheet = 4;
            n = get_quantity(j+2);
            for(k = 0; k < n; k++)
            {
                F1Book1->Sheet = 2;
                time_unit = F1Book1->NumberRC [j+2][2];
                //while chosen person is busy or it is not a person that has knowledge
                do {
                    pop[i].chrom[j][k] = (rand() % MAX_MEMBER_QUANTITY) + 1;

                    F1Book1->Sheet = 4;
                    department = F1Book1->NumberRC [j+2][12+k];
                    min_mkr = F1Book1->NumberRC [j+2][16];
```

```
                        F1Book1->Sheet = 5;
                        mkr = F1Book1->NumberRC [pop[i].chrom[j][k] + 1][department + 1];
                        if (mkr < min_mkr) pass = 0;
                        else pass = 1;

                        F1Book1->Sheet = 3;
                    } while (F1Book1->NumberRC [pop[i].chrom[j][k] + 1] [time_unit + 1] != 0 || pass == 0);
                    F1Book1->Sheet = 4;
                }
            }
        }
}

// feasibility check of chromosomes
void TForm1::phisibility_check(void)
{
        int i, j, k, l, n;
        int department;
        short time_unit, old_time_unit;
        short buf[MAX_MEMBER_PER_TIME_UNIT];
        short index;
        double min_mkr, mkr;
        unsigned short good_chrom[MAX_TASK_NUMBER][MAX_MEMBER_QUANTITY_PER_TASK] = {if
wanted};

        for(i = 0; i < MAX_POP_SIZE; i++)
        {
            index = 0;
            old_time_unit=0;
            for(j = 0; j < MAX_TASK_NUMBER; j++)
            {
                pop[i].feasible = 1;
                F1Book1->Sheet = 4;
                n = get_quantity(j+2);
                for(k = 0; k < n; k++)
                {
                    F1Book1->Sheet = 2;
                    time_unit = F1Book1->NumberRC [j+2][2];
                    if (time_unit != old_time_unit) index = 0;
                    old_time_unit = time_unit;
                    buf[index++] = pop[i].chrom[j][k];

                    // member schedule check
                    F1Book1->Sheet = 3;
                    if (F1Book1->NumberRC [pop[i].chrom[j][k] + 1] [time_unit + 1] != 0)
                    {
                        pop[i].feasible = 0;
                        break;
                    }


                    // member knowledge check
                    F1Book1->Sheet = 4;
                    department = F1Book1->NumberRC [j+2][12+k];
                    min_mkr = F1Book1->NumberRC [j+2][16];
                    F1Book1->Sheet = 5;
                    mkr = F1Book1->NumberRC [pop[i].chrom[j][k] + 1][department + 1];
                    if (min_mkr > mkr)
                    {
```

```
                    pop[i].feasible = 0;
                    break;
                }


                // member workload check
                for (l = 0; l < index - 1; l++)
                    if (buf[l] == pop[i].chrom[j][k])
                    {
                        pop[i].phisible = 0;
                        break;
                    }
            }
            if (!pop[i].feasible) break;
        }
    }
    //checking if any chromosome is feasible, if not introduce one on position 0
    for (i = 0; i < MAX_POP_SIZE; i++)
        if (pop[i].feasible) break;
    if (i == MAX_POP_SIZE)
    {
        for(i = 0; i < MAX_POP_SIZE; i++)
            for(j = 0; j < MAX_TASK_NUMBER; j++)
            {
                F1Book1->Sheet = 4;
                n = get_quantity(j+2);
                for(k = 0; k < n; k++)
                    pop[0].chrom[j][k] = good_chrom[j][k];
            }
        feasibility_check();
        pop[0].feasible = 1;
    }
}

// power
double TForm1::power( int n )
{
    double s = 1.;
    while( n > 1 ) s *= n--;
    return s;
}


//fitness function evaluation
void TForm1::pop_fitness(void)
{
    int i,j,k,m,n;
    short department;
    double mkr, min_mkr, wrr = 0;
    double quality;
    short wrr_row, wrr_coll;

    for(i = 0; i < MAX_POP_SIZE; i++)
    {
        if (!pop[i].feasiblee) continue;
        quality = 0;
        for(j = 0; j < MAX_TASK_NUMBER; j++)
        {
            F1Book1->Sheet = 4;
```

```
                n = get_quantity(j+2);
                for(k = 0; k < n; k++)
                {
                    F1Book1->Sheet = 4;
                    department = F1Book1->NumberRC [j+2][12+k];
                    min_mkr = F1Book1->NumberRC [j+2][16];
                    F1Book1->Sheet = 5;
                    mkr = F1Book1->NumberRC [pop[i].chrom[j][k] + 1][department + 1];
                    quality += mkr - min_mkr;

        if (j >= MIN_INTERDEPENDENT_TASK_NR - 1 && j <= MAX_INTERDEPENDENT_TASK_NR - 1)
                    {
                        F1Book1->Sheet = 7;
                        quality += F1Book1->NumberRC [pop[i].chrom[j][k] + 1][2];
                    }
                }
            }
            for(j = MIN_INTERDEPENDENT_TASK_NR - 1; j < MAX_INTERDEPENDENT_TASK_NR; j++)
            {
                F1Book1->Sheet = 4;
                n = get_quantity(j+2);
                for(k = 0; k < n; k++)
                {
                    F1Book1->Sheet = 6;
                    wrr_row = F1Book1->NumberRC [pop[i].chrom[j][k]+1][3];

                    for (int x = j; x < MAX_INTERDEPENDENT_TASK_NR; x++)
                    {
                        F1Book1->Sheet = 4;
                        m = get_quantity(x+2);
                        for(int y = k + 1; y < m; y++)
                        {
                            F1Book1->Sheet = 6;
                            wrr_coll = F1Book1->NumberRC [pop[i].chrom[x][y]+1][4];
                            wrr += F1Book1->NumberRC [wrr_row] [wrr_coll];
                        }
                    }
                }
            }
            quality += wrr;
            pop[i].quality = quality;
            wrr = 0;
        }
}

// finds best chromosome
void TForm1::find_best(void)
{
    int i = 0;
    while (!pop[i].phisible && i < MAX_POP_SIZE) i++;
    if (i == MAX_POP_SIZE) best_chrom = (rand() % MAX_POP_SIZE);
    else best_chrom = i;
    while (i < MAX_POP_SIZE)
    {
        if (pop[i].quality >= pop[best_chrom].quality && pop[i].feasible) best_chrom = i;
        i++;
    }
}
```

```
// generate random number from range (0,1)
inline double TForm1::rand_01()
{
    return (double)((rand()<<30 | rand()<<15 | rand())%0xffffffff)/(double)(0xffffffff);
}

// reproduction process
void TForm1::reproduction(void)
{
    population_type new_pop[MAX_POP_SIZE]; // population
    int i,i1,i2,j,k,n;
    time_t t;
    unsigned srd = (unsigned) time(&t);

    srand(srd);
    for (i = 0; i < MAX_POP_SIZE; i++)
    {
        //do i1 = rand() % MAX_POP_SIZE; while (pop[i1].feasible == 0);
        //do i2 = rand() % MAX_POP_SIZE; while (i1 == i2 && pop[i2].feasible == 0);
        i1 = rand() % MAX_POP_SIZE;
        do i2 = rand() % MAX_POP_SIZE; while (i1 == i2);

        for(j = 0; j < MAX_TASK_NUMBER; j++)
        {
            F1Book1->Sheet = 4;
            n = get_quantity(j+2);
            for(k = 0; k < n; k++)
            {
                if (i == 0) new_pop[i].chrom[j][k] = pop[best_chrom].chrom[j][k];
                else
                {
                    if (rand_01 < 0.5)
                        new_pop[i].chrom[j][k] = pop[i1].chrom[j][k];
                    else
                        new_pop[i].chrom[j][k] = pop[i2].chrom[j][k];
                }
            }
        }
    }
    for(i = 0; i < MAX_POP_SIZE; i++)
    {
        for(j = 0; j < MAX_TASK_NUMBER; j++)
        {
            F1Book1->Sheet = 4;
            n = get_quantity(j+2);
            for(k = 0; k < n; k++)
            {
                pop[i].chrom[j][k] = new_pop[i].chrom[j][k];
            }
        }
    }
}

//returns best possible quality of chromosome
double TForm1::best_possible (void)
{
    int j, k, n;
```

```
	double result = 0;

	for(j = 0; j < MAX_TASK_NUMBER; j++)
	{
		F1Book1->Sheet = 4;
		n = get_quantity(j+2);
		for (k = 0; k < n; k++)
		{
			F1Book1->Sheet = 4;
			result += 1 - F1Book1->NumberRC [j+2][16];
		}
	}
result += NR_OF_MEM_IN_INTERDEPENDENT_TASKS;
result+=power(NR_OF_MEM_IN_INTERDEPENDENT_TASKS)/(2*silnia(NR_OF_MEM_IN_INTERDEPEN
DENT_TASKS-2));
	F1Book1->Sheet = 1;
	F1Book1->TextRC [1][19] = "Best possible";
	F1Book1->NumberRC [2][19] = result;
	return result;
}

// shows the best chromosome
void TForm1::show_best(short best)
{
	int j,k,n;

	F1Book1->Sheet = 2;
	F1Book1->TextRC [1][4] = "Best chromosome";
	F1Book1->NumberRC [1][6] = best+1;
	for(j = 0; j < MAX_TASK_NUMBER; j++)
	{
		F1Book1->Sheet = 4;
		n = get_quantity(j+2);
		for (k = 0; k < n; k++)
		{
			F1Book1->Sheet = 2;
			F1Book1->NumberRC [j+2][k+4] = pop[best].chrom[j][k];
		}
	}
}
```

APPENDIX F

MATHEMATICAL MODEL

Mathematical Model

This appendix presents a mathematical model for assigning the right team member to the right task at the right time. The model is related to typical employee scheduling models. The appendix is organized into three sections: Notation, Objective Functions, and Constraints. For ease of notation, the model is described in terms of a single period.

Notation

Sets:

T = number of tasks in the entire project

N = number of interdependent task groups

S_n = set of tasks in the interdependent task group n (n = 1 to N)

J = total number of departments

P = total number of members

Decision variable:

x_{pt} = 1 if member p is assigned to task t, 0 otherwise

Parameters:

K'_{jt} = multifunctional knowledge rating required by task t for department j (0 to 1, normalized)

K_{pj} = multifunctional knowledge rating of member p for department j (0 to 1, normalized)

T_p = teamwork capability rating of member p (0 to 1, normalized)

R_{pq} = working relationship rating between members p and q (0 to 1, normalized)

J_t = number of members required by task t

Objective Function:

The objective function $W_{t\text{-}m}$ has three components that must be maximized: multifunctional knowledge rating, teamwork capability rating and working relationship rating. For the current version of the model, each objective is equally weighted.

1. Multifunctional knowledge rating: $\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{j=1}^{J}(K_{pj} \bullet x_{pt})$

2. Teamwork capability rating: $\sum_{n=1}^{N}\sum_{t\in S_n}\sum_{p=1}^{P}(T_p \bullet x_{pt})$

3. Working relationship rating: $\sum_{n=1}^{N}\sum_{t\in S_n}\sum_{p=1}^{P}\sum_{q=p+1}^{P}(R_{pq} \bullet x_{pt} \bullet x_{qt})$

Thus, the overall objective is as follows:

$$\max W_{t\text{-}m} = \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{j=1}^{J}(K_{pj} \bullet x_{pt}) + \sum_{n=1}^{N}\sum_{t\in S_n}\sum_{p=1}^{P}(T_p \bullet x_{pt}) + \sum_{n=1}^{N}\sum_{t\in S_n}\sum_{p=1}^{P}\sum_{q=p+1}^{P}(R_{pq} \bullet x_{pt} \bullet x_{qt}) \quad (1)$$

Note that the third component (working relationship rating) is quadratic.

Constraints

$$K_{pj} \bullet x_{pt} \geq K'_{tj} \qquad t = 1 \text{ to } T,\ j = 1 \text{ to } J,\ p = 1 \text{ to } P \qquad (2)$$

$$\sum_{p=1}^{P} x_{pt} = J_t \qquad t = 1 \text{ to } T \qquad (3)$$

$$x_{pt} \in \{0,1\} \qquad p = 1 \text{ to } P,\ t = 1 \text{ to } T \qquad (4)$$