A Novel Congruent Organizational Design Methodology Using Group Technology and a Nested Genetic Algorithm [1]

Student Paper Submission
(Suggested Track: Modeling and Simulation)

Feili Yu
E-mail: yu02001@engr.uconn.edu

Georgiy M. Levchuk
Aptima, Inc.
12 Gill Street, Suite 1400,
Woburn, MA 01801
Phone: 781-935-3966 ext. 267
Fax: 781-935-4385
E-mail: georgiy@aptima.com

Candra Meirina
E-mail: meirina@engr.uconn.edu

Sui Ruan
E-mail: sruan@engr.uconn.edu

Krishna Pattipati
University of Connecticut, Dept. of Electrical and Computer Engineering
371 Fairfield Road, Unit 1157
Storrs, CT 06269-1157
Fax: 860-486-5585
Phone: 860-486-2890
E-mail: Krishna@engr.uconn.edu

[1] This work was supported by the Office of Naval Research under contract # N00014-00-1-0101.

A Novel Congruent Organizational Design Methodology Using Group Technology and a Nested Genetic Algorithm *

Feili Yu[1], Georgiy M. Levchuk[2], Candra Meirina[1], Sui Ruan[1] and Krishna R. Pattipati[1, 3]

[1]University of Connecticut, Dept. of Electrical and Computer Engineering, Storrs, CT 06269
[2]Aptima, Inc. 12 Gill Street, Suite 1400, Burlington, MA 01801
[3]Correspondence: krishna@engr.uconn.edu

ABSTRACT

A key concept in congruent organizational design is the so-called Strategic Grouping, which involves the aggregation of task functions, positions, and assets into units. Group technology has emerged as a manufacturing philosophy for improving productivity in batch production systems, while retaining the flexibility of a job shop production. In this paper, we propose a methodology to group tasks and assets into several clusters (DMs, command cells) which uses concepts from group technology and genetic algorithms to minimize the weighted total workload, measured in terms of intra-DM and inter-DM coordination workload. An outcome of strategic grouping is a near-optimal layout of the organization, i.e., the assignment of platforms to tasks and the patterns of coordination.

1. Introduction

Each organization, as a whole, displays a degree of fit or a degree of overall congruence with its environment. The greater the total degree of congruence, or fit, among the various components in an organization is, the more effective the organization will be [Nadler 1997]. In a military operation, an organization is said to be *congruent* with its mission, if its *structure* and *processes* are *matched* to the environmental parameters ([Levchuk 2003], [Burton 1998], and [Mackenzie 1986]). The "match" between an organization and a mission can assume a *performance-based* or *structure-based* concept. The performance-based congruence is a relative concept: to find the degree of congruence to a mission, the performance of an organization is compared to the performance of an *optimal* organization for executing the same mission.

Our previously developed mission modeling and three-phase organizational design methodology overcame the computational complexity of operationalizing organizational design by synthesizing a command structure via an iterative solution of a sequence of smaller and well-defined optimization problems [Levchuk 2002a,b]. Application of different optimization algorithms at different stages of the design process leads to an efficient matching between the mission structure and the structure of an organization and its resources/constraints, thereby obtaining an acceptable trade-off among multiple objectives and constraints, as well as between the computational complexity and solution efficiency (desired degree of sub-optimality).

Phase I of our 3-phase organizational design process [Levchuk 2002a, b] finds the optimized schedule and the concomitant platform-task allocation, and Phases II and III identify an organization suitable for this schedule. The schedule obtained in Phase I utilizes a heuristic list scheduling procedure, which does not account for the workload of inter-DM coordination (DMs are not yet defined in this phase). Thus, allocation of platforms to DMs in Phase II can potentially lead to a high degree of sub-optimality, since it is based on the previously obtained platform-task allocation. This is due to the fact that the optimization problem has been decomposed into

* This work was supported by the Office of Naval Research under contract # N00014-00-1-0101.

several sub-problems, which are not separable. Furthermore, the 3-phase design process does not take into account the task execution accuracy; it assumes that all the task requirements can be fully satisfied.

The aforementioned problems can be overcome by first clustering tasks into groups that have similar processing requirements, and then assigning platforms to the corresponding task groups, such that the resource requirements of tasks in each group are satisfied. In the resource allocation phase, we assign platforms to tasks to minimize platform transfer delays and to maximize the task processing accuracy. Tasks that can not be processed by a single DM will be coordinated with other DM nodes based on the same objective. The clustering of tasks and platforms, as well as resource allocation, using such a methodology can significantly improve the organizational performance by reducing the local nature of optimization in Phases I and II of our 3-phase design process.

The rest of the paper is organized as follows. Section 2 outlines the Group Technology (*GT*) algorithms, as well as the relationship between the concept of *GT* and that of congruence. Section 3 formulates the organizational design problem. The solution approach, using *GT* and a nested genetic algorithm (*NGA*), is proposed in Section 4. Section 5 presents an example of congruent organizational design. The paper concludes with a summary and future research directions in Section 6.

2. Group Technology:

2.1 Group Technology Overview

The emerging threats of the 21st century are characterized by high variability in missions and the need for focused and rapid deployment of resources to counter the threats. The design of flexible and responsive organizations to meet the emerging threats is a scientific challenge. Group technology (*GT*) is one of "recognizing and exploiting similarities in three distinct ways: (1) by performing similar operations together, (2) by standardizing similar tasks, and (3) by efficiently storing and retrieving information about recurring problems" [Hyer 1989]. Group technology can be operationalized by dividing a C^2 system into several manageable subsystems or cells, responsible for managing tasks, assets (platforms), and information flow. The advantages of introducing group technology into C^2 systems are:

- Improved speed of command,
- Reduced task latencies (execution delay),
- Reduced resource requirements,
- Reduced mission inefficiencies,
- Reduced synchronization delays,
- Reduced response time, and
- Improved flexibility.

2.2 Grouping Algorithms

Grouping analysis is concerned with clustering of objects into homogeneous groups (cells) based on the object features [Kusiak 1990]. Several categories of clustering algorithms are possible and are discussed below.

A. Matrix-based clustering

The platform-task dependency is represented by a matrix $[a(i, j)]$ consisting of '0, 1' entries, where an entry 1 (0) indicates that platform i is used (not used) for processing task j. Most matrix-based algorithms, such as Bond Energy Algorithm (BEA) [McCormick 1972] and Rank Order Clustering (ROC) [King 1980], resort to transferring the initial matrix into a structured matrix, such as a block-diagonal one. The clustering of a binary incidence matrix may or may not result in separable clusters. Those tasks that cannot be clustered into any group are called exceptional tasks. These are the major determinants of *clustering efficiency*, which is defined as the ratio of the number of separable tasks (i.e., the difference between the total number of tasks and the exceptional tasks) to the total number of tasks.

B. Hierarchical clustering

A hierarchical clustering method is a procedure for transforming a proximity (distance or similarity) matrix into a sequence of nested partitions [Joines 1996]. A hierarchy of clusters or partitions is produced in terms of similarity or distance; the proximity measures can incorporate platform-task dependency data, rather than just the binary incidence matrix. The hierarchical clustering methods involve two stages. The first stage calculates similarity coefficients (or distances) between pairs of tasks/platforms. Then the second stage clusters tasks/platforms into groups based on a threshold. For simple missions with a few tasks and platforms, an analyst can approximate such a threshold. However, when the mission is too complex to estimate an initial value for the threshold, hierarchical clustering becomes infeasible.

C. Graph-Theoretic Clustering

In the graph formulation, the incidence matrix $[a(i, j)]$ is represented by a graph. Three types of graphs are used: bipartite graph, transition graph, and boundary graph. Among these graphs, the bipartite structure is used most often. The tasks and platforms are assigned to two distinct sets; an edge between a (task, platform) pair represents that the task is processed by the concomitant platform.

D. Clustering based on Artificial Intelligence

Artificial neural networks (ANNs) and Fuzzy Logic have been applied to group technology. ANNs have emerged in recent years as a major means for pattern recognition, and it is this particular capability that has made ANNs a useful addition to the tools and techniques applicable for group technology and for the design of cellular systems [Suresh 2000]. The ANN involves supervised learning that performs pattern classification (task/platform cluster formation) from task-platform incidence matrix. Because clusters may overlap, fuzzy logic techniques, such as *Fuzzy c-mean Clustering,* have been applied to solve the grouping problem. When the input contains both analog and binary numbers, a method combining fuzzy logic and Adaptive Resonance Theory (ART), termed *Fuzzy-ART*, can be applied.

E. Evolutionary Clustering Methods

Genetic Algorithm (*GA*) and Simulated Annealing (*SA*) are two evolutionary clustering methods applied to group technology. Both of them are stochastic search algorithms that have been successfully used to solve combinatorial optimization problems. They have been widely used to formulate the cell formation problem of assigning tasks/platforms to different clusters. The stochastic nature of these two algorithms enables them to escape the local minima, and to offer promising solutions for large-scale problems. In Section 4.1, we will provide an overview of *GA* and apply it to solve the task/platform grouping problem in Section 4.2.

F. Mathematical programming methods

The cell formation problem can be formulated as a linear or nonlinear optimization problem with different objectives and constraints. McCormick *et al*. [McCormick 1972] formulated it as a quadratic assignment problem. Steudal and Ballakru [Steudal 1987] developed a dynamic programming approach for the grouping problem. Nagi et al. [Nagi 1990] considered routing and capacity allocation formulation of the cell formation problem, and employed a branch and bound method to solve the problem. Kusiak [Kusiak 1987] developed the *p-median* model to assign *n* tasks into *p* cells.

2.3 The *GT* and congruent organizational design

The philosophy behind the group technology is the concept of a congruent organization. D. Nadler and M. Tushman [Nadler 1997] proposed a *Congruence Model of Organizational Behavior*. This model views an organization as a collection of interacting components. These components, including the environment (e.g. mission, tasks, etc.), formal organization (e.g. groups, coordination structures, command structures, etc.), informal organization (e.g. strategy, management, culture, etc.), and agent (e.g. resources structures, assets, capabilities, etc.) should "fit" with each other. They also proposed a three stage of congruent organizational design process: *strategic grouping, structural linking,* and *system/process design.*

A. Strategic Grouping

Grouping involves the aggregation of task functions, positions, and individuals into units. Grouping explicitly places some tasks, resources, and people together in the same units, and implicitly separates some tasks and resources. Resources belonging to the same group are able to be allocated and scheduled more efficiently. People (one type of resource) will become more skilled and specialized as they dedicate their efforts to a limited range of operations. It also influences the organization's information-processing capacity. However, information becomes easier to process within grouped boundaries ("internal coordination"), although it will also require more effort to communicate and cooperate between groups ("external coordination").

B. Structural Linking

By grouping, the communication and coordination become easier within the group. However, the barriers are built up between different groups, thereby making information sharing and cooperation more difficult. The design of structural linking is one of setting up mechanisms that facilitate the information and resource flows among groups in order to help each group process its tasks and achieve its objectives. Since the cross-boundary coordination is costly (in terms of delays), extensive cooperation will introduce additional "external workload" among groups; this will hinder the flow of information and resources. The key to successful organizational design is to discover a linking structure that optimally balances the needs for coordination.

C. System and Process Design

Systems and processes are designed with groupings and structures in mind to support the movement of information among groups. These can range from information, control, and reward systems to formal processes and meetings.

The group technology is based on the congruent model introduced earlier. It groups homogenous tasks and resources together in the same cells in order to facilitate task planning and execution. If we regard tasks as the information flow in an organization, then, within a group, the efficiency of information-processing is maximized. In addition, coordination among different groups is appropriately assigned, considering both the necessity of coordination for certain tasks, and the increased cost of such coordination due to information (or resource) flow "barriers" among different DM cells. In this paper, we group tasks and assets (platforms) into several cells (DM nodes) using concepts from group technology to minimize the weighted total workload, measured in terms of intra-DM and inter-DM workload.

3. Problem Formulation

3. 1 Motivation

A *task*, derived from mission decomposition, is an activity that entails the use of relevant resources, and is carried out by one or more decision makers (DMs) to accomplish the mission objectives. The DM is an entity with information-processing, decision-making, and operational capabilities that can control the necessary resources to execute tasks. A DM also communicates with other DMs, and cooperates on task execution by sharing his resources. The *resources* are carried by platforms or assets with given resource capabilities, ranges of operation and velocities. The *organization* consists of a set of DMs, the assignment of platforms to DMs, and the coordination structure among DMs.

The process of mission execution is as follows. A set of tasks with specified resource requirements, locations, and precedence relations need to be processed by the organization. The tasks are assigned to DMs based on the fit between the resource requirements of tasks and the resource capabilities of DMs. The assigned DMs select and send their platforms to the locations where tasks appear in order to execute them with minimum lead time and maximum accuracy. In the situation, wherein a DM assigned to a task must utilize the assets/platforms from another DM, they must coordinate to synchronize the operations of their platforms (e.g., arrival time of platforms at the task location). Only when all the platforms needed to process a task have arrived, the task execution begins. Therefore, the delays in task execution are primarily due to synchronization. In order to minimize the overall task completion time, the synchronization delays should be minimized. In addition, the task execution accuracy should be maximized. We note that minimizing the inter-DM coordination delay ("between group delay") outweighs the intra-DM coordination delay ("within group delay"), since there is a "barrier" between any two DM cells. However, there are always some exceptional tasks that need to be processed by more than one DM. Due to these exceptional tasks, the inter-DM coordination delays are inevitable. A tradeoff between internal and external coordination workload is a key aspect of our design approach, and is formalized next.

3.2 Mathematical Formulation

The design parameters are:

- Number of DMs, M
- Task-DM assignment matrix, TD
- Platform-DM assignment matrix, PD
- Task-platform assignment matrix, TP

The ranges of indices are as follows: $i = 1, \cdots, N_t$; $j = 1, \cdots, N_p$; $m = 1, \cdots, M$, where, N_t and N_p are the number of input tasks and the number of available platforms, respectively. Also, assume that there are L resource types (e.g., weapons, sensors) distributed among the N_p platforms. The entries of the above incidence matrices are defined as follows:

$$TD(i,m) = \begin{cases} 1, & \text{if task } i \text{ is assigned to DM } m \\ 0, & \text{otherwise} \end{cases}$$

$$PD(j,m) = \begin{cases} 1, & \text{if platform } j \text{ is assigned to DM } m \\ 0, & \text{otherwise} \end{cases}$$

$$TP(i,j) = \begin{cases} 1, & \text{if platform } j \text{ is assigned to Task } i \\ 0, & \text{otherwise} \end{cases}$$

The input parameters are

- The total number of resource types, L
- The task-resource requirement matrix

  $$R_T = [r_T(i,l)] \quad (i = 1, \cdots, N_t; l = 1, \cdots, L)$$

- The platform-resource capability matrix

$$R_P = [r_P(j,l)] \quad (j = 1,\cdots,N_p; l = 1,\cdots,L)$$

- The vector of velocities of platforms

$$\underline{v} = \{v(j) \mid j = 1,\cdots,N_p\}$$

- The vector of task locations

$$(\underline{x}_T, \underline{y}_T) = \{(x_T(i), y_T(i)) \mid i = 1,\cdots,N_t\}$$

The aggregated workload of each DM, which takes into account both the intra-DM and the inter-DM coordination workloads, is given by:

$$W(m) = \alpha W_{Intra}(m) + (1-\alpha) W_{Inter}(m), \qquad (3.1)$$

where $W_{Intra}(m)$ is the intra-DM coordination workload of DM m, $W_{Inter}(m)$ is the inter-DM coordination workload, and α is the weight assigned to the intra-DM workload. The value of α should be less than 0.5, because the inter-DM workload is more costly than the intra-DM workload.

Although our objective is to minimize the total aggregated workload of the organization, minimizing $W(m)$ alone, when α is small, will lead to smaller number of DMs in the organization with unbalanced workloads. Consequently, we seek to minimize the root mean-square (RMS) value of the aggregated workload, which is given by:

$$\text{Min: } WRMS = \sqrt{\frac{1}{M}\sum_{m=1}^{M} W^2(m)} \qquad (3.2)$$

subject to

$$\sum_{m=1}^{M} TD(i,m) \geq 1 \quad \text{for } i = 1,2,\cdots,N_t \qquad (3.3)$$

$$\sum_{m=1}^{M} PD(j,m) = 1 \quad \text{for } j = 1,2,\cdots,N_p \qquad (3.4)$$

The objective function in Eq. (3.2) minimizes both the mean and variance of workload across the team [Levchuk 2003]. The first constraint in Eq. (3.3) ensures that at least one task is allocated to a DM. The second constraint in Eq. (3.4) implies that each platform can only be allocated to one DM. Although the problem is not separable, for ease of solution, we decompose this optimization problem into two sub-problems: intra-DM workload minimization, and inter-DM workload minimization.

A. Minimize Intra-DM Workload

For a given task/platform grouping, the tasks allocated to a particular DM, *m*, are given by

$$\underline{T}_m = \{T_i \mid i \in \{TD(i,m) == 1, i = 1,\cdots,N_t\}\}$$

The platforms allocated to DM *m* are given by

$$\underline{P}_m = \{P_j \mid j \in \{PD(j,m) == 1, j = 1,\cdots,N_p\}\}$$

A.1 Platform Transfer Delay $t(m,i)$ for Task T_i

Our model assumes that the center of the DM location, $[x_D(m), y_D(m)]$ is the center of all tasks allocated to DM *m,* that is,

$$x_D(m) = \frac{\sum_{i=1}^{N_t} TD(i,m) x_T(i)}{\sum_{i=1}^{N_t} TD(i,m)}; \quad y_D(m) = \frac{\sum_{i=1}^{N_t} TD(i,m) y_T(i)}{\sum_{i=1}^{N_t} TD(i,m)} \qquad (3.5)$$

The model also assumes that all the platforms allocated to this DM are initially located at the center of the DM location, termed the base. When a platform executes a task, it travels from its initial position (base) to the task location, and then travels back to its base after the task is executed. Consequently, the distance between the platform and the task is the distance between the base and the task, defined as

$$d(i,j) = \sqrt{(x_T(i) - x_D(m))^2 + (y_T(i) - y_D(m))^2} \quad \text{for } \forall P_j \in \underline{P}_m \qquad (3.6)$$

Thus, the transfer delay for executing task T_i by DM *m* is given by

$$t(m,i) = \sum_{P_j \in \underline{P}_m} TP(i,j) \frac{d(i,j)}{v(j)} \qquad (3.7)$$

A.2 Task Accuracy

We adopted the concept of task accuracy from [Levchuk 2003], in which the task accuracy for a task T_i executed by DM *m* is defined as

$$A_{Intra}(m,i) = \left(\frac{1}{\tilde{L}} \sum_{l=1}^{L} a(i,l) \cdot TD(i,m) \right), \qquad (3.8)$$

where $a(i,l) = \dfrac{\tilde{r}_T(i,l)}{r_T(i,l)}$, $\tilde{L}$ is the number of resource requirements of task T_i over all types, *i.e.,* $\tilde{L} = \sum_{l=1}^{L} r_T(i,l)$, and $\tilde{r}_T(i,l)$ is the number of resources of type *l* actually used to process task T_i :

$$\tilde{r}_T(i,l) = \min\left\{ r_T(i,l), \sum_{P_j \in \underline{P}_m} [TP(i,j) \cdot r_P(j,l)] \right\} \quad (3.9)$$

Here, $a(i,l)$ is task accuracy for T_i in terms of each resource type *l*. The task accuracy for T_i is the average task accuracy over all *L* resource types.

A.3 Task Accuracy Significance (TAS)

The task accuracy significance of task T_i is defined as

$$TAS_i = [A_{Intra}(m,i)]^{\rho_1} \quad (3.10)$$

where ρ_1 is the task accuracy significance index (*TASI)* for intra-DM workload. The smaller ρ_1 is, the less effect task accuracy has on the intra-DM workload. On the other hand, if ρ_1 is large, task accuracy contributes significantly to the intra-DM workload.

A.4 The Minimization of Intra-DM Workload

The objective function for this sub-problem is a separable problem (for each DM m, $m = 1,2,\cdots M$)

$$\text{Min: } W_{Intra}(m) = \sum_{T_i \in \underline{T}_m} \frac{t(m,i)}{TAS_i} = \sum_{T_i \in \underline{T}_m} \frac{t(m,i)}{[A_{Intra}(m,i)]^{\rho_1}}$$

$$\text{subject to: } TP(i,j) \in \{0,1\} \quad (3.11)$$

B. Minimize Inter-DM Workload

B.1 Candidate Platform Selection

Based on the minimization of intra-DM workload, we obtain a tentative platform-task assignment matrix, $TP = [TP(i,j)]$, and a task accuracy matrix for each resource type, $A = \{a(i,l) \mid i = 1,\cdots,N_t\ ; l = 1,\cdots,L\}$.

The unfinished task set is given by:

$$\hat{\underline{T}} = \{T_i \mid i \in \{i \mid 0 \le a(i,l) < 1 \ \&\ r_T(i,l) > 0, l = 1,\cdots,L\}\}$$

Consider a $T_i \in \hat{\underline{T}}$ that is allocated to DM *m*. Then, the candidate platforms that can process the task T_i is given by

$$\hat{\underline{P}}(i) = \{P_j \mid P_j \notin \underline{P}_m\ ; r_P(j,l) \cdot r_T(i,l) > 0\}$$

In other words, only those platforms that are from different DM cells and that have full or partial capability to process task T_i are selected as candidate platforms. The design variable is the final platform-task assignment matrix $TP = [TP(i,j)]$.

B.2 Platform Transfer Delay $\hat{t}(i)$ for Task T_i

Suppose $P_j \in \hat{\underline{P}}(i)$ is from a DM k, $k \neq m$. The distance between P_j and T_i is given by

$$\hat{d}(i,j) = \sqrt{(x_T(i) - x_D(k))^2 + (y_T(i) - y_D(k))^2}$$

$$\text{for } \forall P_j \in \hat{\underline{P}}(i) \quad (3.12)$$

Thus, the transfer delay for inter-DM task execution is given by

$$\hat{t}(i) = \sum_{P_j \in \hat{\underline{P}}(i)} TP(i,j) \frac{\hat{d}(i,j)}{v(j)}. \quad (3.13)$$

- Task accuracy

Due to inter-DM coordination, the task accuracy of task T_i ($T_i \in \hat{\underline{T}}$) is updated as follows:

$$\hat{A}_{Inter}(i) = \frac{1}{\tilde{L}} \left(\sum_{l=1}^{L} \hat{a}(i,l) \right) \quad (3.14)$$

where $\hat{a}(i,l) = \dfrac{\hat{r}_T(i,l)}{r_T(i,l)}$, where $\hat{r}_T(i,l)$ is given by

$$\hat{r}_T(i,l) = \min\{r_T(i,l), \sum_{P_j \in \hat{\underline{P}}(i)} TP(i,j) r_P(j,l)\} \quad (3.15)$$

B.3 Task Accuracy Significance (TAS)

The task accuracy significance of task T_i for inter-DM coordination is defined as

$$TAS_i = [\hat{A}_{Inter}(i)]^{\rho_2} \quad (3.16)$$

where ρ_2 is the task accuracy significance index (*TASI)* for inter-DM workload.

B.4 Minimization of Inter-DM Workload

The sub-problem of minimizing the inter-DM workload W_{Inter} is given by

$$\text{Min: } W_{Inter} = \sum_{m=1}^{M} W_{Inter}(m) = \sum_{T_i \in \hat{T}} \frac{\hat{t}(i)}{[\hat{A}_{Inter}(i)]^{\rho_2}} \quad (3.17)$$

subject to $TP(i, j) \in \{0,1\}$.

Although these two sub-problems of minimizing the intra-DM workload and inter-DM workload are not separable, we can approximately optimize the objective function in Eq. (3.2) by sequentially minimizing the two sub-problems. Intra-DM sub-problem in Eq. (3.11) is minimized first and then, conditioned on the tentative results from this sub-problem, the inter-DM sub-problem in Eq. (3.17) is minimized.

4. Solution Approach—Nested *GA* (*NGA*)

4.1 Overview of *GA*

Genetic algorithms utilize stochastic search techniques based on the mechanism of natural selection and genetics. Various authors have applied genetic algorithms to the solution of cell formation problem. J.A. Joines *et al* [Joines 1996] developed a genetic algorithm to solve integer programming formulations of the cell design problem. The algorithm was tested on seventeen data sets from the literature and *was able to find as good solutions as, if not better than, those in the literature*. M. Kazerooni *et al* [Kazerooni 1995] used genetic algorithm to solve the cell formation problem based on the machine chain similarity coefficient matrix. [Gupta 1995] proposed a genetic algorithm to minimize the inter-cell and intra-cell moves in cellular manufacturing. [Gravel 1998] developed an efficient double-loop genetic algorithm to solve the cell formation problem with multiple routings. All these efforts have shown that the genetic algorithms are one of the most successful approaches for solving clustering problems with various objectives and constraints.

4.2 Nested GA

We employ a nested *GA* to solve the task/platform grouping problem associated with the platform-task assignment problem. The nested *GA* consists of two loops: the outer-loop *GA* (*OLG*) and the inner-loop *GA* (*ILG*).

A. Outer-loop GA (OLG)

The outer-loop *GA* seeks to find an optimal or near-optimal task/platform grouping, such that both the intra-DM and inter-DM workloads are minimized. The algorithm begins in the outer-loop with a randomly generated initial population of task/platform groupings. For each of these groupings, the inner-loop *GA* (*ILG*) determines the task-platform assignment matrix that minimizes the weighted workload. After the fitness values corresponding to the groupings are fed back from the inner-loop *GA* (*ILG)*, the *OLG* employs the genetic operators until the termination criteria are satisfied. The chromosome representation of task/platform grouping and parameter selection for the *OLG* are as follows.

A.1 Chromosome Representation

A chromosome representation scheme is determined by the structure of problem. Good representation can greatly improve the performance of *GA*. The individual or chromosome is made up of a sequence of genes from a certain alphabet. Binary and floating number representations are used primarily for numerical optimization. For the task/platform grouping problem, which is basically a combinatorial optimization problem, both binary and floating number representations are not efficient, since there is too much redundancy in the search space. Specifically, for the task/platform grouping, an integer alphabet $\{1,2,\cdots,M\}$ is employed, where M is the maximum number of DMs in the organization. The larger the value of *M*, the larger is the search space. In the representation of a chromosome, the task and platform should be considered simultaneously. Therefore, the representation is given by a $1\times(N_p + N_t)$ vector, where the first N_p genes represent platforms, and the last N_t genes represent tasks:

$$\underbrace{([P_1],[P_2],\cdots,[P_{N_p}]}_{Platforms},\underbrace{[T_1],[T_2],\cdots,[T_{N_t}])}_{Tasks}$$

Here $[P_j] \in \{1,2,..,M\}$ and $[T_i] \in \{1,2,..,M\}$ are the DMs assigned to platforms and tasks. Suppose we have three platforms (P_1, P_2, P_3), three tasks (T_1, T_2, T_3), and two DMs (i.e., two groups). The representation $(1,2,1,2,1,1)$ corresponds to the task/platform grouping (DM cells) shown in Figure 4.1

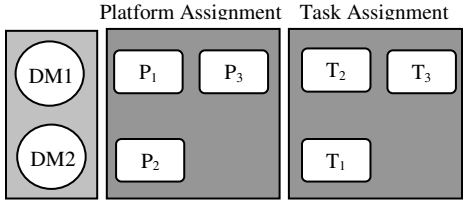

Figure 4.1 Illustration of the chromosome representation

A.2 Initialization of the Population

The initial population is randomly generated so that the value of each gene in a chromosome is between a lower bound and an upper bound. If some of the randomly generated chromosomes are not feasible (e.g., some of the DMs are not assigned any tasks or platforms), the fitness values of those individuals are set to minus infinity. Such solutions are not allowed to be selected for the new generation.

A.3 Selection and Evaluation Strategy

During each generation, chromosomes are evaluated using a measure of fitness. The following four major steps are included in the evaluation and selection phase: (i) convert chromosomes to the task-DM assignment matrix and the platform-DM assignment matrix, (ii) obtain the task-platform assignment by running the Inner-loop *GA* (*IGA*), (iii) calculate the objective function values for these tentative solutions, (iv) convert the objective function values into the corresponding fitness values.

Since minimizing the objective function in Eq. (3.2) is equivalent to maximizing the negative of the objective function, we adopted the *normalized Geometric* approach as the selection procedure because it can handle negative numbers. The chromosomes are ordered according to their fitness values. The probability of selecting a chromosome is based on its rank:

$$P\text{ [Selecting the } r^{\text{th}}\text{ chromosome]} = q'(1-q)^{r-1}, \quad (4.1)$$

where q is the probability of selecting the best individual, r is the rank of the individual, N is the population size, and $q' = \frac{q}{1-(1-q)^N}$.

A.4 Crossover and Mutation Operators

There are two basic considerations when designing crossover operators: (i) make fewer changes when crossing over so as to inherit the parents' features as much as possible, (ii) make more changes when crossing over so as to explore a new pattern of allocation and thereby enhance the search's ability to find a global optimum [Cheng 1999]. The crossover operators we used for the *OLG* are the *arithmetic crossover, two-point crossover (problem specific), and multi-point crossover (problem specific).* All of these operators are modified for integer representation of chromosomes used in our work.

Arithmetic crossover produces two complimentary linear combinations of parents. Suppose $0 \le a \le 1$; $p_1(i)$ and $p_2(i)$ are i^{th} gene of two parents. Then, the children are given by

$$ap_1(i)+bp_2(i) = \begin{cases} \lceil ap_1(i)+bp_2(i) \rceil & p_1(i) > p_2(i) \\ \lfloor ap_1(i)+bp_2(i) \rfloor & otherwise \end{cases}. \quad (4.2)$$

where $b = 1-a$; $\lceil x \rceil$ is the smallest integer greater than or equal to x (i.e., the ceiling function); $\lfloor x \rfloor$ is the largest integer less than or equal to x (i.e., the floor function).

Two-point crossover swaps sections of two parents according to the partitions of platforms and tasks.

Multi-point crossover is similar to the *Two-point crossover*. The difference is that both sections of platform and task are cut into two or more quarters according to one or more randomly generated numbers. The children are created by reassembling the different quarters. The swapping of quarters of two parents only happens to the corresponding parts, that is, the quarters in the platform part of parent p_1 can only swap with the quarters from the platform part of p_2.

Mutation operators alter one parent by changing one or more variables (genes) in some way, or by some random amount, to form an offspring [Joines 1996]. The mutation operators we employed in our algorithm are *uniform mutation, boundary mutation, non-uniform mutation,* and *multi-non-uniform mutation.*

Uniform mutation randomly selects one variable i, and sets it equal to a uniform random integer number $U(l_i, u_i)$, where (l_i, u_i) are the lower bound and upper bound of variable i, respectively.

Boundary mutation randomly sets variable i equal to either its lower bound or upper bound.

Non-uniform mutation randomly sets variable i equal to a non-uniform random integral number:

$$x_i' = \begin{cases} \lceil x_i + (u_i - x_i) f(G) \rceil & if\ r_1 < 0.5 \\ \lfloor x_i - (l_i + x_i) f(G) \rfloor & if\ r_1 \ge 0.5 \end{cases} \quad (4.3)$$

where

$$f(G) = \left(r_2 \left(1 - \frac{G}{G_{\max}}\right) \right)^b, \quad (4.4)$$

where r_1 and r_2 are uniform random numbers in [0, 1], G is the current generation, $G_{\max}$ is the maximum number of generations, and b is a shape parameter.

Multi-non-uniform mutation applies the non-uniform operator to all of the variables in the parent.

Some operators, such as the *two-point* or *multi-point* crossover operators inherit the parents' features more than other operators, such as *non-uniform mutation* and *multi-non-uniform mutation* operators. We applied a combination of the above mentioned operators for different numbers of times. Thus, by adjusting the repetition numbers, an analyst can control the genetic evolutionary process, so that the algorithm can obtain the best possible solution.

A.5 Termination Criterion

When the *GA* reaches a pre-specified number of generations, it terminates.

B. Inner-loop GA (ILG)

The *ILG* is responsible for (i) finding the platform-task assignment for a given grouping; (ii) evaluating intra-DM and inter-DM workloads, given the task/platform grouping provided by the *OLG*. The *ILG* is comprised of two stages. In the first stage, search for optimal or near optimal platform-task assignments for each DM cell is conducted to minimize the intra-DM coordination workload by employing *GA*. Because some tasks require coordination among DMs, in the second stage, the algorithm allocates multiple DMs and platforms to these tasks by minimizing the inter-DM coordination. The outcome of the *ILG* process is a platform-task assignment matrix. We name the two stages of *ILG* as inner-loop 1(*ILG 1*) and inner-loop 2 (*ILG 2*). Both of the inner-loops have similar chromosome representations and similar parameter settings for *GA*. The following discussion is suitable for both loops, unless specified otherwise.

B.1 Chromosome Representation

A platform can only process certain types of tasks according to its resource capability and task-resource requirements. Since the tasks and platforms are randomly grouped in the outer-loop, a pre-selection of candidate platforms for each task should be performed. After the pre-selection, each task has a certain number (may be zero) of candidate platforms related to it. Based on that, a binary chromosome contains $Q = \sum_{i=1}^{|\hat{T}|} | \underline{\hat{P}}(i) |$ genes. Each gene stands for the assignment status of each platform to each task. If its value is '1', it indicates that the platform has been allocated to this task. For example, for a three-task and three-platform instance, a chromosome taking the form [(1 0), (1 0 1), 1] indicates the task-platform assignment shown in Figure 4.2

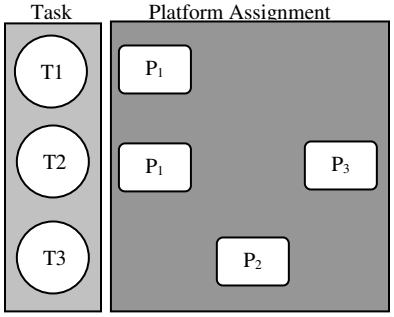

Figure 4.2 Illustration of the chromosome representation

B.2 Initialization of the Population

The initial population is generated by randomly assigning '0' or '1' to the genes in chromosome. However, some of them may not be feasible. For example, suppose there are three platforms and three tasks assigned to one DM cell and the first task T_1 can be processed by all of the platforms. If we obtain one chromosome of the form [(0 0 0) (1 0 0) (1 1 0)], then there is no platform allocated to T_1, which is not the assignment we expect. To avoid this, we modify it in the following way to ensure feasibility: (1) randomly select integers from the set {1, 2, 3}; (2) set the corresponding gene of T_1 to one. By doing this modification, all chromosomes will be feasible.

B.3 Selection and Evaluation Strategy

Similar to OLG, the *normalized Geometric* approach is employed for selection function.

B.4 Crossover and Mutation Operators

Since both *ILG* 1 and *ILG* 2 have binary chromosome representations of their solutions, the following two *GA* operators are applied to produce the offsprings.

Multi-point crossover randomly generates numbers for each task section of the chromosome. These random numbers split each task section into 2 quarters. The children are created by reassembling different quarters in corresponding task sections.

Inverse mutation flips the genes in the randomly selected task section into their complements ($0 \rightarrow 1$, or, $1 \rightarrow 0$). For example, the parent p,

$$p = \left[\cdots, \overbrace{1,\ 0,\ 1,\ 1,\ 0}^{\text{Platforms. for } T_i}, \cdots \right]$$

produces a child shown below:

$$c = \left[\cdots, \overbrace{0,1,\ 0,\ 0,\ 1}^{\text{Platforms. for } T_i}, \cdots \right]$$

B.5 Termination criterion

The *ILG* terminates when a pre-specified generation number is reached. Due to the binary representation of the solution, generally, the *ILG* needs less number of generations than the *OLG*.

The *NGA* is illustrated in Figure 4.3.

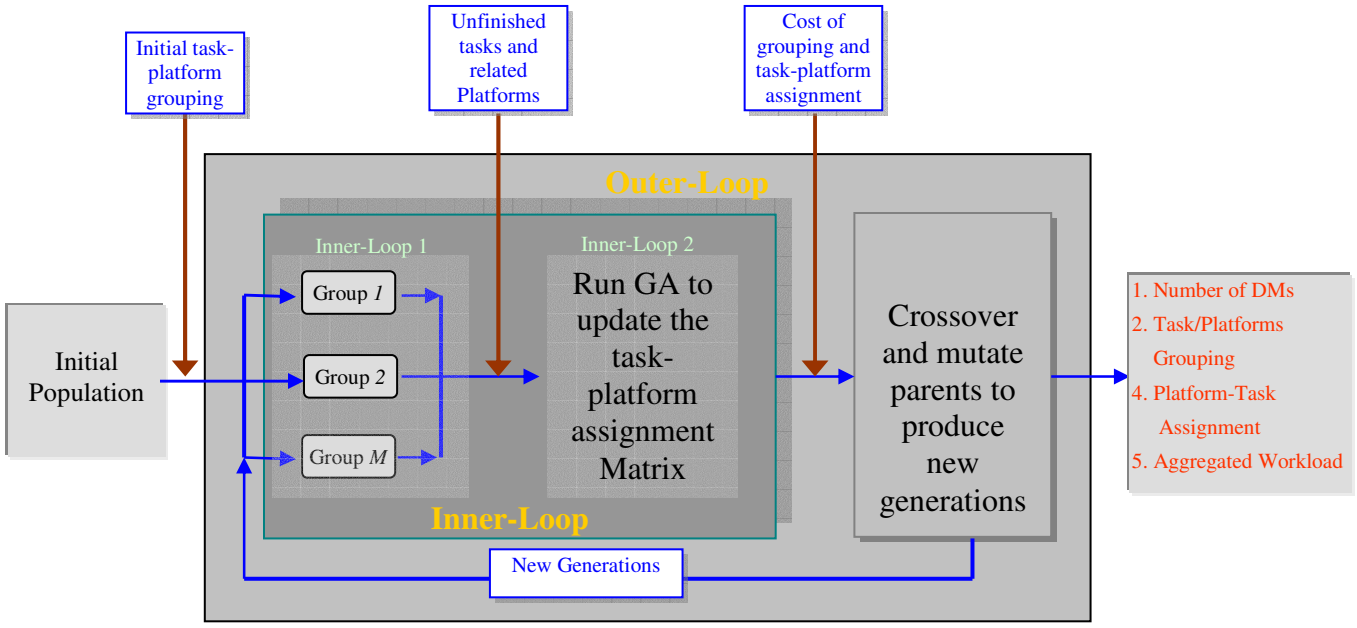

Figure 4.3 Nested *GA* for organizational design

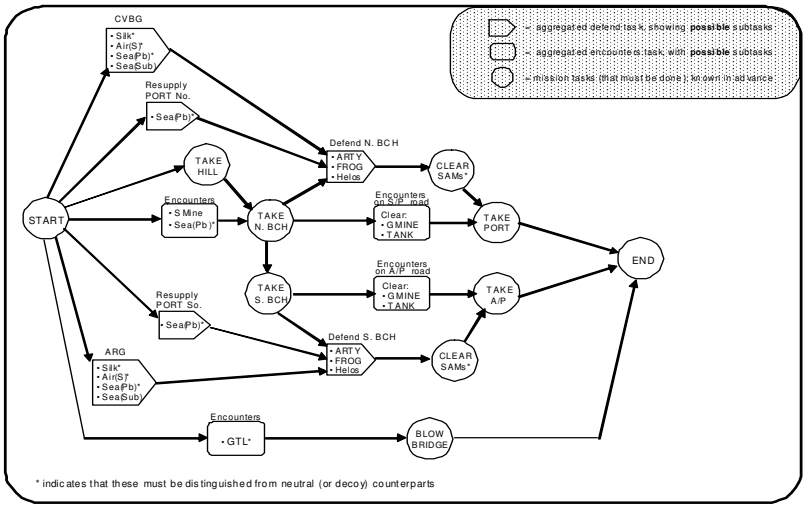

Figure 5.1 Example of task precedence graph

5. Numerical Example

5.1 Description of the example

In this section, we consider an illustrative example of designing an organization to execute a mission scenario consisting of 18 tasks. The mission includes capturing a seaport and an airport to allow for the introduction of follow-on forces. There are two suitable landing beaches designated: "North" and "South". The commander devises a plan for the mission that includes the completion of tasks shown in Figure 5.1. The following 8 resource requirements/capabilities are modeled: AAW, ASUW, ASW, GASLT, FIRE, ARM, MINE and DES. The reader is referred to [Levchuk2003] for details of the scenario and a description of resources and platforms.

Mission tasks, platforms, resource requirement vector for each task, resource capability vector for each platform, and other relevant parameters are presented in Figures 5.2 and 5.3. From Figures 5.2 and 5.3, we note that each task can be processed by a combination of platforms, as long as at least one of the required resources matches any one of the resource capabilities of platforms. The task-platform dependency matrix, illustrated in Table 5.1, is quite dense, that is, most of the entries in the matrix have value '1'. This implies that the search space for the *NGA* is potentially very large.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1	0	1
5	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1
15	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
16	1	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1
17	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 5.1 Original task-platform incidence matrix

Tasks	Locations	Resource Requirement Vector	Processing Time
1 CVBG	70 15	5 3 10 0 0 8 0 6	30
2 ARG	64 75	5 3 10 0 0 8 0 6	30
3 Resupply Port N	15 40	0 3 0 0 0 0 0 0	10
4 Resupply Port S	30 95	0 3 0 0 0 0 0 0	10
5 Encounters N&S	28 73	0 3 0 0 0 0 10 0	10
6 HILL	24 60	0 0 0 10 14 12 0 0	10
7 NORTH BEACH	28 73	0 0 0 10 14 12 0 0	10
8 SOUTH BEACH	28 83	0 0 0 10 14 12 0 0	10
9 Defend N. Beach	28 73	5 0 0 0 0 5 0 0	10
10 Defend S. Beach	28 83	5 0 0 0 0 5 0 0	10
11 S/P Road	25 45	0 0 0 0 0 10 5 0	10
12 A/P Road	5 95	0 0 0 0 0 10 5 0	10
13 SAM SeaPort	25 45	0 0 0 0 0 8 0 6	20
14 SAM AirPort	5 95	0 0 0 0 0 8 0 6	20
15 SEAPORT	25 45	0 0 0 20 10 4 0 0	15
16 AIRPORT	5 95	0 0 0 20 10 4 0 0	15
17 GTL	5 60	0 0 0 0 0 8 0 4	10
18 Blow Bridge	5 60	0 0 0 8 6 0 4 10	20

Figure 5.2 Parameters of tasks in the example

Platforms	Resource Capability Vector	Velocity
1 DDG	10 10 1 0 9 5 0 0	2
2 FFG	1 4 10 0 4 3 0 0	2
3 CG	10 10 1 0 9 2 0 0	2
4 ENG	0 0 0 2 0 0 5 0	4
5 INFA	1 0 0 10 2 2 1 0	1.35
6 SD	5 0 0 0 0 0 0 0	4
7 AH1	3 4 0 0 6 10 1 0	4
8 CAS1	1 3 0 0 10 8 1 0	4
9 CAS2	1 3 0 0 10 8 1 0	4
10 CAS3	1 3 0 0 10 8 1 0	4
11 VF1	6 1 0 0 1 1 0 0	4.5
12 VF2	6 1 0 0 1 1 0 0	4.5
13 VF3	6 1 0 0 1 1 0 0	4.5
14 SMC	0 0 0 0 0 0 10 0	2
15 TARP	0 0 0 0 0 0 0 6	5
16 SAT	0 0 0 0 0 0 0 6	7
17 SOF	0 0 0 6 6 0 1 10	2.5
18 INF (AAAV – 1)	1 0 0 10 2 2 1 0	1.35
19 INF (AAAV – 2)	1 0 0 10 2 2 1 0	1.35
20 INF (MV22 – 1)	1 0 0 10 2 2 1 0	1.35

Figure 5.3 Parameters of platforms in the example

5.2 Performance measures

In this section, we outline the organizational and algorithmic performance measures used to evaluate the results obtained from the *NGA*.

A. Average Platform Transfer Time

$$\bar{t} = \frac{\sum_{m=1}^{M} \sum_{T_i \in \underline{T}_m} t(m,i) + \sum_{T_i \in \hat{\underline{T}}} \hat{t}(i)}{N_p} \quad (5.1)$$

B. Clustering Efficiency

The clustering efficiency seeks to find the effectiveness of the grouping algorithm. It has a value of one when there are no exceptional tasks. It is given by

$$\tau = \frac{K}{\sum_{j=1}^{N_p} \sum_{i=1}^{N_t} TP(i,j)} \quad (5.2)$$

where $K = \sum_{m=1}^{M} \sum_{j=1}^{N_p} \sum_{i=1}^{N_t} PD(j,m)TD(i,m)TP(i,j)$ (5.3)

C. Average Task Accuracy

$$\bar{a} = \frac{\sum_{i=1}^{N_t} [\hat{A}_{Inter}(i)]}{N_t} \quad (5.4)$$

D. Average Platform Utilization

Platform utilization is related to resource usage in the organization. The average utilization of platform P_j is defined as

$$U(j) = \left(\frac{1}{\hat{L}} \sum_{l=1}^{\hat{L}} \frac{\hat{r}_P(j,l)}{r_P(j,l)} \right) \quad (5.5)$$

where $\hat{L}$ is the number of non-zero resource capability of platform P_j, $\hat{L} = \sum_{l=1}^{L} r_P(j,l)$, and $\hat{r}_P(j,l)$ is the resources of type *l* being used on platform P_j

$$\hat{r}_P(j,l) = \min\{r_P(j,l), \sum_{i=1}^{N_t} TP(i,j) r_T(j,l)\} \quad (5.6)$$

The average platform utilization is given by

$$\bar{u} = \frac{\sum_{j=1}^{N_p} U(j)}{N_p} \quad (5.7)$$

5.3 The settings of *GA* parameters

The values of *GA* parameters for the *OLG* and the *ILG* are listed in Table 5.2.

	Parameters	Value		
		OLG	ILG1	ILG2
1	No. of boundary mutation operators	6		
2	No. of uniform mutation operators	4		
3	No. of non-uniform mutation operators	6	4	4
4	No. of inversion mutation operators		4	4
5	No. of two-point crossover operators	7		
6	No. of multi-point crossover operators	8		
7	No. of arithmetical crossover operators	4	7	7
8	Maximum No. of DMs	2-6		
9	*q* (The probability of selecting the best individual)	0.1	0.1	0.1
10	a (The parameter used in arithmetical crossover	0.25	0.25	0.25
11	The maximum number of generations	300-500	200-300	300-400
12	Population size	60	40	50

Table 5.2 Settings of *GA* parameters

5.4 The organization obtained from *NGA* and its performance evaluation

In this section, we discuss the results of applying the *NGA* for the organizational design problem using four cases. In *Cases 1* and *2*, different value of ρ_1 and ρ_2 combinations are tested in order to explore the effect of *TASI* on the performance of *NGA*. The number of DMs is fixed at 4 for these two cases. The sensitivity of design for different numbers of DMs is explored in *Case* 3 and is used to determine the optimal number of DMs for the organization. In *Case 4*, we changed the value of intra-DM workload weight α from 0.1 to 0.5 to test its influence on the performance of *NGA*.

Case 1: *M=4,* $\rho_1 = \rho_2 = \rho$, α *=0.2*

Both *ILG 1* and *ILG 2* use the same task accuracy significance indices (*TASI*). The results for different values of ρ are listed in Table 5.3.

Case 2: $M=4,\ \rho_1 \neq \rho_2,\ \alpha=0.2$

ILG 1 and *ILG* 2 use different task accuracy significance indices (*TASI*). The results for this case are listed in Table 5.4

ϱ	Tran. Time	Clu. Effi.	Task Accu.	Plts. Util.
1	6.13	0.56	0.892	0.447
2	4.37	0.71	0.865	0.529
2.5	7.63	0.52	0.851	0.531
3	7.92	0.85	0.992	0.562
3.5	7.14	0.84	0.994	0.792
4	4.85	0.96	0.996	0.734

Table 5.3 Performance measures for Case 1

ϱ1	ϱ2	Tran. Time	Clu. Effi.	Task Accu.	Plts. Util.
1	2	4.45	0.47	0.816	0.447
	3	13.88	0.42	0.96	0.593
	4	19.77	0.35	0.985	0.643
2	1	4.42	0.73	0.864	0.54
3		5.54	0.84	0.93	0.616
4		7.6	0.91	0.948	0.703

Table 5.4 Performance measures for Case 2

***Discussion of Cases 1 and 2*:** When $\rho_1 = \rho_2 = 4$, we obtain moderate average platform transfer time, the highest average task accuracy, reasonable platform utilization, and the highest clustering efficiency. The clustered task-platform incidence matrix is presented in Table 5.5 for $\rho_1 = \rho_2 = 4$.

From Table 5.5, we note that there are 4 blocks on the diagonal of the matrix. They represent the task/platform grouping. The first row of the Table is the task ID and the first column represents the platform ID. There are only 2 exceptional tasks, T_5 and T_{18}; they can not be processed by a single DM. The coordination pattern resulting from the external coordination workload (inter-DM workload) is shown in Figure 5.4. DM*2* and DM*3* (DM*3* operates autonomously and is not shown in Figure 5.4) can execute their assigned tasks independently, while DM*1* and DM*4* need to coordinate with DM*2*. The amounts of external coordination between DM*2*→DM*1* and DM*2*→DM*4* are shown on the links. These correspond to $W_{Inter}(1)$ and $W_{Inter}(4)$ computed from Eq. (3.17).

Case 3: $\rho_1 = \rho_2 = 4,\ \alpha=0.2$

The solution generated by *NGA* depends on the specified organization size (i.e., number of DM cells). Although we did not consider workload constraints on each DM in this paper, a reasonably balanced workload results from the *NGA*. This is because the *WRMS* criterion minimizes both the mean and variance of workload across DM cells. We define the average aggregated workload as:

$$\overline{w} = \frac{\sum_{m=1}^{M}[W_{Intra}(m) + W_{Inter}(m)]}{M} \qquad (5.8)$$

	3	15	16	18	6	11	12	13	14	17	1	2	4	5	7	8	9	10
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0
17	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0
20	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Table 5.5 Clustered task-platform incidence matrix

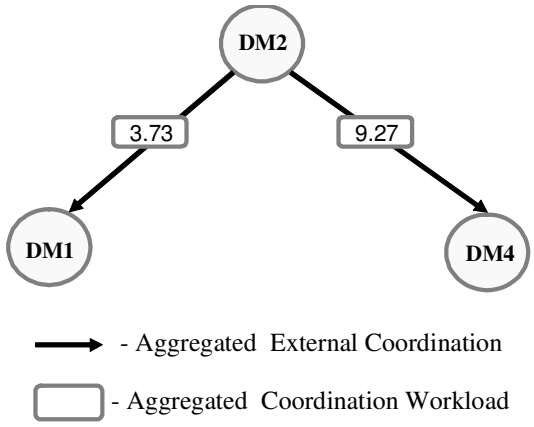

Figure 5.4 Coordination pattern of the organization

In Table 5.6, we compare the performance measures of each DM when we vary the number of DMs in the organization. Although increasing the number of DMs in the organization may potentially reduce the workload on each DM, the frequent inter-DM coordination will

impose an additional external workload for DMs in larger organizations. From Table 5.6, we find that when M=4, we obtain the best solution in terms of the average aggregated workload, as well as other measures.

	Tran. Time	Clu. Effi.	Task Accu.	Plts. Util.	Avg. W.L.
M=3	5.69	0.95	0.992	0.71	42.78
M=4	4.85	0.96	0.996	0.732	28.68
M=5	5.91	0.87	0.986	0.716	29.7
M=6	9.31	0.68	0.943	0.77	35.1

Table 5.6 Performance measures for case 3

Case 4: $\rho_1 = \rho_2 = 4$, *M=4*

The impact of weight α on the performance and workload distribution is shown in Tables 5.7 and 5.8, respectively.

	Tran. Time	Clu. Effi.	Task Accu.	Plts. Util.	WRMS
α=0.1	5.38	0.82	0.994	0.69	32.64
α=0.2	4.85	0.96	0.996	0.732	30.99
α=0.3	6.73	0.95	0.998	0.599	45.24
α=0.4	11.94	0.72	0.972	0.744	93.3
α=0.5	14.69	0.79	1	0.697	90.66

Table 5.7 Performance measure of *NGA* for case 4

		α=0.1	α=0.2	α=0.3	α=0.4	α=0.5
Intra	*D1*	0	224.8	17.3	0	0
	D2	274.5	101.8	119.4	177.9	235.3
	D3	367.9	141.4	106.7	388.6	264.5
	D4	0	53.6	207.7	0	0
Inter	*D1*	19.6	3.7	3.3	59.1	73.9
	D2	17.3	0	0	25.8	0
	D3	0	0	0	7.6	0
	D4	30.3	9.3	20	36.2	25.8

Table 5.8 Workload distribution for case 4

From Table 5.7, we note that both α = 0.2 and α = 0.3 achieve good results. When α = 0.2, the *WRMS* in Eq. (3.2) is significantly lower than that for α = 0.3. This implies that, when α = 0.2, workload is much better balanced among the DMs. Table 5.8 shows that an appropriate weight should be chosen. If α (e.g., 0.1) is too small, the task/platform grouping tends to produce less number of DMs in order to reduce the inter-DM workload; if α is large (e.g., 0.4 or 0.5), the minimization of inter-DM coordination workload is not guaranteed.

6. Conclusions and Future Extensions

In this paper, we proposed a novel congruent organizational design methodology based on group technology and a nested genetic algorithm to cluster the tasks and platforms into several DM cells to minimize the intra-DM and inter-DM workloads and maximize task execution accuracy. Different from our previous 3-phase organizational design, this algorithm solves the clustering problem and assignment problem simultaneously. Another advantage of this algorithm is that the scheduling of tasks and platforms, which is based on task/platform clustering and platform-task assignment, can be easily determined, which avoids the problem of solving a complicated mission scheduling problem during organizational design phase.

This algorithm needs to be extended along two directions. First, it does not take into account task precedence structure. Usually this will not affect the final results significantly. However, if there are too many simultaneously processed tasks at a single DM, some platforms may be overloaded. In our future research, task precedence constraints will be considered. How to extend the design approach to include flexibility and robustness criteria in the design process is another extension that will be addressed in our future work.

References

[Burton 1998] R. M. Burton, and B. Obel, "Strategic Organizational Diagnosis and Design: Developing Theory for Application", (2nd Ed.). *Boston, MA: Kluwer Academic Publishers*, 1998.

[Cheng 1999] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms: Part II. Hybrid genetic search strategies," *Comp. Ind. Eng.,* vol. 36, no. 2

[Gravel 1998] M. Gravel, A. L. Nsakanda, W. Price, "Efficient solutions to the cell-formation problem with multiple routing via a double-loop genetic algorithm," in *European Journal of Operational Research* 109 286-298 1998.

[Gupta 1995] Y. P. Gupta, M.C. Kumar, A. Sundaram, "Minimizing total inter-cell and intra-cell moves in cellular manufacturing: A genetic algorithm approach," in *International Journal of computer integrated manufacturing 8(2), 92-101.*

[Hyer 1989] N. L. Hyer and U. Wemmerlov, "Group technology in the US manufacturing industry: A survey of current practices". *IJPR*, 27(8): 1287-1304, 1989

[Joines 1996] J. A. Joines, C. T. Culbreth, and R. E. King, "Manufacturing cell design: An integer programming model employing genetic," *in IIE Transactions*, 28(1):69-85, 1996.

[Kazerooni 1995] M. Kazerooni, L. Luong, and K. Abhary, "Cell Formation using Genetic Algorithms," in *International Journal of Flexible Automation and Integrated Manufacturing,* 3:283-299(1995)

[King 1980] J. R. King, "Machine-component grouping formation in group technology", *International Journal of Management Science,* 8(2):193-199, 1980.

[Kusiak 1990] A. Kusiak "Intelligent Manufacturing Systems – Group Technology: Models and Algorithms", *in Prentice Hall international series in industrial and systems engineering* 1990.

[Kusiak 1987] A. Kusiak "The generalized group technology concept." *International Journal of Production Research,* 25(4) 1987

[Levchuk 2002a] G. M. Levchuk, Y. N. Levchuk, J. Luo, K. R. Pattipati, and D. L. Kleinman, "Normative Design of Organizations - Part I: Mission Planning", in *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 32, No. 3, May 2002, pp. 346-359.

[Levchuk 2002b] G. M. Levchuk, Y. N. Levchuk, J. Luo, K. R. Pattipati, and D.L. Kleinman, "Normative Design of Organizations - Part II: Organizational Structure", in *IEEE Transactions on SMC*, Vol. 32, No. 3, May 2002, pp. 360-375.

[Levchuk 2003] G. M. Levchuk, Y. N. Levchuk, C. Meirina, and K. R. Pattipati, and D. L. Kleinman, "Normative Design of Organizations - Part III: Modeling Congruent, Robust, and Adaptive Organizations", submitted to *IEEE SMC, 2003*

[Mackenzie 1986] K. D. Mackenzie, *Organizational Design: The Organizational Audit and Analysis Technology*, Alex Pub. Co., Norwood, New Jersey (1986).

[McCormick 1972] W. T. McCormick, Jr., P. J. Schweitzer, and T.W. White, "Problem decomposition and data reorganization by a cluster technique." *Operations Research*, 20(5): 993-1009, 1972.

[Michalewicz 1992] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs," *Springer-Verlag, New York.*

[Nadler 1997] D. A. Nadler and M. L. Tushman "Competing by Design," *New York Oxford* 1997.

[Nagi 1990] R. Nagi, G. Harhalakis and J.M. Proth, Multiple routings and capacity considerations in group technology applications," in *International Journal of Production Research,* 28(12) 1990

[Steudal 1987] H. Steudal and A. Ballakur. "A dynamic programming based heuristic for machine grouping in manufacturing cell formation." *Computers Industrial Engineering*, 12(4): 215-222, 1987

[Suresh 2000] C. Suresh "Neural Network Applications for Group Technology and Cellular Manufacturing", in "*Computational Intelligence in Manufacturing Handbook Chapter 4*", CRC Press LLC, 2000