

3-23-2011

A Quantitative Approach to the Organizational Design Problem

Jesus A. Mena

Florida International University, alonso.mena@itesm.mx

DOI: 10.25148/etd.FI11050610

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>

Recommended Citation

Mena, Jesus A., "A Quantitative Approach to the Organizational Design Problem" (2011). *FIU Electronic Theses and Dissertations*. 413.
<https://digitalcommons.fiu.edu/etd/413>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

A QUANTITATIVE APPROACH TO THE ORGANIZATIONAL DESIGN PROBLEM

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

INDUSTRIAL AND SYSTEMS ENGINEERING

by

Jesús A. Mena

2011

To: Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by Jesús A. Mena, and entitled A Quantitative Approach to the Organizational Design Problem, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Irtishad Ahmad

Ronald Giachetti

Shih-Ming Lee

Chin-Sheng Chen, Major Professor

Date of Defense: March 23, 2011

The dissertation of Jesús A. Mena is approved.

Dean Amir Mirmiran
College of Engineering and Computing

Interim Dean Kevin O'Shea
University Graduate School

Florida International University, 2011

DEDICATION

This thesis is dedicated to my mother and sister. There is no doubt in my mind that without their love, patience, understating, and support, the completion of this work would not have been possible.

ACKNOWLEDGMENTS

I wish to thank the members of my committee for their support and patience. I would like to thank my major professor, Dr. Chin-Sheng Chen, who from the beginning, has confidence in my abilities to not only complete a degree, but complete it with excellence.

I would like to thank Amira Diaz, from the Engineering Dean's office, because working on my dissertation as a student overseas was a challenge. Every time I had a problem I could always count on her for help. For the same reason, I would like to thank my friend, Dr. Desiree Tejada.

ABSTRACT OF THE DISSERTATION

A QUANTITATIVE APPROACH TO THE ORGANIZATIONAL DESIGN PROBLEM

by

Jesús A. Mena

Florida International University, 2011

Miami, Florida

Professor Chin-Sheng Chen, Major Professor

The span of control is the most discussed single concept in classical and modern management theory. In specifying conditions for organizational effectiveness, the span of control has generally been regarded as a critical factor. Existing research work has focused mainly on qualitative methods to analyze this concept, for example heuristic rules based on experiences and/or intuition.

This research takes a quantitative approach to this problem and formulates it as a binary integer model, which is used as a tool to study the organizational design issue. This model considers a range of requirements affecting management and supervision of a given set of jobs in a company. These decision variables include allocation of jobs to workers, considering complexity and compatibility of each job with respect to workers, and the requirement of management for planning, execution, training, and control activities in a hierarchical organization. The objective of the model is minimal operations cost, which is the sum of supervision costs at each level of the hierarchy, and the costs of workers assigned to jobs.

The model is intended for application in the make-to-order industries as a design tool. It could also be applied to make-to-stock companies as an evaluation tool, to assess the optimality of their current organizational structure.

Extensive experiments were conducted to validate the model, to study its behavior, and to evaluate the impact of changing parameters with practical problems. This research proposes a meta-heuristic approach to solving large-size problems, based on the concept of greedy algorithms and the Meta-RaPS algorithm. The proposed heuristic was evaluated with two measures of performance: solution quality and computational speed. The quality is assessed by comparing the obtained objective function value to the one achieved by the optimal solution. The computational efficiency is assessed by comparing the computer time used by the proposed heuristic to the time taken by a commercial software system. Test results show the proposed heuristic procedure generates good solutions in a time-efficient manner.

TABLE OF CONTENTS

CHAPTER	PAGE
1. INTRODUCTION	1
1.1 Background and motivation	1
1.2 Problem description.....	3
1.3 Research objectives	6
1.4 Contribution and significance	7
1.5 Assumptions	8
1.6 Dissertation organization.....	8
2. LITERATURE REVIEW	11
2.1 Organizations: structure and design	11
2.1.1 Managerial roles	14
2.2 Empirical studies of span of control.....	15
2.4 Heuristic and meta-heuristics methods	22
2.5 Summary	28
3. RESEARCH PLAN	31
3.1 Problem definition and modeling.....	32
3.2 Valuation of coefficients affecting modeling factors.....	35
3.3 Proposed solution approach	37
4. PROBLEM DEFINITION AND MODELING	39
4.1 Problem characterization.....	39
4.2 Mixed-Integer formulation.....	40
4.3 Computational experiments.....	42
4.3.1 Eight-job problem and four stations (Case one, Section one)	43
4.3.2 Case one: Section two (Level 1 to n).....	52
4.3.3 Limit size problem for a comercial solver (Case two)	57
4.4 Conclusions	62
5. VALUATION OF COEFFICIENTS AFFECTING MODELING FACTORS.....	63
5.1 Complexity and compatibility factors (level 0).....	63
5.2 Primary dimensions of supervision work (level 1 to n)	69
5.2.1 Setting limits for the factors	71
5.2.2 Identification of relevant activities.....	73
5.2.3 Identification of instruments of measure	77
5.2.4 Application and validation.....	85
5.3 Conclusions	91
6. PROPOSED HEURISTIC APPROACH.....	92
6.1 Introduction of the proposed heuristic approach.....	92

6.2 Problem decomposition and design framework	92
6.3 Proposed heuristic approach.....	96
6.5 Solution improvement by a meta-heuristic (Meta-RaPS)	106
6.5.1 Introduction to Meta-RaPS.....	106
6.5.2. Modify priority rules using randomness.....	107
6.5.2 Composition of the “next available task”	110
6.6.3 Setting parameters	114
6.3 Evaluation of solution quality	118
6.4 Results and discussion.....	123
7. CONCLUSIONS AND FUTURE WORK	127
7.1 Summary	127
7.2 Future work	130
LIST OF REFERENCES	132
APPENDICES	139
VITA.....	174

LIST OF TABLES

TABLE	PAGE
Table 1-1 Data for six-job problem and four stations.....	6
Table 2-1 Summary of empirical studies of span of control.....	17
Table 2-2 Mathematical approaches of span of control.....	22
Table 2-3 Meta-heuristics for solving combinatorial problems.....	27
Table 2-4 Summary of similar models.....	30
Table 4-1 Problem data of eight jobs and four stations	44
Table 4-2 Complexity (α) and compatibility (η) data for case one.....	45
Table 4-3 Job allocations (Sub-section one).....	45
Table 4-4 Complexity (α) of jobs (J) in relation to workers (G)	46
Table 4-5 Processing time after complexity data.....	47
Table 4-6 Job allocations (Sub-section two).....	47
Table 4-7 Compatibility (η) for workers (G) and jobs (J)	48
Table 4-8 Processing time after compatibility data	49
Table 4-9 Job Allocations (Sub-section three).....	49
Table 4-10 Job Allocations (Sub-sections one, two and three)	51
Table 4-11 Data for Sub-section two (levels 0 \rightarrow 1).....	52
Table 4-12 Data for Sub-section two (levels 1 \rightarrow 2).....	53
Table 4-13 Supervisor allocations (Sub-section one).....	54
Table 4-14 Supervisor factors (levels 0 \rightarrow 1).....	54
Table 4-15 Manager factors (levels 1 \rightarrow 2).....	54
Table 4-16 Supervisor allocations (Sub-section two).....	55

Table 4-17 Supervisor factors (levels 0→1).....	55
Table 4-18 Manager factors (levels 1 → 2).....	56
Table 4-19 Supervisor allocations (Sub-section three).....	56
Table 4-20 Capacity (α) and cost (α) data for levels 0 → 1.....	57
Table 4-21 Supervisor factors (Case two).....	58
Table 4-22 Capacity (α) and cost (α) data for levels 1 → 2.....	58
Table 4-23 Manager factors (Case two).....	59
Table 4-24 Job allocations (Case two).....	59
Table 4-25 Supervisor and manager allocations (Section one).....	59
Table 4-26 Supervisor factors (Section two).....	60
Table 4-27 Manager factors (Section two).....	61
Table 4-28 Supervisor and manager allocations (Section two).....	61
Table 5-1 Complexity factors.....	66
Table 5-2 Compatibility questions.....	68
Table 5-3 Complexity and compatibility factors.....	69
Table 5-4 Relevant activities in supervisor factors.....	74
Table 5-5 Instruments for measure of each factor.....	77
Table 5-6 Instruments used for the planning factor.....	79
Table 5-7 Instruments used for the execution factor.....	80
Table 5-8 Instruments used for the training factor.....	83
Table 5-9 Instruments used for the control factor.....	83
Table 5-10 Level of importance of primary dimensions of supervision work.....	85
Table 5-11 Quantitative limit of each factor.....	86

Table 5-12 Evaluation of primary dimensions of supervision work.....	87
Table 6-1 Data of the numerical example.....	102
Table 6-2 Processing time with complexity and compatibility factors.....	103
Table 6-3 Procedure for Step 4.....	103
Table 6-4 Job allocations at level 0.....	104
Table 6-5 Data of the numerical example (level 1)	104
Table 6-6 Supervision factors	105
Table 6-7 Data of capacity and cost (level 1)	105
Table 6-8 Procedure for level 1	106
Table 6-9 Processing time.....	108
Table 6-10 Candidates for “next available activity”	108
Table 6-11 Size of problems used to evaluate the meta-heuristic	114
Table 6-12 Subset of problems selected for setting parameters	115
Table 6-13 Deviation (%) from the optimal solution at level 0.....	116
Table 6-14 Frequencies out of six problems getting a deviation of 5% or less	117
Table 6-15 Deviation (%) from the optimal solution and the meta-heuristic	118
Table 6-16 Preliminary Meta-Heuristic Results (Case A and B)	118
Table 6-17 Average, std dev and maximum % difference from optimal.....	119
Table 6-18 Comparison of the meta-heuristic versus greedy algorithm.....	120
Table 6-19 Time of response.	120
Table 6-20 Approximate running time for bigger problems.....	122
Table 6-21 Average, std dev and max dev (%) between greedy and meta-heuristic.....	124

LIST OF FIGURES

FIGURE	PAGE
Figure 1-1 Mapping Work Breakdown Structure to Organizational Structure	2
Figure 1-2 Assigning jobs to resources.....	4
Figure 1-3 Graphic representation of the problem.....	6
Figure 3-1 Research plan	32
Figure 3-2 Research plan “A0 PROBLEM”	34
Figure 3-3 Research plan “A1 VALUATION (Level 0)”	36
Figure 3-4 Research plan “A1 VALUATION (Levels 1 to n)”	36
Figure 3-5 Research plan “A2 PROPOSED SOLUTION APPROACH”	38
Figure 4-1 Computational experiments	42
Figure 4-2 Sub-sections for analysis at level 0	43
Figure 4-3 Graphical representation of Section one	44
Figure 4-4 Graphical representation of results in Sub-section one.....	46
Figure 4-5 Graphical representation of results in Sub-section two.....	48
Figure 4-6 Graphical representation of results in Sub-section three.....	50
Figure 4-7 Utilization of each group en each Sub-section.....	51
Figure 5-1 Method for complexity factor	64
Figure 5-2 Example of skills and skill levels between workers and jobs	66
Figure 5-3 Method for compatibility factor	67
Figure 5-4 Method for supervisor factors	70
Figure 5-5 Impact of sum of factors in span of control	71
Figure 5-6 Factors between level 0 → 1	71

Figure 5-7 Design of measure rule.....	73
Figure 5-8 Activity implementation process.....	76
Figure 5-9 Evaluation of planning and training factor.....	88
Figure 5-10 Evaluation of training and control factor.	89
Figure 5-11 Relationship between groups and supervisor factors	90
Figure 6-1 Problem decomposition.....	93
Figure 6-2 Input/output for sub-problems.....	94
Figure 6-3 Meta-heuristic framework.....	95
Figure 6-4 Procedure of the proposed solution approach (level 0).....	98
Figure 6-5 Procedure for the proposed heuristic approach.....	99
Figure 6-6 Procedure of the heuristic approach (level 1 to n)	100
Figure 6-7 Procedure for the modified Lagoudakis greedy algorithm	101
Figure 6-8 Procedure of the meta-heuristic proposed at level 0	111
Figure 6-9 Meta-RaPS procedure for the solution approach	112
Figure 6-10 Procedure of the meta-heuristic proposed at level 1	113
Figure 6-11 Meta-RaPS procedure for the solution approach (level 1).....	113
Figure 6-12 Number of variables and constraints.....	115
Figure 6-13 Time of response.....	121
Figure 6-14 Interpolation of the number of variables.....	122
Figure 6-15 Average run times for meta-heuristic.....	125
Figure 6-16 Average, std dev and max percentage of solution.....	126
Figure 7-1 Future work	130

1. INTRODUCTION

1.1 Background and motivation

The span of management is the most discussed single concept in classical and modern management theory. Throughout the evolution of this concept, it has been referred to by various alternative terms, such as span of control, span of supervision, and span of authority [1]. Existing research work has focused mainly on qualitative methods to analyze this concept, for example heuristic rules based on experiences and/or intuition. Currently there are only a few reports that focus on a quantitative approach to an organizational structure design problem.

The challenge of mass customization (e.g., building cars to customer order), brings great value to both the customer and the company. For example, building cars to customer order eliminates the need for companies to hold billions of dollars worth of finished stock. Any company able to free this capital would improve their competitive position, and be able to reinvest in future product development. The question for many company executives is how efficient the organizational structure could be. The need for frequent adjustment to an organizational structure can be found in this type of make-to-order or project-based companies, where work contents and its organizational structure could vary dramatically over a short period of time. For companies that are going through their transitional stages (e.g., development and growth), an optimal design tool can help evaluate, justify, and optimize their organizational structure from time to time.

Meier and Bohte [2] have recently reinvigorated the debate on span of control and the optimal manager-subordinate relationship. They offer a theory concerning the impact and

determining factors of span of control, and they test it using data from educational organizations. Theobald et al. [3] suggest that manager-subordinate ratios, along with other structural influences on production, deserve considerably more attention than they have received in modern administration research.

According to Van Fleet [1], the specific number of subordinates supervised by any given superior is not crucial in and of itself but rather functions as an intermediate variable between factors, such as “routine work”, “trained subordinates”, “personal assistant used”, “stable operations”, “supervision shared”, and “need for control”, among others. Because of these factors, the number of span of control is only an intermediate variable between the factors and the effectiveness of supervision.

Figure 1.1 represents a typical problem of allocation of work given a Work Breakdown Structure (WBS) and its corresponding Organization Breakdown Structure (OBS). Each job is assigned to a resource (worker and/or workstation), which in turn is integrated into layers of hierarchical managerial units of the company.

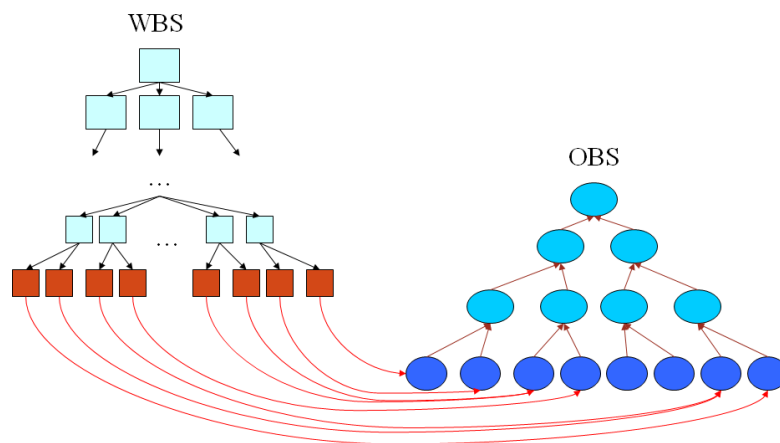


Figure 1-1 Mapping Work Breakdown Structure to Organizational Structure

In specifying conditions for organizational effectiveness, the span of control has generally been regarded as a critical factor. It is apparent that certain variables affect the span of control and hence organizational effectiveness, but a clear expression of the concept has been slow to emerge.

The span of control is an old problem which has generated a lot of qualitative approaches, but there are only a few reports on quantitative treatments of this problem. It is imperative to work with this problem by identifying, not only the factors that may affect the span of control, but also the criteria upon which any value judgment is to be based.

1.2 Problem description

Formally, the problem can be described as follows: a set of jobs (J) and a set of different processing workstations (G) are given. Each job in (J) is described by its processing time for the jobs to be completed, the skills needed for the job, the level of each skill (complexity factor), and compatibility according to the information given to the workstations. The objective is to find a set of workstations (G) to schedule the jobs (J) in such a way that the utilization of each workstation could be maximized, and to find the span of control needed for this Work Breakdown Structure (WBS). All of the above while maintaining the minimum cost of supervision. The number of “span of control” to be formed depends upon the number of workstations (G) considered in an instance, and also depends on the factors identified for supervision: planning, control, execution, and training. The following figure summarizes the description of the problem, where a group of jobs must be assigned to a group of available resources.

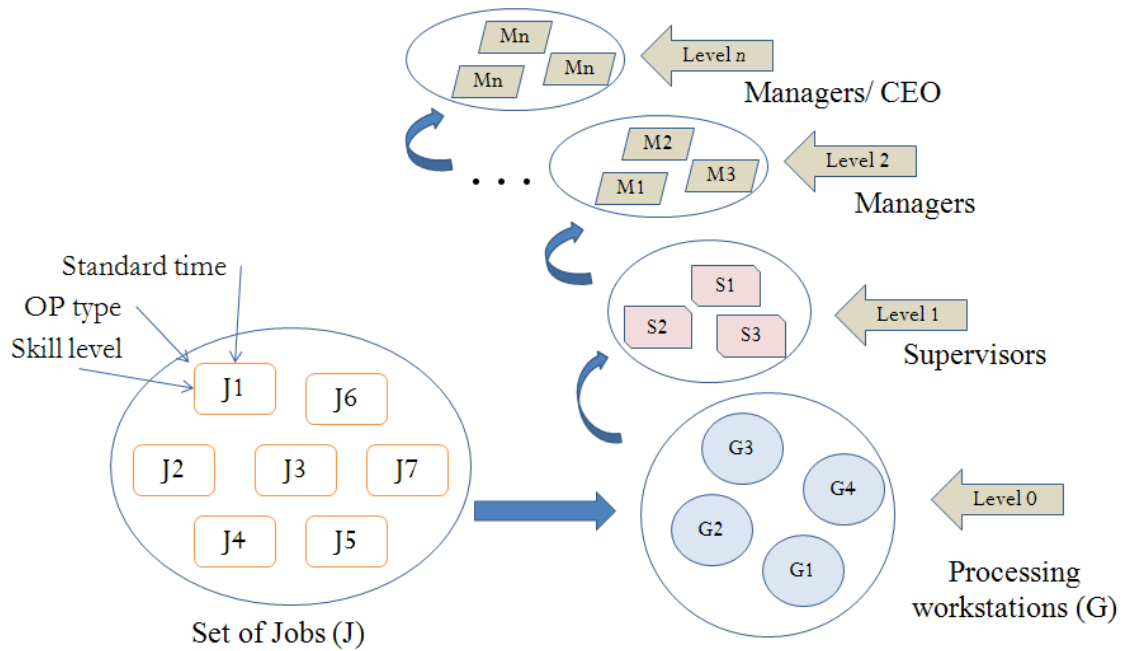


Figure 1-2 Assigning jobs to resources

Once the Work Breakdown Structure (WBS) has been established, it must be related to the Organization Breakdown Structure (OBS), as in Figure 1-1. In this process, the main assumptions made are: (1) job processing times and capacity of the workstations are deterministic and known in advance, (2) information about skills and skill level of each workstation and each job is available, (3) job splitting is not acceptable, (4) workstation is available according to the remaining capacity for the jobs, (5) it is feasible to eliminate or to add any workstations or supervisors as needed.

To exemplify and validate this description, Information Systems (IS) companies were contacted. The client determined the requirements for software development, which in turn established the WBS and indicated the tangible or intangible object produced as a result of the project (deliverables) that was assigned to accomplish this project. Inside the company, a set of programmers had to accomplish these deliverables. To assign the job

(or number of jobs) for every programming station, the standard time that every job needed and the capacity of every station had to be established first, in such a way that the sum of jobs assigned to every station did not exceed the capacity of every workstation.

The skills of every workstation can change depending on the level of certification that exists in each one. For such an effect, it is necessary to correlate, across the parameter "complexity", the type of specialty that the job needs to be completed. What causes the "complexity" factor, depending on the level of specialty that exists at a station, is that the standard time stays constant, or that it increases or diminishes according to the skill vs the level of workmanship needed for every operation. The job is generally described in terms of the tasks to be completed, the skills needed for the task, and the level of each skill. Skills are different from one another, but not necessarily unique to their tasks, therefore the same skill could be required, whether at the same level or at a different level, for more than one task. Each job is assigned to exactly one worker (or station), while workers may be assigned multiple jobs provided the worker's capacity is not exceeded. The same information is used to define the possible incompatibility of jobs in regards to the workers.

Table 1-1 presents the data of a six-job problem instance, and figure 1-3 presents a graphic representation of the problem. Each rectangle represents a job and each circle represents a workstation; β represents the standard time of each job and r represents the capacity of each workstation.

Table 1-1 Data for six job and four station problem

J	1	2	3	4	5	6
β (hrs)	1	0.5	0.5	1	1.5	1

G	1	2	3	4
r (hrs)	2	2	2	1

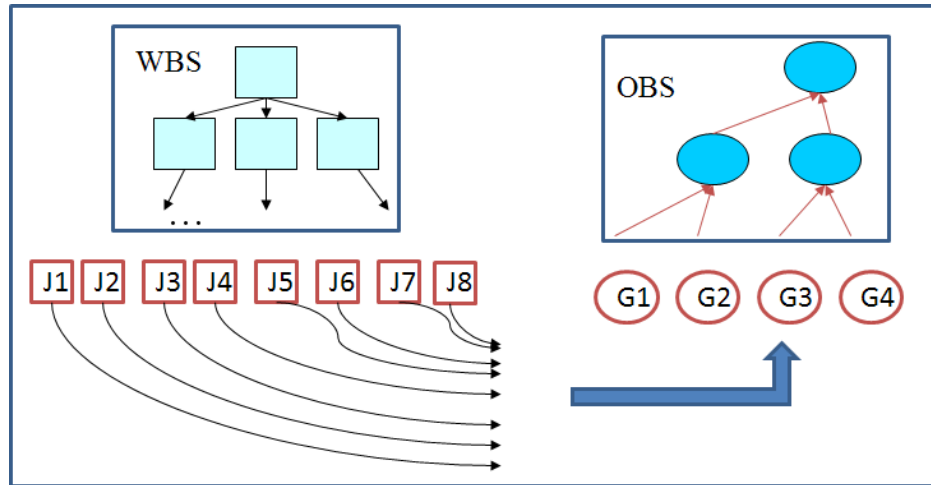


Figure 1-3 Graphic representation of the problem

1.3 Research objectives

The objectives of this research are to determine the feasibility of designing (or re-organizing) the organizational structure based on a quantitative model, and to determine the possible impact on organizational costs. From these objectives, one of the goals is to determine how flat an organizational structure should be. Another goal is to develop a method for the customization of the quantitative model proposed for some particular types of industries, and to develop instruments in order to measure up the parameters that affect the managerial role within the organizational structure. In order to accomplish these research objectives, the analytical modeling is a critical tool for the methodology that mathematically formulates the problem under study, and that leads to the

development of practical solution approaches to effectively solve large-size problem instances.

Quantitative models for combinatorial problems have been demonstrated to be NP-hard class, in such a way that, to solve this type of problems, the run-time of an exact algorithm increases exponentially with the instance size, implying that usually, only small instances can be solved in practice. Therefore, in order to solve real (large) problems, a heuristic and meta-heuristic approach are proposed in this research. The performance of the approach is measured by the computational cost and the solution quality.

To accomplish these objectives, the deliverables of this research include: (1) the development of a quantitative model for optimal organizational structure design, (2) an evaluation of the parameters used in the model for an industry application, and (3) the development of a practical solution approach to the analytical model.

1.4 Contribution and significance

The principal contribution of this research is the development of a quantitative tool for study of the organizational design problem. This research makes a contribution to the analytical linkage between work contents and resource requirements, quantifying managerial task requirements, and relating management requirements quantitatively to the organizational hierarchy design. This quantitative model enables the user to analytically study the organization design problem, and prescribe an optimal design for real-life industrial applications.

This study leads to a new (i.e. analytical) research approach to the organizational design problem, and lays a solid foundation for further academic research. This model and its solution approach will help companies to evaluate or optimally design their organizational structure. The proposed research takes an innovative approach to this organizational design problem, which hopefully can lay the basis for quantitative research approaches.

1.5 Assumptions

For the problem under consideration, assumptions were made that the job processing times and the capacity of the workstations are deterministic and known in advance. Another assumption for the model proposed is that in any new type of industry, the method for job and worker allocations must be customized, which could involve the development of additional models according to the type of business. Once the Work Breakdown Structure (WBS) has been established, job splitting is not acceptable. For constructing a solution, the model starts with an existing set of workers, supervisors, and managers. In order to run the model, it is feasible to eliminate or to add any workstation or supervisor as needed. The minimum overhead as the target function was set, and assuming that material and other costs are not affected by the change of number of managerial layers, these were not considered. The model in this research is intended for applications as a design tool in the make-to-order industries and can also be applied to make-to-stock companies as an evaluation tool.

1.6 Dissertation organization

Chapter 1 outlines the motivation behind this study, the objective, the scope, and a summary of the contents. Chapter 2 presents a review of the academic literature that addresses the organizational design problem, as well as the empirical and quantitative methods used to solve it. Also an extensive review of the academic literature regarding related research problems was conducted. Chapter 3 discusses the research plan that includes the problem definition and modeling, the plan for the valuation of the factors that affect the job allocation and supervision effort, and the proposed solution approach.

Chapter 4 introduces the problem description, formulation, and computational experiments, clearly specifying the problem object of this research, the mathematical formulation for the problem under study, and also the preliminary results using computational experiments. Chapter 5 develops the valuation of the factors affecting job allocation and supervision efforts. In order to validate the model, data was collected within the context of Information Systems (IS) companies based in the city of Chihuahua, Mexico. The objective was to construct an instrument for providing a measure of IS organizational factors (planning, execution, training, and control).

In Chapter 6, the heuristic approach was developed to solve the mathematical model, and includes the analysis of the computational experiments conducted. The proposed solution approach was based on the concept of greedy algorithms and the Meta-RaPS algorithm. It includes the computational experimentation results and experiments using a heuristic, and improving it with a meta-heuristic. Then the solution quality of the meta-heuristic algorithm was compared with the results obtained using commercial software, and

finally, the response time for those bigger problems that cannot be solved by the commercial software was analyzed. Conclusions and future work are drawn in Chapter 7.

2. LITERATURE REVIEW

There exists a large body of literature in the area of organizational design. This review focuses on the study of span of control and other aspects relevant to quantitative modeling and solution approaches. Section 2.1 is an overview of organizational structure design; section 2.2 is a review of empirical models of span of control; section 2.3 reviews mathematical models of span of control; and section 2.4 reviews heuristic solution approaches to solving large-size problems.

2.1 Organizations: structure and design

Organization Theory (OT) has embraced multiple perspectives because it draws inspiration from a wide variety of other fields of study, and because organizations are too complex and malleable to ever be summed up by any single theory, design or structure. In this review, the objective is to analyze those aspects of OT that are directly related to the organizational structure design and consequently to the span of control.

In order to design hierarchical organizations, Smith [4] believed in the practice of work decomposition, and proposed the concept of “division of labor”, determining that division of labor represents a qualitative increase in productivity. Taylor [5] proposed the theory of “the scientific management”, whose principal objectives were the development of science for each element of a man’s work, the scientific selection, training and development of workers, and the division of work between workers and management in almost equal share. Based on the above, workers are to be supervised by a specialist foreman. Taylor [5] expected that the expertise of managerial workers could be maximized like the ones of specialists. However, vertical division of labor is conflicting

with one design principle of management theory: unity of command. This principle requires that each worker be directed by one and only one supervisor. The scalar principle is introduced so that the managerial roles are organized as a pyramid hierarchy of authority.

Fayol [6] developed a general theory of management to add a managerial perspective to the problem of organizational governance. According to Scientific Management Theory, he laid down the following “principles of management”: specialization, which is to encourage continuous improvement in skills and the development of improvements in methods; unity of command, where each employee has one and only one supervisor; line of authority, which implies formal chain of command running from top to bottom of the organization; and coordination by managers, including authority and unity of direction. In order to put this system to work, Taylor’s functional foremanship has to be abandoned, and unity of command needs to be established.

Blau and Scott [7] argue that it is impossible to find the best way to structure the organization. They point out that the size of the organization, the technology level, and the environment all have strong impacts on the organizational structure. The number of layers, the number of departments, and the number of job titles of an organization increase with size, but the rate decreases as the size increases. In similar terms, Kulik et al. [8] suggested there isn’t a single model for organizational structure design, and any proposed model should be based on the characteristics of the type of organization for which development is being done, taking into account the possible influences of the environment. Woodward [9] suggested that the organizational structure strongly depends

on the work structure, which in turn is decided by the organization's primary system of production. He classified organizations into three different categories according to their primary production system: (1) project/order based company with unit production/small batch, (2) mass production/large batch, and (3) continuous production. Make-to-order companies, which produce one-of-a-kind customized or small volume products, may not need as many levels of hierarchy since the production line system is not stable. In this type of companies, human workers frequently have to change the processes in order to meet requirements for different projects or orders. Therefore, more coordination efforts are needed to handle the exceptions. Companies that produce large volumes of identical products, referred to as "mass production", are typically highly automatic, and machines dominate instead of people. They may have more layers of structure and at the bottom level, the number of workers is very large. The companies that belong to the continuous production category are using machines to do all the work, and there isn't much coordination and control work involved. These organizations have the highest hierarchical structure.

Choo [10], among others, outlined distinct forms of organizational structures: functional, divisional, and matrix structure. Functional structure applies when the organization is small, geographically centralized, and provides few goods and services. A divisional structure applies when the organization is large, geographically disperse and when it produces a wide range of goods. Matrix structure applies when the organization needs constant coordination of its functional activities, and the organization could modify many traditional management practices. In general, the same organizational structure has been

proposed by different authors such as Price [11], who also included one variant of the divisional structure, and Armandi [57]. Burton [12] proposed practically the same classification as follows: functional, decentralized, and mixed functional. He included in the classification the environmental complexity (simple for functional and complex for the others), the environmental change (dynamic for functional and static for the others), and the environmental segmentation (present in decentralized and absent in the others). Lim et al. [13] used the same structure classification for their study on “Organizational Structure for the Twenty-first Century”. They also proposed that the development of the organizational structure is dependent on the expression of the strategies and behavior of the management and the workers as limited by the distribution of power between them, and influenced by their environment. The main contribution of the authors is the identification of the characteristics of the organizations in order to establish the variables and their impact in organizational design.

2.1.1 Managerial roles

Fayol [6] described a manager as a person who plans, organizes, commands, coordinates, and controls. Mintzberg [14] suggested that managerial roles are classified into three broad categories: interpersonal roles (figurehead), informational roles (monitor, disseminator, and spokesman), and decisional roles (entrepreneur, disturbance handler, resource allocator, and negotiator). Pavett and Lau [15] pointed out the most important roles for managers are that of leader, resource allocator and disseminator. Davis [16] categorized the managerial work into physical work and mental work. Physical work includes coordinating, guiding, and discussing. Mental work includes scheduling,

controlling, and planning. Katz [17] identified that administrators use three skills, in executing their work: technical, human, and conceptual.

Based on this literature review, managerial tasks are classified into four types: planning (such as initiation, preparation, resource allocation, and scheduling), control (such as monitoring, coordinating, decision making, dissemination, dispatching, and performance evaluation), training (such as lecturing, coaching, guiding and discussing), and execution (such as problem solving and implementation).

These definitions give a very clear context to define the organizational structure proposed through a mathematical model, and most of all, generates a context of the difficulties that can occur when wanting to model this type of organizational design.

2.2 Empirical studies of span of control

The span of control is defined, according to Van Fleet et al. [1], as “the number of subordinates who can be successfully directed by a superior”. The authors consider the terms of “span of management”, “span of control”, “span of supervision”, and “span of authority” as synonymous.

Empirical studies report that the ideal span of control is around five, but there are not conclusive tests that say this would be universally true. Entwisle and Walton [18] collected information on the importance of span of control and showed an average of five to seven in schools, and four to seven in small companies. In a study examining the span of control for manufacturing companies Udell [19], based on 67 personal interviews, reported spans of up to 30. Udell found that formal job descriptions did not add to span of control, that non-supervisory roles such as planning, budgeting, or meeting with

customers did not reduce span of control. Bell [20], based on a study of 33 departments in a community hospital, found correlations between the complexity of tasks realized by subordinates, and a limited span of control. His data indicated that the fewer the subordinates they regulate, the closer the supervision, and the more complex the supervisor's task. Blau [7] reported on a study of 250 government agencies confirming that organizations requiring higher qualifications of their personnel are more decentralized, exhibit narrow spans of control, and have large number of managerial levels in their hierarchy. Ouchi et al. [21] surveyed 78 department stores and reported that the typical supervisor handled 8.7 subordinates on average, and used about 50% of their time on supervision.

Woodward [9] conducted a study of over 200 British industrial firms and found that variations in spans of control were present across the different types of firms. He classified the investigated firms into three types: unit production firms (e.g., shipbuilding), mass production firms (e.g., food or mined minerals), and continuous production firms (e.g., chemical). Within each category, successful firms used similar spans. Meier and Bohte [2, 22] collected data on 678 Texas school districts and found that the optimal span of control is dependent upon three factors: diversity of functions performed by subordinates, tenure of subordinates, and the size of the organization. The more diverse group a supervisor oversees, the smaller the span of control. The more time subordinates have been doing the same job, the less supervision they require, and consequently, a large span of control is possible. Finally, the larger organizations tend to use specialists instead of generalists, which allows for wider spans. One conclusion that

can be reached from this review is that in modern administration research, the manager-subordinate ratios, along with other structural influences on production (as suggested by Theobald et al. [3]) deserve considerably more attention than they have received.

Table 2-1 Summary of empirical studies of span of control

Authors	Study of Span of Control	SOC
Van Fleet et al. (1977) [1]	History of span of control	~5
Entwisle and Walton (1961) [18]	Schools and small companies	~5-7
Udell (1967) [19]	67 personal interviews	30
Bell (1967) [20]	33 departments in a community hospital	*
Blau (1968) [7]	250 government agencies	**
Ouchi et al. (1974) [21]	78 department stores	8.7
Woodward (1980) [9]	200 British industrial firms	***
Meier and Bohte (2003) [2]	678 Texas school districts	****
<p>* Complexity of tasks performed by subordinates and a limited supervisory SOC. ** Higher qualifications of their personnel exhibit narrow spans of control, and have large number of managerial levels in their hierarchy. *** Successful firms used similar spans. **** Optimal SOC is dependent on: diversity of functions performed by subordinates, tenure of subordinates, and the size of the organization</p>		

2.3 Mathematical approaches of span of control

Graicunas [23] was among the first to propose a mathematical approach to the span of control. He defined three types of interactions, namely, single relationships, cross-relationships, and direct group relationships. Denote n^* as the number of subordinates and t^* as the unit of time a manager needs to spend for the interaction, and then the total managerial work (measured as time), w^* for a supervisor is estimated as $w^* = n^*(n^* - 1) + n^*(\frac{2^{n^*}}{2} - 1)] * t^*$.

Graicunas' [23] formula shows that as each additional subordinate increases, the managerial work increases significantly. Thus, he established that a supervisor, who

handles interactions with groups of subordinates in addition to direct interactions, leads to the exponential increase of the number of relationships and interactions that a supervisor must handle after a certain number of subordinates. His conclusions were that a number around four subordinates was right to avoid the exponential growth of relationships and interactions. With regards to this research, it is valid to conclude that the explanation of a span of control is far more complex for a precise specification using only one mathematical formula without valid constraints.

Urwick [24] imposed (in Graicunas' formula) a strict limit to the number of employees that an executive should have. Urwick's main contribution was to propose ten principles of management to avoid increasing the administrative or supervisory overhead cost. Urwick's principles are: objective (the overall purpose), specialization (one group, one function), coordination and definitions (jobs with duties and relationships), among others. Urwick [24] and Graicunas [23], rather than proposing a mathematical model that would identify an optimal number of span of control, sought to determine, through a mathematical formula, the effect of increasing the number of employees by direct and indirect relationships that are built with subordinates. This model serves as a reference of the first attempt to quantify the relationship between the number of subordinates and the supervisory work. However, this approach is only based on a formula that does not have a goal of optimizing, and neither places constraints on the formulation, assuming as input a number of existing subordinates, and the time required for interfacing with them, while the output is only the time required by the manager.

Mackenzie [25] focused on the determinants of, and the calculation for the maximum span of control. In Mackenzie's model, G_0 represents the person from which he is analyzing the maximum span of control. G_0 has n subordinates and m colleagues or superiors. The main assumptions in the model are: (1) Person G_0 must spend time because interactions and solving problems between interactions; (2) others interact with person G_0 in order to satisfy their own needs; and (3) there is an upper bound for how much the group will allow person G_0 to complete their activities. Then, the span of control is the available time divided by the communication time for each subject. Mackenzie based his model on interaction between supervisors and subordinates, assuming that the communication time could be specific to each contact and jobs. The model uses as constraint the time of coordination between superiors and subordinates, slack time, and internal calculation time. Thus, the principal difference from our model is that Mackenzie's model is leaving aside issues identified in the work of supervision, such as control and training.

Keren and Levhari [26] use an "internal computation time", which is proportional to the size and complexity of the task, in order to compute the span of control. They compute the span of control for each level in the hierarchy using the speed of completion, and the amount of the resources employed by a planning job as a decision variable. In this case, the "optimum span of control" is an increasing function of wages and fixed time costs, and the increment of the span of control from one level to the next is an increasing function of wage costs. In their model, the objective function has two elements: the direct cost of the members employed (C) in the hierarchy and the cost of profits (w) lost

through slow planning ($\min [C(y_r) + w\sum_{t=1}^T x_t]$). The constraints basically take into account the time of planning through the hierarchy without considering the factors of control, execution, and training as part of the model proposed in our research.

Nasrallah and Levitt [27] suggest that organizations of different sizes could have similar optimal structures, as long as they have similar normalized levels or interdependence between interactions. Their model establishes that the organization is at its most effective when it maximizes the interaction value (interaction that person i initiates with person j , h_{ij}), proportion of interaction time per person (p_{ij}), and the success probability (the probability that an attempt by i to interact with j will succeed); with a restriction assuming that p_{ij} must be equal to one. Although the above mentioned research is not oriented to span of control, the authors pursue a pragmatic "organizational design" paradigm, which is useful to analyze and compare the proposed organizational design in this research.

Yassine et al. [28] developed an analytical model for describing hierarchical organizations, through the creation and use of models of organizational interaction to balance the communication time within and between organizational hierarchies. Their research concludes by proposing a hypothetical managerial framework for characterizing different types of organizations, but they only base their model on the communication-time ratio (between and within hierarchies). The model of Yassine et al. [28] with the "size of the organization" (m) as objective, shows how it is affected by the span of control and the communication between different levels in the hierarchy (where $m =$

$\exp[k \ln k]^{\frac{T_1}{T_2}}$). The model does not allow making an estimate of the optimal span of control that the organization must have.

DePuy et al. [29] proposed a mathematical model to match employees to tasks based on the logistics skills of the employee and the skills necessary for the task. They presented a mathematical model which assigns tasks to workers, in such a way that the total training costs are minimized. The objective function is based on minimizing the training required by an operator at the first level of the organization ($\min \sum \sum \sum G_{ijk_1}$). This model is used as a reference in order to assign jobs to workers at level 0, but the DePuy model does not address the problem of span of control within the hierarchy of the organization.

LeBlanc et al. [30] developed and implemented an optimization model to minimize the total cost of the assignments, while maintaining a balanced workload for different managers assigned to construction projects ($\min \sum \sum C_{ij}X_{ij}$, where C is the intensity of assigning manager i to project j and X the decision variable). Their research focused only on one type of industry and for balancing the workload at supervision levels.

Awad et al. [31] developed a computer-based system to replace an existing manual method for assigning proctors (Carleton University in Canada, needs proctors during examination sessions to oversee the students as they write their final examinations), and as a result, freeing up valuable staff time in the overloaded scheduling office. They used “A Basic Genetic Algorithm for the Initial Assignment of Proctors” where the number of possible proctor assignments is extremely large. In a worst case scenario, there are p proctors, t examination time slots, and a possible assignment in every time slot. This

situation is similar to the assignment problem in our research at level 0. The authors combined some problem-specific heuristics and genetic-algorithm framework. This research is relevant in order to evaluate their heuristics and genetic-algorithm for their assignment problem, but the application is too specific in order to use it for a more general organizational design. Below is a summary of the literature on empirical studies and mathematical models of span of control.

Table 2-2 Mathematical approaches of span of control

	Mathematical Approaches	Objective Function	Constraints
Level 1 to n	Graicunas (1937) [23]	Total managerial work (measured as time)	-----
	Keren and Levhar (1979) [26]	Direct cost of subordinates	Planning time
	Mackenzie (1974) [25]	Maximum SOC	Coordination time
	LeBlanc et al. (2000) [30]	Minimize cost of assignments	Workload (balanced)
	Yassine et al. (2005) [28]	Size of the organization	Communication time
Level 0	Awad et al. (1998) [31]	Assigning proctors	Examination time
	DePuy et al. (2006) [29]	Minimize training cost	Job complexity

2.4 Heuristic and meta-heuristics methods

Feasible solutions of mathematical models of this organizational design problem could not be found in a reasonable amount of time for large-size problems, therefore a heuristic solution methodology is developed. The solution quality of the heuristic is compared to optimal for small problems using commercial software.

Foulds [32] defined the term heuristic as a “method which, on the basis of experience or judgment, seems likely to yield a reasonable solution to a problem, but which cannot be

guaranteed to produce the mathematically optimal solution”. Greedy algorithms have been reported to give a good quality solution for combinatorial problems, as in the research of Il'ev et al. [33]. He used this solution approach for a maximization problem as an objective function, and a performance guarantee of the greedy algorithm is proven in terms of the parameters of a feasible set of problems. McGovern et al. [34] used a greedy algorithm for a combinatorial optimization analysis of the unary NP-complete disassembly line balancing problem. Other analysis on similar problems with greedy algorithms can be found in the research work of Agnihotri [35], Angelopoulos [36], among others [37, 38].

The term meta-heuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for a local optimality [40]. Candidate solution methods for a problem of this type include Specific Heuristic Rules (SHR), Simulated Annealing (SA), Tabu Search (TS), Genetic Algorithms (GA), and Meta-heuristic for Randomized Priority Search (Meta-RaPS). The Meta-RaPS framework was chosen using results from recent research [44-49] that reports the following:

1. This method performs better than other meta-heuristics in problems such as the “Resource Allocation Problem (RAP)”, “Traveling Salesman Problem (TSP)”, and “Vehicle Routing Problem (VRP)” that have similar characteristics to the problem in this research.
2. The computation times are better than other meta-heuristics that have similar or the same quality solution.

3. Run times are not dramatically affected by the size of the problem and generate feasible solution at each iteration.
4. The method is relatively easy to implement

The extensive work of the authors has been analyzed to propose a revision of this meta-heuristic that allows to obtain a good solution in a reasonable computation time. Genetic Algorithms (GA), Tabu Search (TS), Simulated Annealing (SA) and Meta-RaPS are modern heuristics that succeed in leaving the local optimum by temporarily accepting moves that cause a worse value for the objecting function. As shown in Table 2-3, these meta-heuristics are the most common proposed in combinatorial problems.

Genetic Algorithms imitate some of the natural processes of evolution and selection. According to the literature on GA, this algorithm consists of the following components: Chromosomal Representation (each chromosome represents a legal solution to the problem, and is composed of a string of genes), Initial Population (serves as the starting point for the GA), Fitness Evaluation (defining an objective of fitness function), Selection (the selection procedure picks out two parent chromosomes), Crossover (cross over the parents to form a new offspring) and Mutation (mutate new offspring at each locus).

Simulated Annealing (SA) simulates the change in energy of the system when subjected to a cooling process, until it converges to a steady “frozen” state; in Tabu Search (TS) the idea is to derive and exploit a collection of principles of intelligent problem solving. This has adaptive memory using neighborhood search, and it moves from one current solution to the next after every iteration.

As reported by DePuy [41], Meta-RaPS is the result of research on the application of a modified COMSOAL approach to solve several types of combinatorial problems. COMSOAL is a computer heuristic originally reported by Arcus [42] as a solution approach to the assembly line balancing problem. The philosophy behind Meta-RaPS consists of three basic ideas: (1) the incorporation of randomness in a particular heuristic may dramatically improve the solution quality, (2) random combinations of heuristics may lead to better results than each heuristic individually, (3) the recent and ongoing increase in computer speed and capacity encourages the use of practical although computer-intensive methods for solving combinatorial problems.

Meta-RaPS constructs feasible solutions through the utilization of a greedy algorithm in a randomized fashion. A greedy algorithm often constructs a solution by iteratively adding elements or activities to the solution based the activity's priority value. Meta-RaPS modifies the way a greedy construction heuristic chooses the next activity to add to the solution by occasionally choosing an activity that does not have the best priority value.

DePuy al. [41] reported that, for this category of problems, it is possible to find a good type of heuristics based on the concept of greedy algorithm that produces good solutions, such as the one proposed by Lagoudakis (2001) and Eilon et al. (1971).

Table 2-3 presented below analyzes different meta-heuristics for solving combinatorial problems. Applications discussed include well known problems in the industrial engineering field such as: Traveling Salesman Problem (TSP), Resource Constrained Project Scheduling Problem (RCPSP), Multidimensional Knapsack Problem (MKP), Quadratic Assignment Problem (QAP), Early/Tardy Single Machine Scheduling (E/T

SMS), and Vehicle Routing Problem (VRP). These problems have a similar construction in objective function and constraints as the one proposed in this research.

DePuy et al. [43] developed and applied Meta-RaPS to solve three typical combinatorial problems: TSP, RCPS, and MKP. The solution approach (Meta-RaPS) is applied to several heuristic algorithms showing that the solution quality significantly improved. They compared their method against other meta-heuristics (GA, SA, TS) also getting a better quality of solution when comparing the percentage of deviation from optimal and run times. Results indicate the Meta-RaPS heuristic performed as well or better than other, more complicated heuristics.

Hepdogan et al. [44] for the E/T SMS problem used an heuristic and an improvement with Meta-RaPS, then he compares the results vs the SA approach. Their results report that Meta-RaPS obtains a better quality solution and shorter run times. Mora et al. [45] develops a Meta-RaPS for the QAP. He uses a five-rule construction heuristics, and then combines the greedy heuristics in Meta-RaPS. His results show that the use of Meta-RaPS significantly improves the solution quality of each of the five greedy algorithms. Hepdogan et al. [46] studied Meta-RaPS for 0-1 MKP. In their application, the priority rule selected uses the normalization of weights and then improvement with Meta-RaPS and GA, the latter having the best solution performance (0.53% deviation of GA vs 0.73% of Meta-RaPS). However, the computation time for the GA ranges between six to 65 minutes comparing to Meta-RaPS with seven to 35 minutes per problem.

Keith et al. [47] studied Meta-RaPS for the TSP. Two fairly simple priority rules were tested by these authors, the nearest neighbor and cheapest insertion rules, and then

improvement with Meta-RaPS showing that this method outperforms the priority rules with respect to percent difference from optimality. Song et al. [48] reports in the meta-heuristics for the VRP good results in a reasonable amount of time using Meta-RaPS, when compared to optimal solutions of other heuristics like the GA and the TS.

Table 2-3 Meta-heuristics for solving combinatorial problems

Author	Problem	METHOD				
		GrH	GA	SA	TS	MR
Mora et al. [45]	QAP	●				☐
Hepdogan et al. [44]	E/T MSP	●		●		☐
DePuy et al. [29]	TSP	●	●	●	●	☐
	RCPSP	●	●	●	●	☐
	MKP	●	●	●	●	☐
Hepdogan et al. [46]	MKP	●	☐			●
Keith et al. [47]	TSP	●				☐
Song et al. [48]	VRP	●	●		●	☐
Moraga et al. [49]	MKP	●	☐	●	●	☐
<u>Problem:</u>		<u>Method:</u>				
QAP: Quadratic Assignment Problem		GrH: Greedy Heuristic				
E/T SMS: Early/Tardy Single Machine Scheduling		GA: Genetic Algorithm				
TSP: Traveling Salesman Problem		SA: Simulated Annealing				
RCPSP: Resource Constrained Project Scheduling Problem		TS: Tabu Search				
MKP: Multidimensional Knapsack Problem		MR: Meta-RaPS				
VRP: Vehicle Routing problem		●: Heuristics methods used by the author				
		☐: Best solution found				

According to the authors, one of the improvements of Meta-RaPS is that the run times are not dramatically affected by the size of the problem, and generate feasible solution at each iteration. Moraga et al. [49] for the MKP report the same solution quality between Meta-RaPS and the GA. Although Meta-RaPS has practically the same average deviation from optimal than the GA, its performance is with a very short run time. The results shown here lead to the conclusion that this meta-heuristic is the best for the problem proposed in this research.

2.5 Summary

According to definitions and empirical studies of span of control discussed in the literature review, the following conclusions can be drawn: there isn't a generally applicable optimum span of control, the size of span of control is higher in the lower level, but smaller in the higher level, and the more complex the subordinates' tasks, the smaller the supervisor's span of control, and the more complex the supervisor's tasks, the narrower the span of control.

Regarding the mathematical models found in the literature review, conclusions lead to the fact that those models take into account a series of assumptions that are complex to use for a practical analysis. These models focus on a particular aspect of the work of a supervisor, and do not take into account the different parameters that affect this type of work.

The constraints of the models found in the literature focus only on one type of work performed by the supervisor (training or coordination). Therefore one of the main differences with the model presented in this research is that the constraints take into

account the complexity (and compatibility) of the jobs in the first level, and the activities that involve the supervision work needed at the following levels of the organization (classified as planning, control, training, and execution time).

The first constraint (assigning jobs to groups) is virtually ignored by the models related with the span of control (they take into account a group of permanent workers at the first level), and those models that focus on the task allocation problem ignored the required span of control needed for these working groups.

The model presented in this research extends the works discussed in the literature review (mathematical models), particularly the one of DePuy et al. [29], extending their original formulation and exploring a broad range of decision-making organizational structures that take into account factors in deciding the minimum cost of supervision. These factors include planning, training, execution, and control at higher levels, while taking into account the skills of workers for specific jobs in the lower level. DePuy et al. [29] model used five constraints to determine training needs, ensuring that all jobs are assigned, and maintain the capacity of workstations. The latter two being similar constraints in our model at the first level ($n=0$). However the quoted model does not consider other relevant factors in the allocation of jobs to workers, such as the compatibility of each job with respect to each of the workers involved.

The model proposed in this research includes six constraints to ensure that all jobs are assigned (similar to that proposed by DePuy et al. [29]), to maintain the capacity of the workstation, including the factors of complexity and compatibility to ensure the correct

allocation of jobs. In the following levels, the constraints seek to maintain the capacity of the supervisor including four relevant factors: planning, control, execution, and training.

The main difference from this research in regard to other models in the literature, takes into account only one of the factors, such as planning time [19], coordination and slack time [25] or training requirements [29]. The following restrictions on the model proposed in this research ensure that all working groups are assigned to a supervisor, and they in turn, are assigned to a manager whose capacity is affected also by the time spent on planning, control, execution, and training. The following table is a summary of the characteristics reported in models that analyze a similar problem to that outlined in this research.

Table 2-4 Summary of similar models

Research	Level 0*	Level 1 to n*	Planning **	Training **	Control **	Execution **
Keren and Levari	NO	YES	YES	NO	NO	NO
Mackenzie	NO	YES	NO	NO	YES	NO
DePuy et al	YES	NO	NO	YES	NO	NO
Proposed research	YES	YES	YES	YES	YES	YES
<p>* The research includes assignment of jobs to workers (level 0) and/or assignment of workers to supervisors (level 1 to n)</p> <p>** The research includes in the mathematical model a planning, training, control or execution factor.</p>						

3. RESEARCH PLAN

The main questions that concern this research are to determine, based on analysis of several cases, the feasibility of designing (or re-organizing) the organizational structure from a quantitative model, and to determine the possible impact in organization costs. Taking into account the above statement, one question arises: if an organizational structure is suggested to be flat, ¿how optimal should it be? The general solution approach to design or re-organize the organizational structure is through a quantitative modeling as integer-programming consisting of different levels of decision systems.

Authors from the existing Organization Theory establish a list of parameters and variables that affect the managerial role within the organizational structure. This research questions the possibility of developing instruments in order to measure up those variables in a quantitative method, and the customization of the model for some particular types of industries.

Quantitative models for combinatorial problems (TSP, MKP, VRP) have been demonstrated to be NP-hard class, in such a way that, in order to solve real (large) problems, it would require a heuristic solution approach. Therefore this research analyzes heuristic methods, and questions whether there is a particular one to obtain a good quality solution for the model proposed in a reasonable computation time. In order to answer these questions, the proposed research plan consists of: (1) problem definition and modeling, (2) valuation of the coefficients affecting modeling factors, and (3) proposed solution approach.

The following figure draws a map for the general research plan:

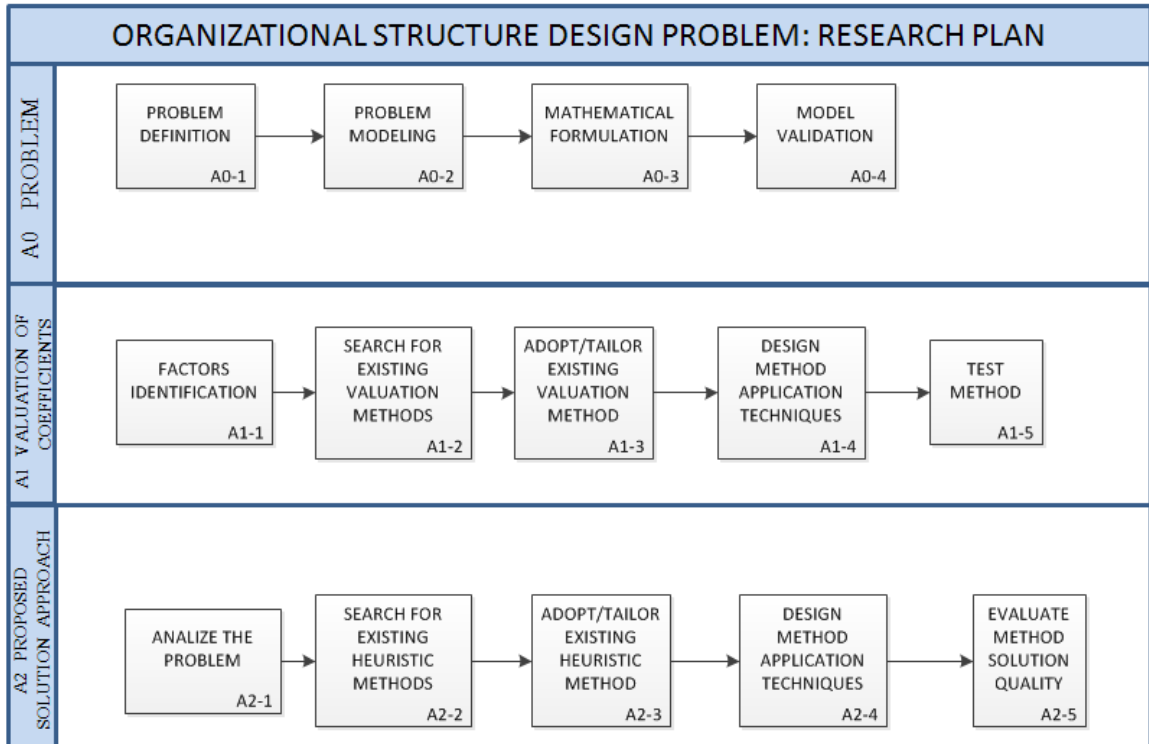


Figure 3-1 Research plan

3.1 Problem definition and modeling

In order to answer the first question regarding the feasibility of designing (or re-organizing) the organizational structure, it is necessary first to optimize the number of subordinates that report to each manager. The objective is to answer the question through a quantitative model design.

There are two main objectives for this modeling development: to provide an assignment of jobs to workers taking into account the skills of the workers in regards to the complexity and compatibility of the job, and to develop a more extensive model that assigns workers to supervisors and managers. The total cost required to achieve all the jobs and to assign all the workers has to be minimized. In this model, the minimum

overhead is set as the objective function, mapping the work structure to the organizational structure through the relationship between the number of jobs and the number of workers for each job.

The literature review allows the identification of limitations found in current mathematically and organizational structure design models. The results of the reviews help in the creation of a detailed set of equations that describe in reasonable detail how Organization Theory (OT)-based quantitative analysis might be constructed. There are two goals that companies would like to reach when they are working on organizational structure redesigning: one is to keep the cost untouched, but improve the organization performance, for example the productivity and quality level of products is kept the same or higher. The other one is to maintain the performance level and to reduce the overhead. A typical organizational structure consists of a set of workstations that process a variety of jobs and relate to different levels of supervision. The assignment of jobs is dependent upon the complexity and compatibility of the jobs to be done. The allocation of jobs in different workstations must maximize the utilization of installed capacity, while minimizing the number of workstations required. Jobs that are successfully processed are assigned only to one workstation and workstations are assigned, in turn, to only one supervisor. These characteristics of organizational structure make them suitable for modeling as integer-programming consisting of different levels of decision systems. DePuy et al. [29] proposed an integer model for the problem under study, but only for the first level (0). The proposed model in this research is compared with the DePuy model with respect to the number of decision variables and linear constraints, and with the ones

proposed by Keren and Levhari [26] and Yassine et al. [28], who also establish an objective function and constraints.

In this research a binary-integer programming model was first formulated and solved using a commercial solver (LINGO) to obtain an optimal solution. The proposed model intends to optimize the entire organizational structure by eliminating unnecessary work units, instead of optimizing local spans of control to improve the organizational structure. Because the model is NP-hard, an optimal solution can be obtained in reasonable time for only small problem instances.

The following figure establishes the design of this part of the research plan:

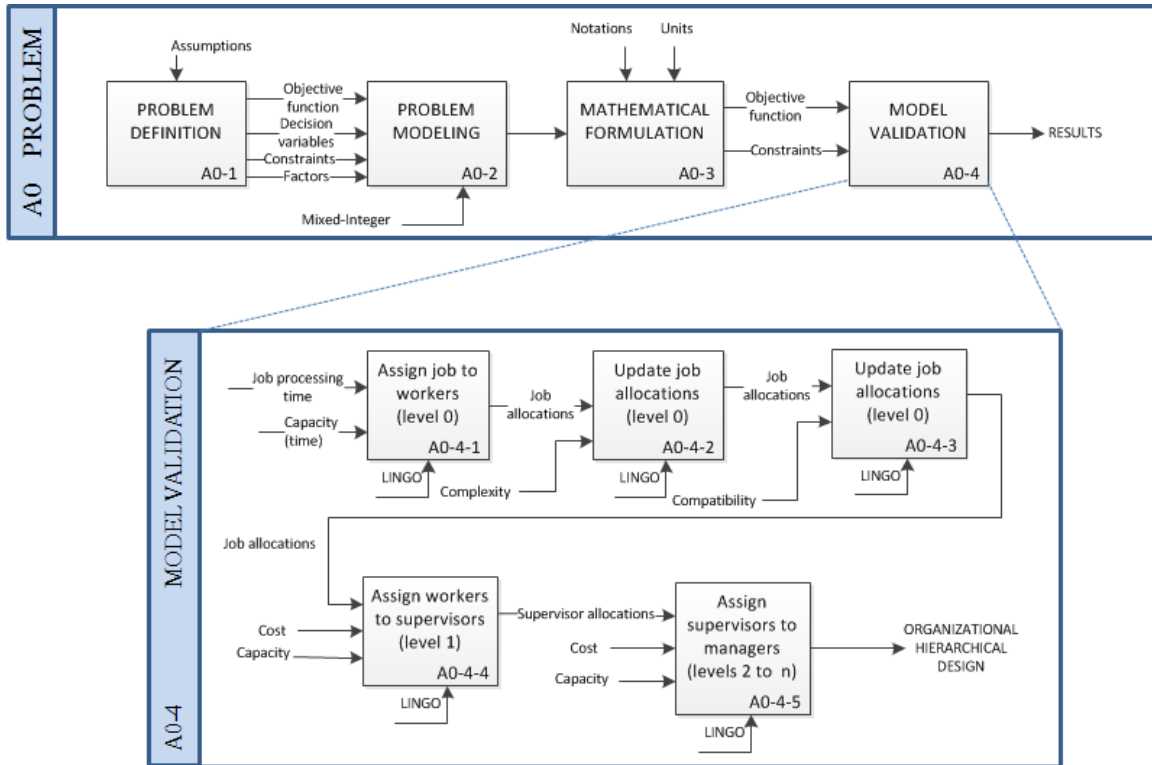


Figure 3-2 Research plan “A0 PROBLEM”

3.2 Valuation of coefficients affecting modeling factors

In order to develop and customize efficient problem instances for an industrial application, the valuation of the factors affecting job allocation and supervision effort (complexity, compatibility, planning, execution, training, and control) into the mathematical model were analyzed. Within the model, factors are set at the first level of decision (0) and for the following levels (1 to n). To quantify these factors, an instrument was developed, which gives us tools to determine the quantification of the factors in the model for decision making. On the first level (0), information of a job consists of the tasks (to be completed), the skills (needed for the task), and the level (required to complete each task). The objectives in estimating parameters at this level are: (1) determining their existing workers' skills and their levels, and (2) assigning the workers to their complexity and compatibility factors at an appropriate level regarding the jobs.

For the following levels (1 to n) regarding supervision effort, an initial set of criteria for assessing organizational factors were formulated. These factors take into account the concept of managerial work [6], its different categories [14], their classification (physical and mental work) [16], the skills needed for administrators [17], the roles of managers [15], and the classification of the types of managerial tasks [6, 16]. In order to be able to quantify and test the dimensions of supervision effort, data were collected on four Information System (IS) companies in the city of Chihuahua, México, with 30 supervisors and managers.

The following figures establish the design of this part of the research plan:

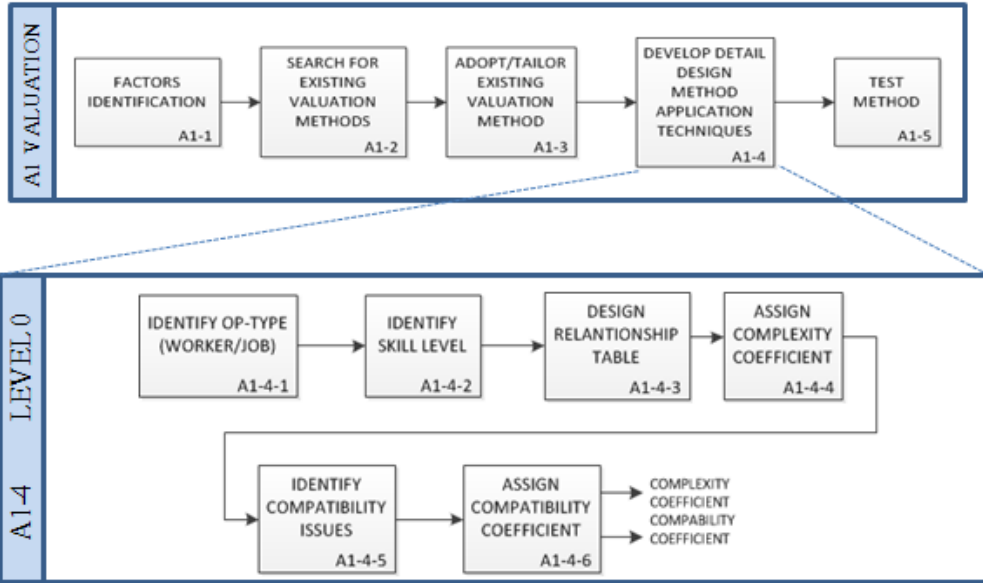


Figure 3-3 Research plan “A1 VALUATION (Level 0)”

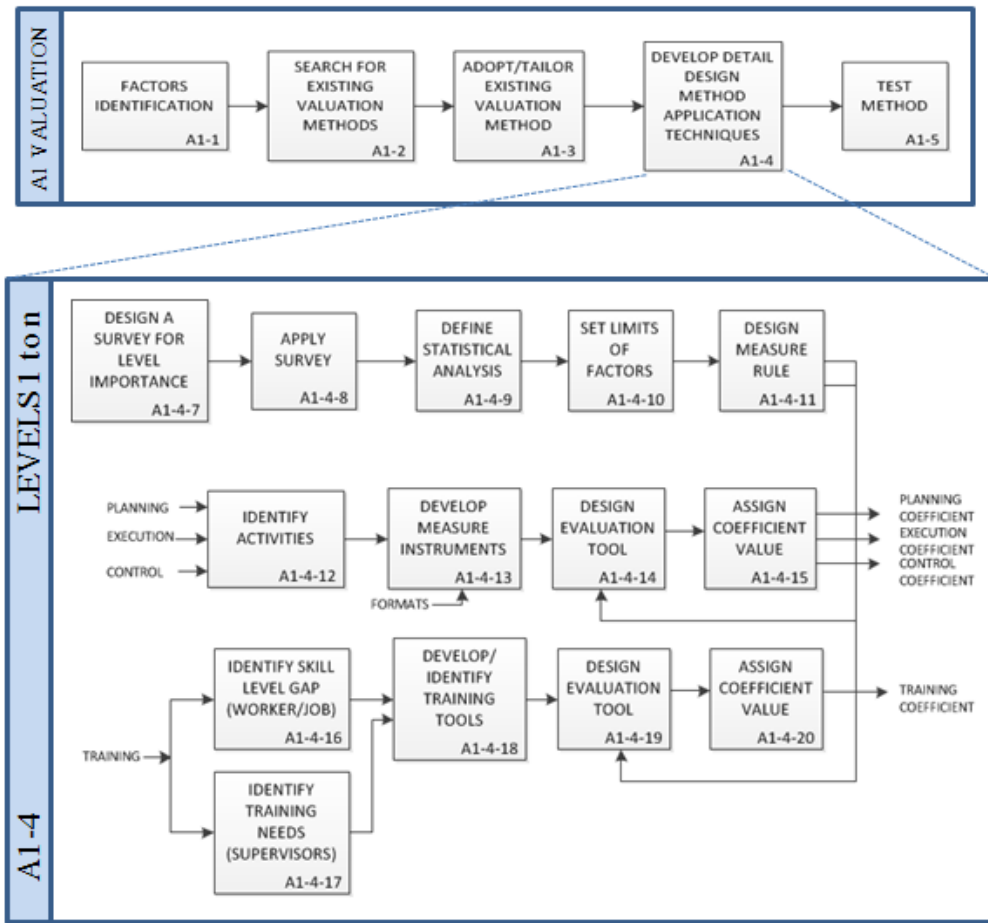


Figure 3-4 Research plan “A1 VALUATION (Levels 1 to n)”

3.3 Proposed solution approach

The proposed formulation inherits properties which help to decompose the problem under study into smaller sub-problems. For the problem (NP-hard class), there isn't an exact algorithm that terminates within polynomial-bounded computational time (for large-size problems). In order to solve such problems, heuristic algorithms have to be used to find a good solution at a reasonable computational time.

The literature reviewed on heuristic approaches (Table 2-3) justifies the utilization of Meta-RaPS framework for the type of model proposed in this research. In order to develop this solution approach, the philosophy behind Meta-RaPS was used consisting of three basic ideas: (1) the incorporation of randomness in a particular heuristic may dramatically improve the solution quality, (2) random combinations of heuristics may lead to better results than each heuristic individually, and (3) the recent increase in computer speed and capacity encourages the use of practical computer-intensive methods for solving combinatorial problems.

The general framework for constructing the solution approach development in this research follows the following criteria: (1) the structure of the problem consists of m constraints that will bound the construction of any feasible solution, when a *job* is chosen to be in the feasible solution, the addition of the new *job* cannot exceed the capacity constraints, (2) the framework begins with a simple rule called "greedy algorithm" proposed by Lagoudakis [41] in order to select the best job candidate with the best suitable worker at level 0, and the best suitable supervisor at the following levels, (3) the greedy algorithm is modified by altering the composition of the "next available activity",

adding randomness to the selection of jobs to workers, these to supervisors, and so on, (4) construct feasible solutions using priority rule and randomness, (5) pick the best answer for a number of iterations, and (6) evaluate the solution quality compared to optimal and computational requirements.

The following figure draws the design of the proposed solution approach:

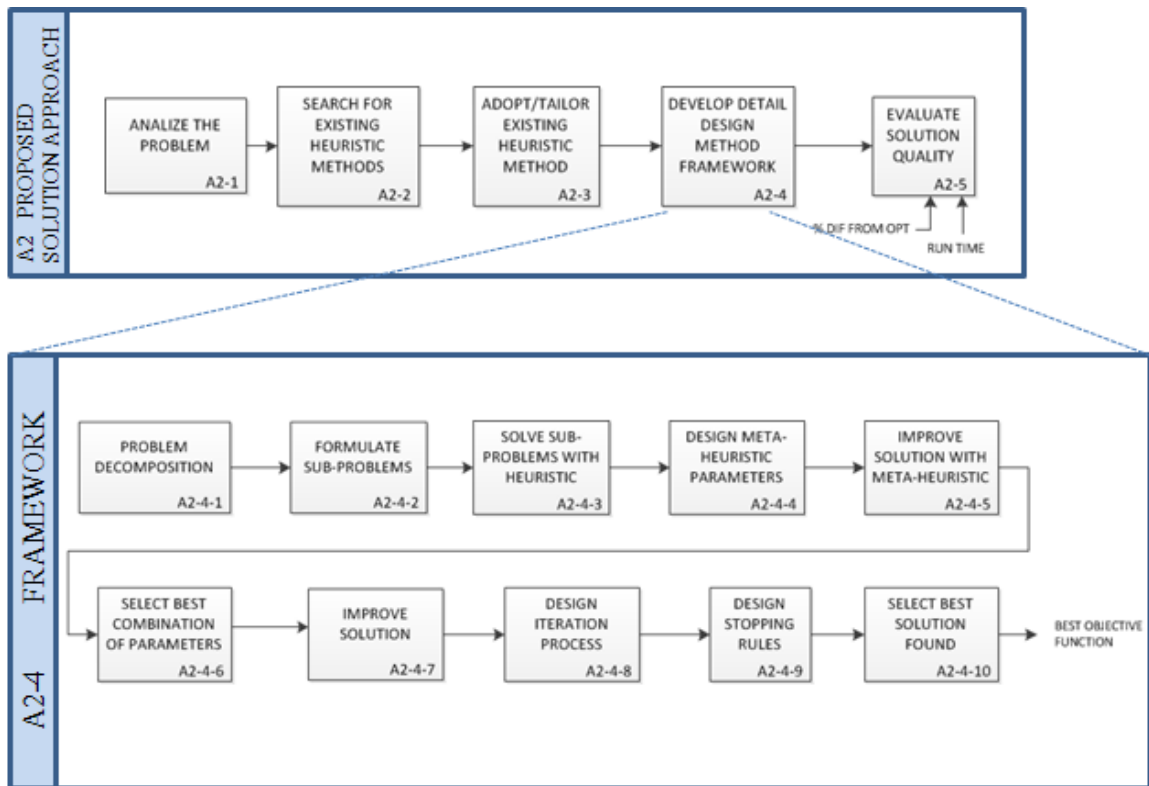


Figure 3-5 Research plan “A2 PROPOSED SOLUTION APPROACH”

4. PROBLEM DEFINITION AND MODELING

4.1 Problem characterization

The design process begins with decomposition of each work into elemental work units (called OP), and assigning each OP to direct resources (machine and worker required), in relevance to work complexity, resource competency (skill and proficiency), and their compatibility. It then moves to organizing the required resources into groups, sections, and departments, forming a hierarchical management structure. It assumes work can be decomposed into a hierarchy of standard OPs, of which specifications, requirements and methods have been formalized for each OP type. It further assumes management effort for each managerial activity is measurable and is unique for each company, depending on the experience and expertise it has, and tools and techniques it uses to run the operation.

Work in this study is originated as customer orders, work orders, contracts, projects, and/or forecasts. Work content is applied as either an order-specific process plan, or a standardized process plan, defined through the use of the Group Technology (GT) concept. The modeling effort begins with the first level (0) problem, where there is a given set of structured tasks to be processed by workers at workstations (G). Each task is defined with an OP type and a proficiency level. Each worker (or workstation) possesses different skills. Complexity and compatibility are two factors used at this decision level (0), to ensure that each job is assigned to a proper workstation. Once all direct jobs are assigned and the requirement for each station is established at level 0, the decision effort moves to managerial requirements at higher levels of the organizational hierarchy, (i.e. levels 1 to n). This prescriptive model seeks an optimal organizational structure, i.e. span

of control and management layers, that minimizes the overall operation costs. Along with the modeling, an initial set of factors was identified and used in the constraints to assess organizational parameters pertaining to the nature of managerial effort, work type, and required skills.

4.2 Mixed-Integer formulation

The following notations were used to model the problem. The unit of time is the hour.

0 is the bottom level; n is the top level, $n \in N$

Parameters:

- w_{nj} : the unit cost coefficient for section j at level n
- β_i : the processing time for job i , $i \in I$
- r_{oj} : the capacity of the working group j , $j \in J$
- α_{ij} : the complexity factor when job i is assigned to group j
- p_{nj} : the planning factor for section j at level n
- c_{nj} : the control factor for section j at level n
- t_{nj} : the training factor for section j at level n
- e_{nj} : the execution factor for section j at level n
- f_{nj} : the capacity of section j at level n
- n_{ij} : the compatibility factor when job i is assigned to group j

Decision variables:

- $z_{ij} = \begin{cases} 1 & \text{if job is assigned to group } j \\ 0 & \text{otherwise} \end{cases}$

- $y_{0jp_0} = \begin{cases} 1 & \text{if group } j \text{ with section } p_0 \text{ at bottom level is occupied} \\ 0 & \text{otherwise} \end{cases}$
- $y_{njp_n} = \begin{cases} 1 & \text{if section } j \text{ with section } p_n \text{ at level } n \text{ is occupied} \\ 0 & \text{otherwise} \end{cases}$

The mixed-integer formulation developed is as follows:

$$\text{Min } \sum_{n=1}^N \sum_{j=1}^J w_{nj} f_{nj} y_{njp_n} + \sum_{j=1}^J y_{0jp_0} \quad (\forall j \in J), (\forall n \in N) \quad (3.2.1)$$

Subject to:

$$\sum_j z_{ij} = 1 \quad (\forall i=1 \dots m) \quad (3.2.2)$$

$$\sum_i \beta_i \alpha_{ij} z_{ij} / n_{ij} \leq r_{0j} y_{0jp_0} \quad (\forall j=1 \dots n) \quad (3.2.3)$$

$$\sum_j r_{0j} y_{0jp_0} (\rho_{ij} + c_{ij} + t_{ij} + e_{ij}) \leq \sum_j f_{ij} y_{ijp_1} \quad (\forall i=1 \dots m) \quad (3.2.4)$$

$$y_{0jp_0} \leq \sum_{p_1} y_{1p_0p_1} \quad (3.2.5)$$

$$\sum_j f_{n-1,j} y_{n-1,jp_{n-1}} (\rho_{ij} + c_{ij} + t_{ij} + e_{ij}) \leq \sum_j f_{nj} y_{njp_n} \quad (\forall i=1 \dots m) \quad (3.2.6)$$

$$y_{n-1,jp_{n-1}} \leq \sum_{p_n} y_{np_{n-1}p_n} \quad (3.2.7)$$

$(z_{ij}, y_{0jp_0}, y_{njp_n} \text{ binary}, (\forall j \in J), (\forall i \in I))$.

In this formulation, the objective function (3.2.1) is to minimize the supervision cost. Constraint one (3.2.2) ensures each job (operation type) is assigned to only one direct labor group. Constraint two (3.2.3) ensures each direct labor group has a finite capacity, and all job assignments are limited by this capacity. The following two constraints (3.2.4 and 3.2.6) ensure that the total managerial work requirement of its child units cannot exceed the finite capacity imposed on the parent unit. Finally, the last two constraints (3.2.5 and 3.2.7) ensure that each child unit must have a parent unit.

4.3 Computational experiments

LINGO, a commercial software system, is used to help validate the proposed 0-1 integer program, study its behavior, and explore its limits. A preliminary computational experiment was conducted that validates the model with practical problems, and evaluates the impact of changing the complexity and compatibility parameters at level 0. It also evaluates the impact in managerial-related parameters for planning, training, execution, and control at the higher levels. The proposed model was implemented on LINGO and run on a workstation computer (DELL PRECISION | M4400) equipped with processor Intel Core 2 Duo T9400 2.53 GHz. This experimentation demonstrates the use of the proposed model, and reveals that the software system is limited to solving small size problems. It cannot produce an optimal solution for large-size problems in a feasible computer (run) time.

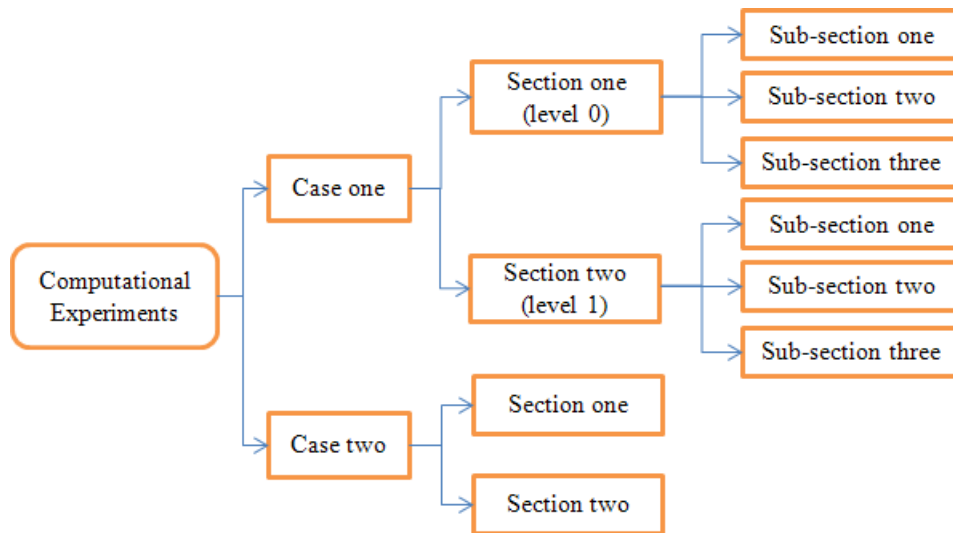


Figure 4-1 Computational experiments

Figure 4-1 shows the design structure of cases performed for this study. The objective of the first case is to determine the impact on the variation of all the factors involved in the model at all levels (0 to n). The aim of the second case is to investigate the size of problems that can be solved using LINGO.

4.3.1 Eight-job problem and four stations (Case one, Section one)

The following figure shows how the first case proposed was divided.

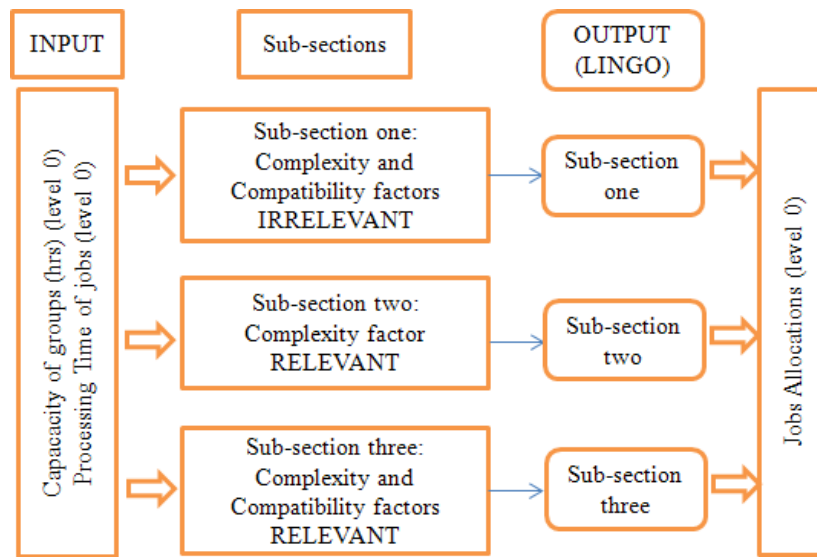


Figure 4-2 Sub-sections for analysis at level 0

It consists of two sections: the first one for allocation of jobs to workers (level 0), and section two for assignment of workers to supervisors (level 1 to n). In order to verify the allocation of jobs (from WBS) at level zero, section one was divided into three Sub-sections: (1) sub-section one where complexity and compatibility factors of jobs regarding the workers are irrelevant, (2) Sub-section two in order to explore the impact

on the solution when complexity is relevant, and (3) Sub-section three in order to explore the impact of both factors in the solution.

The following figure shows, graphically, the structure of the first section of the case where there are eight jobs to be processed in four working groups.

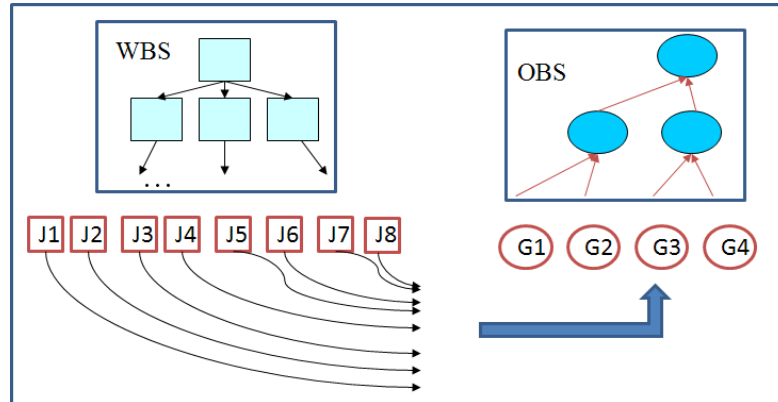


Figure 4-3 Graphical representation of Section one

The data for section one of this case consist of the capacity of each working group and the time required to accomplish each job (Table 4-1). In order to solve the model, data for level 1 and 2 need to be fed into the model (these tables are in Appendix A). All equations to run this case in LINGO are shown in Appendix A.

Table 4-1 Problem data of eight jobs and four stations

GROUPS	G1		G2		G3		G4	
CAPACITY (HRS)	1		2		1		1	
JOBS (i)	J1	J2	J3	J4	J5	J6	J7	J8
TIME *	0.2	0.3	0.5	1	0.8	0.75	0.1	0.15

* Time required to accomplish the job (β in hours)

4.3.1.1 Sub-section one

Assuming the complexity and compatibility factors of the jobs are irrelevant to the groups, the value of the complexity and compatibility factors is equal to one, as reported in Table 4-2.

Table 4-2 Complexity (α) and compatibility (η) data for case one

$\alpha = \eta$	J1	J2	J3	J4	J5	J6	J7	J8
G1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
G2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
G3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
G4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Once the model was run, the optimal solution found by the solver includes the allocation of the following jobs in working groups as shown in the table below:

Table 4-3 Job allocations (Sub-section one)

GROUPS (WORKERS)	JOB ALLOCATIONS		
	Alloc #1	Alloc #2	Alloc #3
G1	-----	-----	-----
G2	J1	J4	J5
G3	J6	J8	
G4	J2	J3	J7

The next figure identifies the use (in time) of each of the groups or workstations. The graph has identified the maximum capacity of each group (e.g., capacity of group 1 is less than or equal to 1, $G1 \leq 1$). Group 2 has three job assignments with a total time of two hours, and the allocation for groups 3 and 4 has been identified in the same way. Group 1

was not assigned, and therefore, this station can be eliminated from the design of the organizational structure.

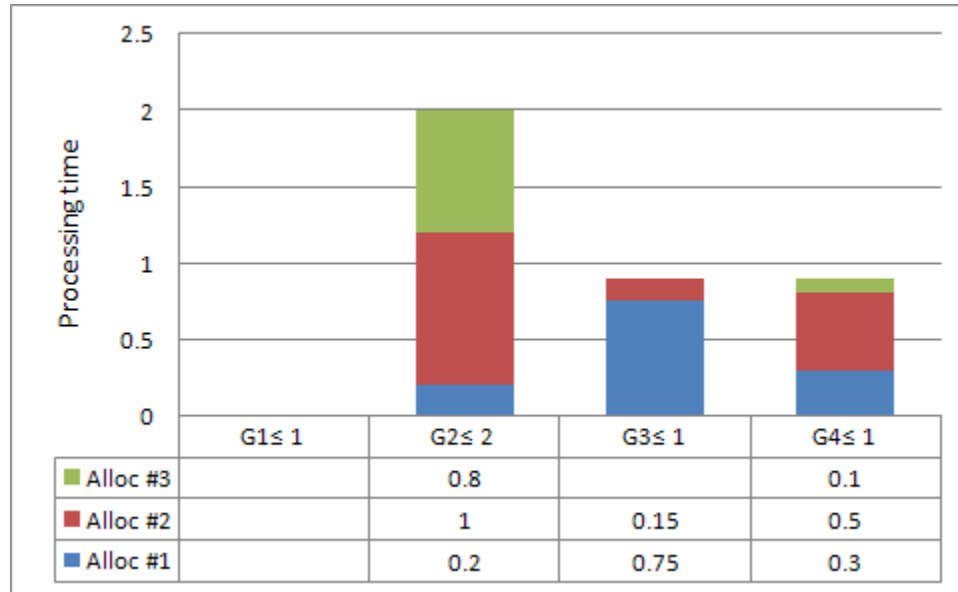


Figure 4-4 Graphical representation of results in Sub-section one

4.3.1.2 Sub-section two

This sub-section discusses the solution when the factor of complexity between jobs and workers was incorporated. To validate the model, the data for the factor shown on Table 4-4 were randomly assigned between jobs and groups. If the complexity of the jobs with regards to the groups was not irrelevant, then the problem changes according to the following data (the remaining data stay the same):

Table 4-4 Complexity (α) of jobs (J) in relation to workers (G)

GROUPS (workers)	JOBS							
	J1	J2	J3	J4	J5	J6	J7	J8
G1	0.60	1.20	0.80	1.30	0.60	1.10	1.30	1.50

G2	0.70	1.30	0.90	1.20	0.70	1.40	1.50	1.60
G3	0.80	1.50	1.00	1.40	0.80	1.60	1.40	1.10
G4	1.00	1.70	1.50	1.00	1.00	1.70	1.30	1.00

The previous table causes a change in the processing time required for each job with respect to each worker, updating the values according to the table below using the formula $a_{ij} = \beta_i * \alpha_{ij} / \eta_{ij}$.

Table 4-5 Processing time after complexity data

GROUPS	JOBS							
	J1	J2	J3	J4	J5	J6	J7	J8
G1	0.12	0.36	0.4	1.3	0.48	0.825	0.13	0.225
G2	0.14	0.39	0.45	1.2	0.56	1.05	0.15	0.24
G3	0.16	0.45	0.5	1.4	0.64	1.2	0.14	0.165
G4	0.2	0.51	0.75	1	0.8	1.275	0.13	0.15

The optimal solution obtained by the solver gave as a result the allocation of jobs to workers according to the next table.

Table 4-6 Job allocations (Sub-section two)

GROUPS (WORKERS)	JOB ALLOCATIONS			
	Alloc #1	Alloc #2	Alloc #3	Alloc #4
G1	J1	J3	J5	
G2	J2	J6	J7	J8
G3	-----	-----	-----	-----
G4	J1			

The chart below shows how the allocation of jobs at the workstations changed according to the complexity data. In this solution, group 1 has the assignment of three jobs, while

all assignments of group 3 were eliminated. As the data are analyzed, job one (J1), which was assigned in the previous results to group 2, has now changed its allocation to group 1. In the complexity table, job one has the lowest value with respect to that group.

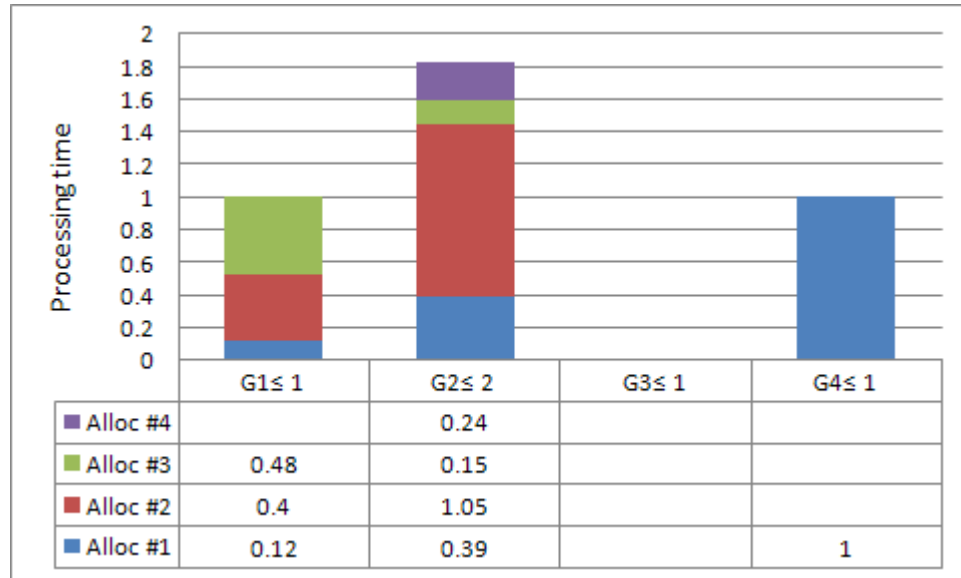


Figure 4-5 Graphical representation of results in Sub-section two

4.3.1.3 Sub-section three

This sub-section discusses the solution when the complexity and compatibility factors between jobs and workers were incorporated. To validate the model, the data for the compatibility factor were randomly assigned between jobs and groups (while complexity factor remains constant). If the complexity and the compatibility of the jobs with regards to the groups are not irrelevant, then the problem changes according to the following table (the remaining data stay the same):

Table 4-7 Compatibility (η) for workers (G) and jobs (J)

GROUPS	JOBS
--------	------

(workers)	J1	J2	J3	J4	J5	J6	J7	J8
G1	0.01	1.00	1.00	1.00	1.00	1.00	0.01	1.00
G2	1.00	0.01	1.00	1.00	1.00	0.01	1.00	0.01
G3	1.00	1.00	0.01	1.00	0.01	1.00	1.00	1.00
G4	1.00	1.00	1.00	0.01	1.00	1.00	1.00	1.00

The previous table causes a change in the processing time required for each job with respect to each worker, updating the values according to the table below using the formula $a_{ij} = \beta_i * \alpha_{ij} / \eta_{ij}$.

Table 4-8 Processing time after compatibility data

GROUPS	JOBS							
	J1	J2	J3	J4	J5	J6	J7	J8
G1	12	0.36	0.4	1.3	0.48	0.825	13	0.225
G2	0.14	39	0.45	1.2	0.56	105	0.15	24
G3	0.16	0.45	50	1.4	64	1.2	0.14	0.165
G4	0.2	0.51	0.75	100	0.8	1.275	0.13	0.15

The optimal solution obtained by the solver gave as a result the allocation of jobs to workers according to the following table:

Table 4-9 Job Allocations (Sub-section three)

GROUPS (WORKERS)	JOB ALLOCATIONS			
	Alloc #1	Alloc #2	Alloc #3	Alloc #4
G1	J6			

G2	J3	J4		
G3	J1	J2	J7	J8
G4	J5			

The figure below shows how the allocation of jobs at the workstations changed according to the complexity and the compatibility data. In this case, the model does not make any allocation of those jobs that don't have compatibility with the groups. For example, group 1 doesn't have jobs 1 and 7 assigned; group 2 doesn't have jobs 2, 6 and 8 assigned; group 3 doesn't have jobs 3 and 5 assigned; and group 4 doesn't have job 4 assigned due to the fact that they are not compatible. These constraints force the model to consider a longer processing time of jobs, and therefore, the solution to this problem involves 4 groups instead of 3.

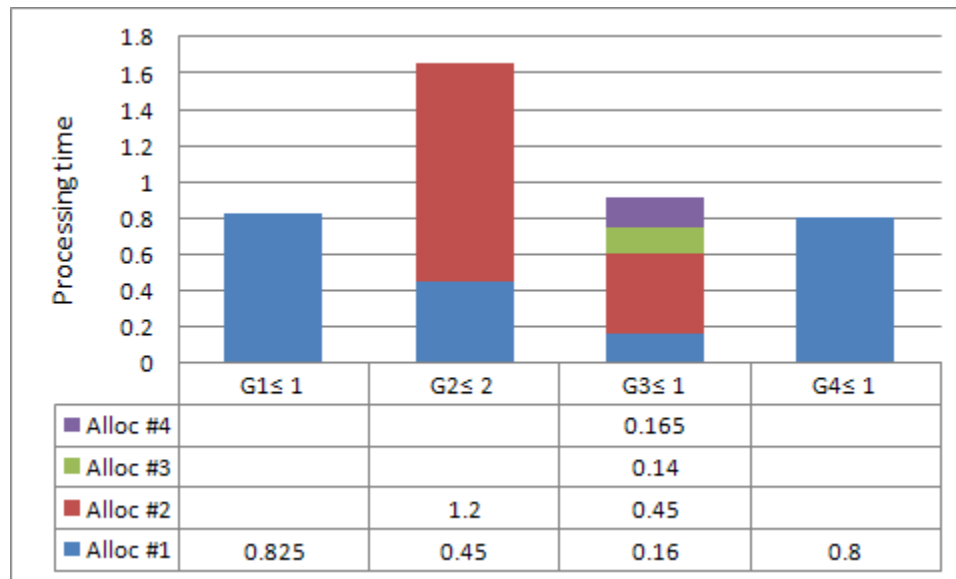


Figure 4-6 Graphical representation of results in Sub-section three

The conclusion in this section one shows that the model minimizes the number of workstations looking for jobs that have the best processing time, avoiding the selection of jobs without compatibility to the assigned station. The following table shows how the job allocations were being re-organized in the different groups according to the complexity and compatibility factors within the model. Also, the chart below shows how the utilization (processing time) of each of group is changing according to the conditions of each Sub-section. Capacity constraint of each group was maintained in all Sub-sections.

Table 4-10 Job Allocations (Sub-sections one, two and three)

Sub-sections	GROUPS (WORKERS)			
	G1	G2	G3	G4
1		J1-J4-J5	J6-J8	J2-J3-J7
2	J1-J3-J5	J2-J6-J7-J8		J1
3	J1-J7	J2-J6-J8	J3-J5	J4

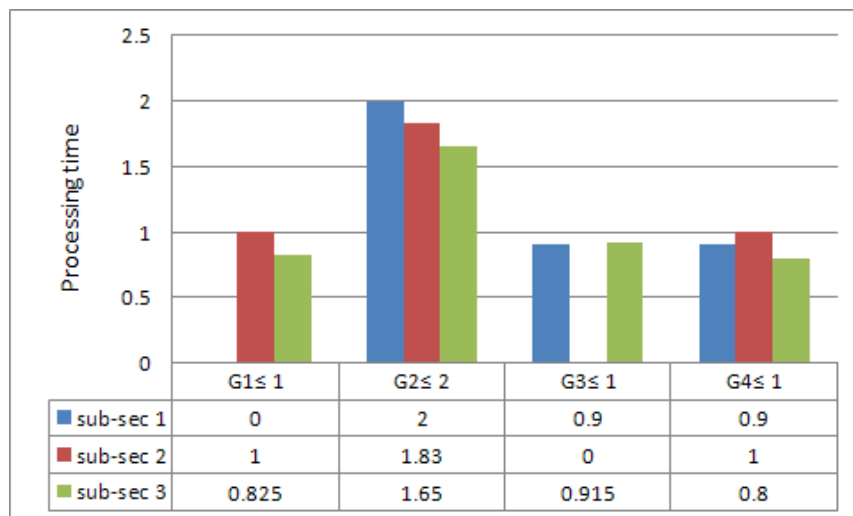


Figure 4-7 Utilization of each group en each Sub-section

4.3.2 Case one: Section two (Level 1 to n)

Once the validation of the model for level 0 (job allocations) was implemented, section two analyzes the impact of the factors involved in the supervision effort. This section analyzes the effect of changing the values of the different parameters for supervising. Taking the data and results of Sub-section two (from Section one), the results of level 0 were used to analyze the variation in the following levels of decision.

4.3.2.1 Sub-section one

The following table shows the capacity and cost of supervisors (level 1). This sub-section has been assigned a minimum value to the factors of supervision (close to zero).

Table 4-11 Data for Sub-section two (levels 0 → 1)

DATA OF SUPERVISORS		SUPERVISORS			
LEVEL 0 → 1		S1	S2	S3	
CAPACITY (Hrs)		3	2	1	
COST (w)		\$200	\$150	\$100	
FACTORS OF SUPERVISOR EFFORT		WORKERS (GROUPS)			
		G1	G2	G3	G4
PLANNING (ρ)		0.01	0.05	0.1	0.2
CONTROL (c)		0.01	0.05	0.1	0.2
TRAINING (t)		0.01	0.05	0.1	0.2
EXECUTION (e)		0.01	0.05	0.1	0.2

The following table shows the capacity and cost of managers (level 2). This Sub-section has been assigned a minimum value to the manager factors (close to zero).

Table 4-12 Data for Sub-section two (levels 1 → 2)

DATA OF MANAGERS		MANAGERS	
LEVEL 1 → 2		M1	M2
CAPACITY (Hrs)		3	3
COST (w)		\$450	\$500
FACTORS OF MANAGER EFFORT	SUPERVISORS		
	S1	S2	S3
PLANNING (ρ)	0.01	0.05	0.1
CONTROL (c)	0.01	0.05	0.1
TRAINING (t)	0.01	0.05	0.1
EXECUTION (e)	0.01	0.05	0.1

Once the allocation of jobs to working groups has been done (from Sub-section two of Section 1), supervisors and managers are selected taking into account the cost and capacity, as well as the planning, control, training, and execution factors of each.

Results of this Sub-section show that the values given for the supervisor factors, only requires the supervisor 3 with a minimum cost of \$100; for level 2 the solution requires only the manager 2 with a minimum cost of \$1350, giving a total minimum cost of \$1453.

Table 4-13 Supervisor allocations (Sub-section one)

SUPERVISORS (LEVEL 1)	GROUPS	COST	MANAGERS (LEVEL 2)	SUPERVISORS	COST
S1	-----	-----	M1	S3	\$1350
S2	-----	-----	M2	-----	
S3	G1-G2-G4	\$100	TOTAL COSTS		\$1453

4.3.2.2 Sub-section two

This Sub-section analyzes the behavior of the results of the model increasing (randomly) the requirements for each of the factors involved, as shown in the following table:

Table 4-14 Supervisor factors (levels 0→1)

GROUPS (workers)	SUPERVISOR FACTORS			
	PLANNING (p)	CONTROL (c)	TRAINING (t)	EXECUTION(e)
G1	0.1	0.1	0.1	0.1
G2	0.12	0.12	0.12	0.12
G3	0.15	0.15	0.15	0.15
G4	0.2	0.2	0.2	0.2

For the following level two, the values of the factors are also increased according to the table below:

Table 4-15 Manager factors (levels 1 → 2)

SUPER- VISORS	MANAGER FACTORS			
	PLANNING (p)	CONTROL (c)	TRAINING (t)	EXECUTION (e)
S1	0.1	0.1	0.1	0.1

S2	0.12	0.12	0.12	0.12
S3	0.15	0.15	0.15	0.15

Results of this Sub-section show that increasing the values of supervisor factors also increases the requirements of supervision hours. Therefore the decision changes to supervisor 2, who has a higher capacity, while also increasing the cost of this level (1). The result for level two remains the same and the total cost increases, as shown in the following table:

Table 4-16 Supervisor allocations (Sub-section two)

SUPERVISORS (LEVEL 1)	GROUPS	COST	MANAGERS (LEVEL 2)	SUPERVISORS	COST
S1	-----	-----	M1	S3	\$1350
S2	G1-G2-G4	\$300	M2	-----	
S3	-----	-----	TOTAL COSTS		\$1653

4.3.2.3 Sub-section three

Sub-section three explores the impact of assigning the maximum value of the supervisor factors. The maximum value assigned to each of the factors is 0.25, which implies that the sum of the factors is equal to one, and therefore, one hour of supervision is required for each hour assigned to the worker. The values of the factors for this sub-section appear in the following tables:

Table 4-17 Supervisor factors (levels 0→1)

GROUPS (workers)	SUPERVISOR FACTORS			
	PLANNING (p)	CONTROL (c)	TRAINING (t)	EXECUTION(e)

G1	0.25	0.25	0.25	0.25
G2	0.25	0.25	0.25	0.25
G3	0.25	0.25	0.25	0.25
G4	0.25	0.25	0.25	0.25

For the following level two, the value or factors was also assigned in such a way that one hour of supervision requires one hour from the manager at the next hierarchical level, as shown in the table below:

Table 4-18 Manager factors (levels 1 → 2)

SUPER-VISORS	MANAGERS FACTORS			
	PLANNING (p)	CONTROL (c)	TRAINING (t)	EXECUTION (e)
S1	0.25	0.25	0.25	0.25
S2	0.25	0.25	0.25	0.25
S3	0.25	0.25	0.25	0.25

Results of this Sub-section show that increasing the values of supervisor factors (sum equals to one), also increases the requirements of supervision hours. Therefore the decision changes to supervisor 2 and 3 increasing the cost of this level (1). The result for level two includes both managers and the total cost increases, as shown in the following table:

Table 4-19 Supervisor allocations (Sub-section three)

SUPERVISORS (LEVEL 1)	GROUPS	COST	MANAGERS (LEVEL 2)	SUPERVISORS	COST
S1	-----	-----	M1	S2	\$1350
S2	G2-G4	\$300	M2	S3	\$550

S3	G1	\$100	TOTAL COST	\$2303
----	----	-------	------------	--------

In this Section two, increasing the planning, execution, control, or training requirements, also increases the time required for each supervisor, increasing as a result, the total cost. The data of cost (w) and capacities (f) of supervisors and managers have remained constant in order to know how they influence the values of the factors mentioned before.

4.3.3 Limit size problem for a commercial solver (Case two)

For this research, it is important to know the size of the problem that the commercial solver could handle, in order to know what size of problems is feasible to get an optimal solution. For bigger problems, a heuristic method that is able to get a good quality solution is required. Therefore, the main objective in this Case two is to determine the number of constraints and variables that the commercial solver can handle. After several experiments with LINGO, it is concluded that the software can handle problems with up to 30 jobs and 10 groups. In this case, supervising parameters were selected close to zero and then close to one to review how the allocation of supervisory requirements was affected. Data of capacity, time, complexity, and compatibility factors for a problem instance of 30 jobs and 10 groups at level 0 is found in Appendix A (this data is the same for both Sections).

4.3.3.1 Case two, Section one

The following table shows the capacity and cost of supervisors (level 1):

Table 4-20 Capacity (α) and cost (α) data for levels 0 \rightarrow 1

DATA OF SUPERVISORS	SUPERVISORS
---------------------	-------------

LEVEL 0 → 1	S1	S2	S3	S4	S5
CAPACITY (f) (Hrs)	4	4	4	8	8
COST (w)	\$50	\$100	\$150	\$200	\$250

This sub-section has been assigned a minimum value to the factors of supervision (close to zero) as in the following table:

Table 4-21 Supervisor factors (Case two)

GROUPS	SUPERVISOR FACTORS			
	PLANNING (p)	CONTROL (c)	TRAINING (t)	EXECUTION (e)
G1	0.01	0.01	0.02	0.03
G2	0.001	0.004	0.003	0.02
G3	0.01	0.01	0.02	0.03
G4	0.005	0.06	0.02	0.01
G5	0.01	0.01	0.02	0.03
G6	0.001	0.004	0.003	0.02
G7	0.01	0.01	0.02	0.03
G8	0.001	0.004	0.003	0.02
G9	0.01	0.01	0.02	0.03
G10	0.001	0.004	0.003	0.02

The following table shows the capacity and cost of managers (level 2):

Table 4-22 Capacity (α) and cost (α) data for levels 1 → 2

DATA OF MANAGERS	MANAGERS		
LEVEL 1 → 2	M1	M2	M3
CAPACITY (f) (Hrs)	4	8	8

COST (w)	\$100	\$200	\$30
----------	-------	-------	------

For the following level two, the values of the factors are according to the table below:

Table 4-23 Manager factors (Case two)

SUPE- VISORS	MANAGER FACTORS			
	PLANNING (p)	CONTROL (c)	TRAINING (t)	EXECUTION (e)
S1	0.002	0.003	0.01	0.02
S2	0.002	0.003	0.01	0.02
S3	0.002	0.003	0.01	0.02
S4	0.002	0.003	0.01	0.02
S5	0.002	0.003	0.01	0.02

Results for job allocations for this case are shown in table below:

Table 4-24 Job allocations (Case two)

GROUPS	G1	G3	G4
JOBS	J3-J4-J5-J6-J8-J10-J13- J16-J18-J24-J25-J28	J1-J2-J9-J12-J20- J22-J27-J17	J7-J11-J14-J15-J19- J21-J23-J26-J29-J30

Results for level 1 and 2, regarding the assignment of groups to supervisors and supervisors to managers are in the following table. This sub-section only requires one supervisor (S1) and one manager (M1) with a total cost of \$603.

Table 4-25 Supervisor and manager allocations (Section one)

SUPERVISORS (LEVEL 1)	GROUPS	COST	MANAGERS (LEVEL 2)	SUPERVISORS	COST
S1	G1-G2-G3	\$200	M1	S1	\$400

S2	-----	-----	M2	-----	-----
S3	-----	-----	M3	-----	-----
S4	-----	-----	TOTAL COST		\$603
S5	-----	-----			

4.3.3.2 Case two, Section two

This section two explores the impact of assigning the maximum value of the supervisor factors. The maximum value assigned to each of the factors is 0.25, which implies that the sum of the factors is equal to one, and therefore, one hour of supervision is required for each hour assigned to the worker. The values of the supervisor factors for this Section appear in the following table:

Table 4-26 Supervisor factors (Section two)

GROUPS	SUPERVISOR FACTORS			
	PLANNING (p)	CONTROL (c)	TRAINING (t)	EXECUTION (e)
G1	0.1	0.2	0.2	0.1
G2	0.2	0.2	0.3	0.1
G3	0.1	0.2	0.2	0.1
G4	0.2	0.2	0.3	0.1
G5	0.1	0.2	0.2	0.1
G6	0.2	0.2	0.3	0.1
G7	0.1	0.2	0.2	0.1
G8	0.2	0.2	0.3	0.1
G9	0.1	0.2	0.2	0.1
G10	0.2	0.2	0.3	0.1

The values of the manager factors for this Section appear in the following table:

Table 4-27 Manager factors (Section two)

SUPER-VISORS	MANAGER FACTORS			
	PLANNING (p)	CONTROL (c)	TRAINING (t)	EXECUTION (e)
S1	0.1	0.2	0.2	0.1
S2	0.2	0.2	0.3	0.1
S3	0.1	0.2	0.2	0.1
S4	0.2	0.2	0.3	0.1
S5	0.2	0.2	0.3	0.1

The number of groups remains constant since there were no changes in the complexity and compatibility of jobs with respect to workers (groups). Increases in the requirements of planning, control, execution, and training resulted in an increase in the number of supervisors (1 to 2) and for the next level in the selection of a manager with higher capacity. Results for level 1 and 2, regarding the assignment of workers to supervisors and supervisors to managers are in the following table. This Section requires two supervisors (S1 and S2) and one manager (M2) with a total cost of \$2203, increasing the total cost by 230%.

Table 4-28 Supervisor and manager allocations (Section two)

SUPERVISORS (LEVEL 1)	GROUPS	COST	MANAGERS (LEVEL 2)	SUPERVISORS	COST
S1	G1	\$200	M1	-----	
S2	G2-G3	\$400	M2	S1,S2	\$1600
S3	-----	-----	M3	----	
S4	-----	-----	TOTAL COST		\$2203

S5	-----	-----		
----	-------	-------	--	--

4.4 Conclusions

Analysis has been carried out at least in 40 separate cases in order to ensure that allocations occur in a logical way as the data were fed into the system. The number of jobs, groups, supervisors, and managers had a significant impact on the computational cost because the number of variables had an exponential growth. In this case, the combination of jobs with groups had the highest impact. Increasing the number of jobs and groups after 30 jobs and 10 groups dramatically increases computational requirements.

The allocations made by the model with LINGO indicate that it works by making the correct allocation of activities taking into account the complexity and compatibility of the jobs with respect to each group for level 0. For levels 1 and 2, supervisors and managers were selected in order to have the capacity to monitor and manage all the working groups, and it is also optimal for carrying out such activities considering factors of planning, control, training, and execution. In these cases, the model makes the assignment based on capacity and cost of supervision and management.

Conclusions regarding the cases report that the size of span is greater in the lower level, but smaller in the higher level. The more complex the subordinates' tasks (associated with complexity factors), the smaller the supervisor's span of control. The more complex the supervisor's tasks (associated with supervisor factors), the narrower the span of control. Solving these cases concludes that the model is functional and takes into account all the constraints that were designed for the model.

5. VALUATION OF COEFFICIENTS AFFECTING MODELING FACTORS

The objective of this chapter is to determine a method to quantify the factors that determine the allocation of jobs to workers (level 0), and those related with supervising time, divided into the following factors: planning, control, execution, and training. In order to do this, the level of importance between the factors has to be determined, and a quantitative range has to be identified with which it is possible to make an objective measurement that can be incorporated in solving the mathematical model proposed. This part of the research is conducted through the study of the activities of people who have supervisory duties. For this purpose, data were collected from four different Information System (IS) companies in the city of Chihuahua, Mexico.

5.1 Complexity and compatibility factors (level 0)

The objectives pursued with the following method are based in two points: (1) determining their existing workers' OP-types and skill levels, and (2) assigning the workers to their complexity and compatibility factors at an appropriate level regarding the jobs. The method for assessment of the complexity and compatibility factors at level zero is based on the information of a job proposed to the company which consists of the jobs to be completed, the OP-Type, and the level of skill (required to complete each job). The theoretical foundation for this method is based on the analysis of workers' skills, and makes a comparison with respect to the skills required for a particular job. The following figures show the steps needed for this process. The proposed model is partially built on the one developed by Depuy et al. [29].

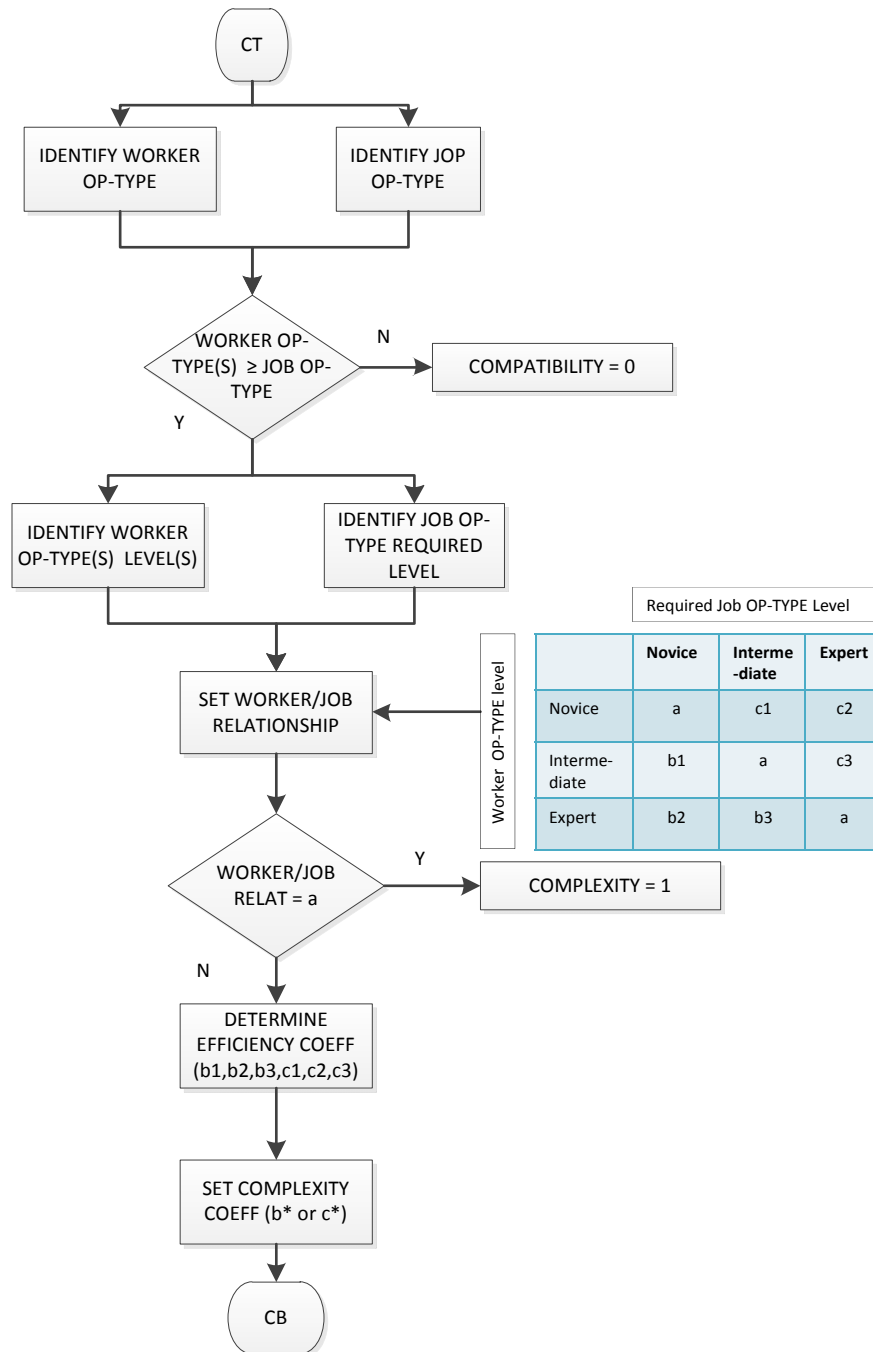


Figure 5-1 Method for complexity factor

On the left side (worker), the first step is to identify the skills (OP-Types) of the worker. Once the different skills of each worker have been established, the next step is to determine, based on technical tests, the level of each skill. On the right side (job), the OP-

Type and required skills of each job are defined. Next step is to determine the required skill level of each job. In order to assign a skill relationship between jobs and workers, it is necessary to design a skill relationship table. For this table, the first step is to determine how many levels are required (e.g., novice, intermediate, expert), and the numerical relationship between the different combinations (e.g., novice-novice, novice-intermediate, novice-expert).

As an example, consider a project with a total number of three jobs with three required skills identified, and each skill is further defined by a skill level ranging from 1- novice, 2- intermediate, and 3- expert. Some jobs can require the same skill, but at different levels. Multiple jobs can be assigned to the same employee, subject to a capacity constraint. The figure below shows the following situation: worker 1 (group G1) has the skills 1 and 3, with skill levels of 1 and 2, such that, if assigned to job 1, there would be the same skills relationship, and therefore, the complexity factor would be 1. If however, that same employee were assigned to job 2, there would be a skills relationship of “intermediate-expert”, and therefore, according to Table 4-1, the complexity factor would be 1.5. If worker 1 were assigned to job 3, there would not be a skills relationship, and therefore, the compatibility factor would be close to zero.

Once the relationship between workers and jobs has been established, the complexity factor is determined based on Table 5-1.

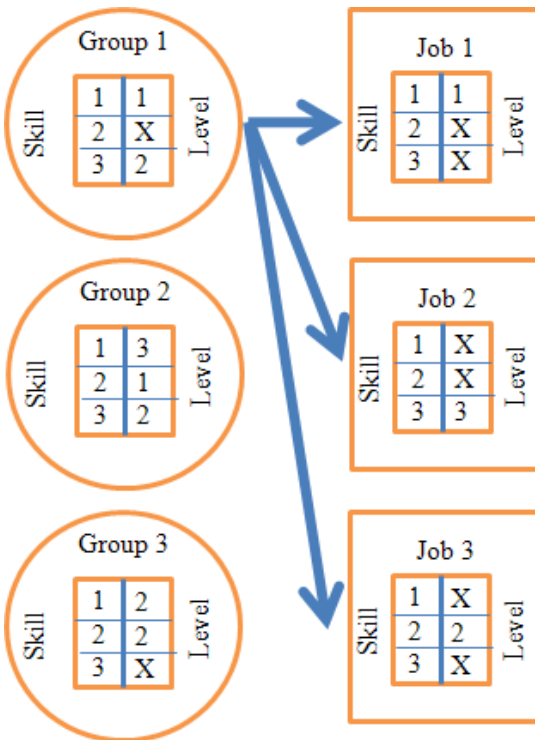


Figure 5-2 Example of skills and skill levels between workers and jobs

Table 5-1 Complexity factors

		Req Job Skill		
Station Skill	Station/Job	Novice (1)	Inter (2)	Expert (3)
	Novice (1)	1	1.5	2
	Inter (2)	0.8	1	1.5
	Expert (3)	0.6	0.8	1

The previous table is based on the relationship found by analyzing 20 individual events in the companies selected for this study. This general format can be applied to any industry, as long as the values assigned to each of the relationships (e.g., novice-intermediate equals 1.5) are reviewed. It is clear that the weights that are equal to one are the same for

any situation, where the same worker skill level against the same job skill level required is compared. For the rest of the relationships, there must be an adjustment made depending on the situations in which this format will be used. Therefore, a relationship of a worker skill level "expert" for a job skill level requirement of "novice" is set for the worker to require only a 60% of standard time to do such work. In a different situation, this percentage may be lower (30-50%) or higher (70-90%).

The compatibility factor, as shown in the following figure, verifies the compatibility and then assigns a value to this factor updating the relationship between a job and a worker.

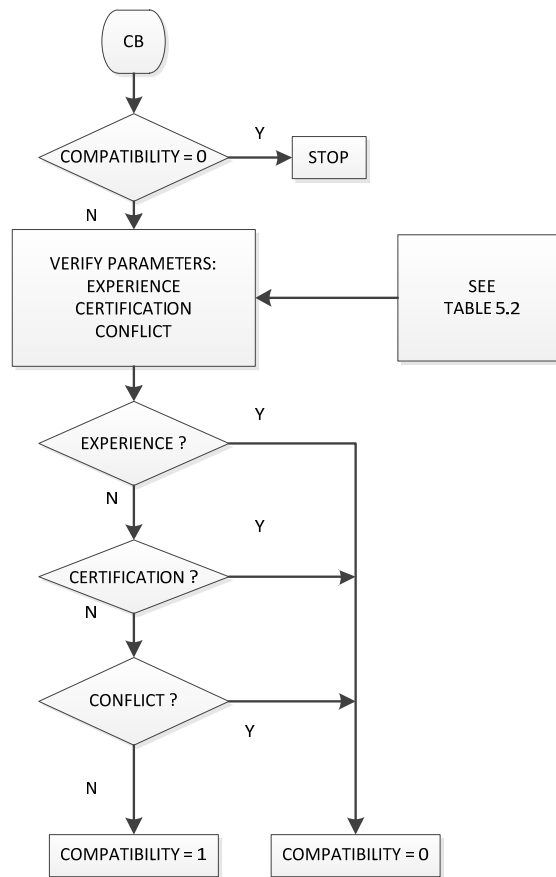


Figure 5-3 Method for compatibility factor

In the table below are some questions that must be made regarding the compatibility factor between worker and job assignments.

Table 5-2 Compatibility questions

Compatibility	Questions	Worker/Job
Experience	<p>Are there any issues regarding the experience of the worker related with a particular job?</p> <p>Is there any requirement in relation to the experience that the worker must have to perform a particular job?</p>	YES/NO
Certification	<p>Is there any certification process that the worker must meet in order to be able to execute any particular job?</p> <p>Is there any academic degree that the worker must meet in order to execute any particular job?</p> <p>Is there a specific license (e.g., from government) in order to execute a particular job?</p>	YES/NO
Conflicts	<p>Are there any conflicts that the worker has with the company/supervisor in order to assign jobs?</p> <p>Does the worker have any kind of conflict in order to execute a particular job?</p>	YES/NO

Compatibility values, in this case, consider only close to 0 and 1, in such a way that the model assigns jobs to workers only when compatibility is equal to one. When compatibility is close to zero, the model assigns a very large number for the processing time of that activity and does not assign that activity to that individual worker.

The no-compatibility factor is used when a worker does not have the appropriate level to perform a job. For example, assume that it is not appropriate to assign a worker with a

skill level "novice", who requires double time for a job with an expert level requirement (200% according to Table 4-1), to that job. This “novice-expert” relationship could be eliminated through the compatibility factor. In some scenarios it may also be forbidden to assign jobs that require minimum skill level to expert workers; in such case the use of compatibility factor ensures the type of assignment needed for the job. At other times, even when an operator can perform the required work changing their standard time, they may not have a proper certification to do the job or they may not have the necessary tools at their workplace. Therefore a value of non-compliance with respect to that job must be assigned. If the solution of the model, with the compatibility values that were assigned, is not feasible, then it is possible to analyze the causes of worker compatibility in order to have higher flexibility in finding a workable solution. The result of this method is the development of a model that provides an assignment of workers to jobs, such that the total skills gap is minimized. The complexity and compatibility factors between the three groups and jobs (of Figure 4-1) are found in table below:

Table 5-3 Complexity and compatibility factors

Complexity				Compatibility			
α	J1	J2	J3	η	J1	J2	J3
G1	1	1.5	X	G1	1	1	0.01
G2	0.6	1.5	1.5	G2	1	1	1
G3	0.8	X	1	G3	1	0.01	1

5.2 Primary dimensions of supervision work (level 1 to n)

In this section, the objective is to construct an instrument for providing a measure of organizational factors (planning, execution, training, and control). This develops a more extensive model that assigns workers to supervisors, in such a way that the total cost

required achieving all the jobs is minimized. From examination of the literature on Organization Theory, four primary dimensions of supervision work were defined: (1) planning, which is concerned with process, product standardization in the organization, initiation, preparation, resource allocation, and scheduling; (2) control, which is concerned with the extent to which good project management control is employed, such as monitoring, coordinating, decision making, dissemination, dispatching, and performance evaluation; (3) training, which is concerned with effective training tools such as lecturing, coaching, guiding, and discussing in the organization; and (4) execution, which is concerned mainly with the alignment of organization objectives, problem solving and implementation. These definitions are going to be translated into operational definitions, and developing the methods and results of scaling the supervisor work variables.

The theoretical foundation for this method is based on the analysis of the relevant activities related to each factor, and the identification of instruments for measure. This method is partially based on the work of Khaled et al. [58] for measuring organizational maturity. The following figure shows the general steps:

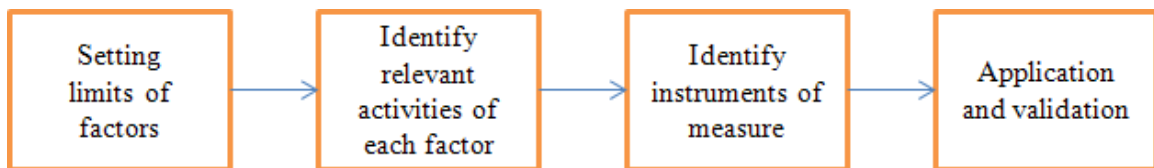


Figure 5-4 Method for supervisor factors

5.2.1 Setting limits for the factors

In the model, if the sum of the factors (constraints 3.2.4, 3.2.6) is equal to one, then that implies that supervision at level 1 requires the same quantity of time than the worker at level 0. On the other hand, if the sum is less than one, this implies that the supervision time required is less than the operating time of the group; therefore, a supervisor can have more than one subordinate as shown in the following figure.

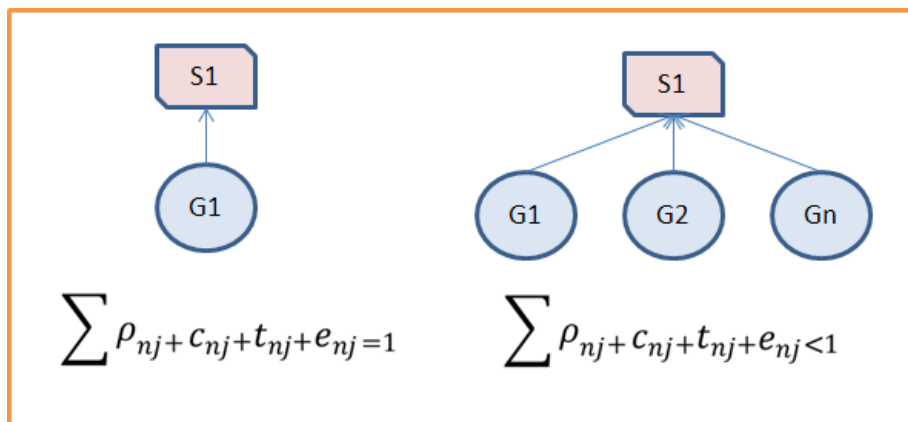


Figure 5-5 Impact of sum of factors in span of control

As shown in Figure 5-6, a method was developed to identify the boundary of each of the factors within the constraints indicated.

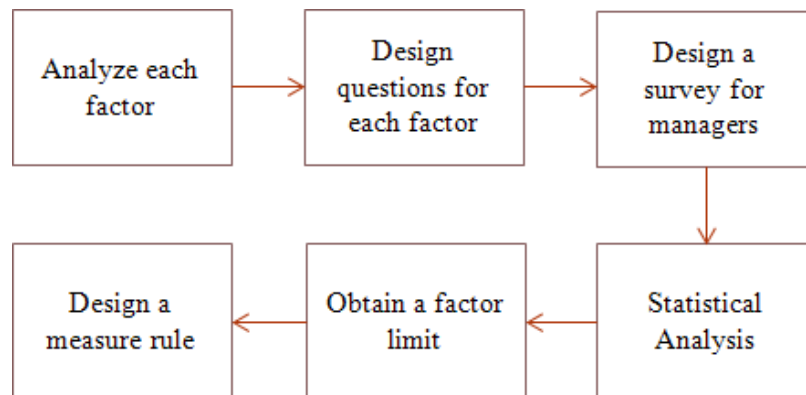


Figure 5-6 Factors between level 0 → 1

The first step in the previous model includes the analysis of each factor according to the type of industry. Planning, control, execution, and training could have different meaning in different types of industries. In order to know the importance level of each factor within the company, a survey with different questions regarding each factor was designed. The survey was applied to all supervisors and managers in the company and the results were analyzed using statistical software (in order to obtain mean, standard deviation, and relevant parameters). If for a particular type of business, the importance of each factor was the same, then the limit would be 0.25 for each (in such a way that the sum of the four factors equals one: $0.25 + 0.25 + 0.25 + 0.25$). Otherwise, applying linearity to the results, a factor limit is obtained. Once the factor limit is known, a measure rule could be designed adjusting the minimum and maximum number of subordinates, in such a way that the rule is according to the span of control for each particular industry. In order to show how this process works, data were collected in four IS companies and 30 supervisors (within these companies) in the city of Chihuahua, Mexico. The results are discussed in section 5.2.4.

The next figure shows how to design a measure rule: once the survey that establishes the level of importance of each of the supervisor factors has been designed and applied, the next step is to get the mean (a_1, a_2, a_3, a_4) and standard deviation of each factor. The next step is to get the sum of the means values (a_t), and assuming that the sum of the coefficients must be less or equals to one, applying linearity the coefficients limits could be obtained (equals a^*/a_t). Each of the instruments, in order to valuate supervision effort, used in this process, has an evaluation from one (expert), two (intermediate), and three

(novice). The coefficient limit at level three (novice) will be the mean value of each factor. In order to get the measure rule, the number of steps must be established and from there the measure rule could be designed dividing the coefficient limit by the number of steps with regular increments (as shown in figure).

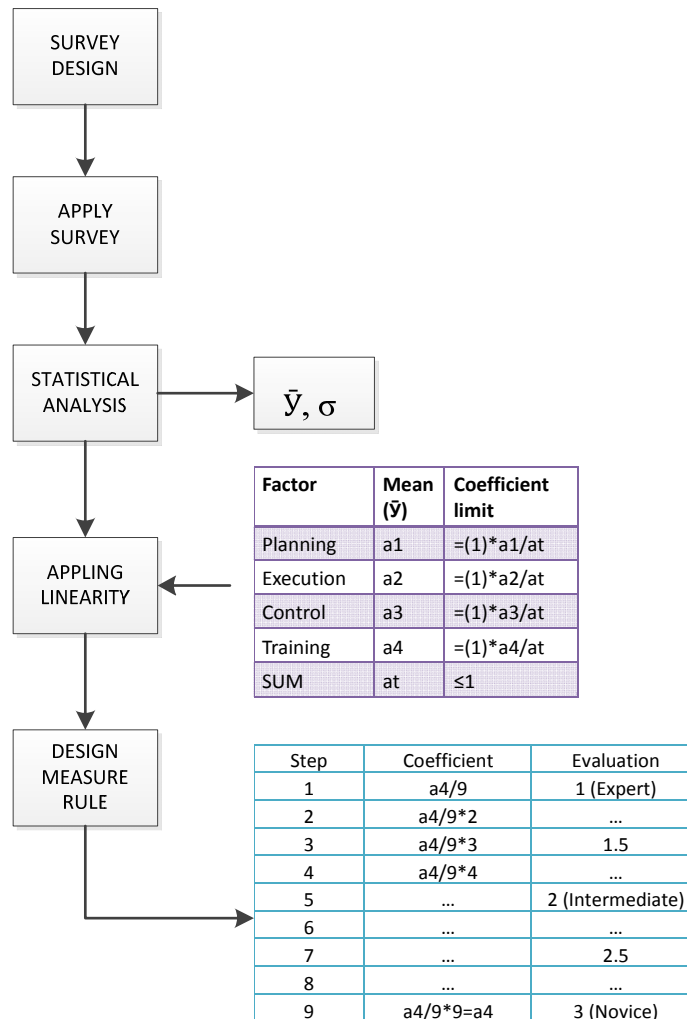


Figure 5-7 Design of measure rule

5.2.2 Identification of relevant activities

Once the coefficients limits of each factor have been identified, it is necessary to establish the relevant activities for each factor in order to identify possible instruments to measure the required time of supervision effort. From definitions found on the work of Iyigün [50], Raishi [51], Annand et al. [52], and Hales [55] the principal activities for the planning, control, training, and execution factors were defined. The following table shows these activities:

Tabla 5-4 Relevant activities in supervisor factors

Planning	<ol style="list-style-type: none"> 1. Plan development: where the processes to implement, monitor and supervise are defined, prepared, coordinated, and integrated. 2. Scope planning: defines the project based on who will be responsible for carrying out the duties. 3. Scope definition and activities: develops a detailed statement of the scope of the project as a basis for future decision making, and the process to identify the activities to be done to develop the deliverables. 4. Development of schedules: establishes the sequence of activities, their duration, the responsible personnel, requirements, resources, and constraints of the project. 5. Estimated cost of the resources required for each activity. 6. Quality planning: identifies the quality standards that are appropriate and relevant. 7. Human resource planning: identifies the roles, responsibilities,
----------	--

	and reporting relationships based on the information obtained.
Control	<ol style="list-style-type: none"> 1. Perform activities to meet project objectives. 2. Make efforts and invest funds to meet project objectives. 3. Obtain, manage, and use resources including materials, tools, equipment, and facilities. 4. Implement the planned methods and standards. 5. Create, monitor, verify, and validate project deliverables. 6. Define the risks and implement risk response activities. 7. Adapt approved changes to the scope, plans, and project environment. 8. Establish project communication channels, both external and internal to the team. 9. Collect data and report on project cost, schedule, quality, and technical progress and status information to facilitate projections.
Training	<ol style="list-style-type: none"> 1. To improve the performance of a job using new technologies, policies or procedures carried out in the organization. 2. To facilitate (or teach) courses, workshops, conferences or any activities that facilitate new knowledge, techniques or procedures. 3. To identify skills of workers related to the skills needed in the company.

Execution	<ol style="list-style-type: none"> 1. Effective communication: related to the exchange of information among members of the organization. 2. Influence within the organization: refers to the ability to "get things done". 3. Leadership: to develop a vision and strategy, and motivating people to achieve that vision and strategy in its own right. 4. Encourage people to achieve high levels of performance and overcome obstacles in changing the organizational environment. 5. Negotiation and conflict management: to consult with others to agree or reach agreement with each team member or department. 6. Problem solving: identification and analysis of alternatives and decision making.
-----------	---

The next steps in this process, after the activities were identified for each factor, are focused on the following processes:

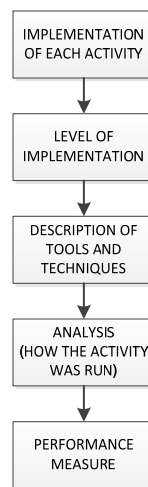


Figure 5-8 Activity implementation process

5.2.3 Identification of instruments of measure

This section provides the procedure of the field test used, and a brief description of how variables were measured, the data analysis procedures, and discussion of the results of the study. The applications were validated with data collected within the context of four different Information Systems (IS) companies based in the city of Chihuahua, México. Supervisors, who might be a work manager, an area superintendent, or some other administrator, were first contacted by a letter addressed to the general manager of each IS company. Field work began with interviewing them at length, and interviews were conducted with standard schedules listing the data desired. This instrument is used as part of the software process diagnosis and improvement services that the IS companies provided.

The study was specifically conducted to confirm the patterns found in the literature with the organizational practice and to collect data that are useful for measuring the factors of planning, control, training, and execution. The thirty people (from four IS companies) who participated in this first process were selected based on the functions they perform in the workplace in supervising activities. The following table summarizes the instruments for measure in each of the factors:

Table 5-5 Instruments for measure of each factor

Factor	Instruments
Planning	SWOT Matrix (Strengths, Weaknesses, Opportunities, Threats) (see Appendix B) Standardization

	<p>Project action plan (see Appendix B)</p> <p>Description or job profile (see Appendix B)</p>
Control	<p>Meeting report</p> <p>Support tools:</p> <ol style="list-style-type: none"> 1. Charts 2. Schedules 3. Time Reports 4. Efficiency Reports
Execution	<p>Performance (see Appendix B)</p> <p>Interviews and tests</p> <p>MBO (Analysis of job performance) (see Appendix B)</p> <p>Checklist</p> <p>Performance in relation to work (see Appendix B)</p> <p>Checklist</p>
Training	<p>Evaluation center</p> <p>Practice tests</p>

The formats specified are a compilation of the tools most commonly used in IS companies. The hypothesis at this point assumes that a supervisor properly using the tools necessary for the activities identified in each factor, requires less time to monitor these activities. The basic idea is to identify the tools that a supervisor knows and uses to carry out their work more efficiently. To the extent that a supervisor knows and uses the tools,

the time required for each one of the activities identified in supervising factors will be lower. The following instruments were identified to evaluate the planning factor:

Table 5-6 Instruments used for the planning factor

Instrument #1:	SWOT Matrix
Objective	Analyze the situation where the enterprise is found, stay on top of events, and anticipate situations in order to reduce threats and seize opportunities
Instructions	<ol style="list-style-type: none"> 1. Make a list of external opportunities 2. List the external threats 3. List the internal strengths 4. Make a list of internal weaknesses 5. Compare the strengths with the opportunities and plan strategies 6. Compare the weaknesses with the opportunities and plan strategies 7. Compare the strengths of threats and propose strategies 8. Compare the weaknesses with threats and propose strategies
Measure	<ol style="list-style-type: none"> 1. Tool is known and used 2. Tool is known but not used 3. Tool is not used or known
Instrument #2	Standardization
Objective	Establish criteria and rules based on business requirements to determine if there is progress or setbacks in the organization
Instructions	<ol style="list-style-type: none"> 1. Check if there is a ISO format applicable to the company

	<p>processes</p> <ol style="list-style-type: none"> 2. Study and analyze the format (i.e. ISO 9000) 3. Implement formats 4. Audit process
Measure	<ol style="list-style-type: none"> 1. Tool is known and used 2. Tool is known but not used 3. Tool is not used or known
Instrument #3	Description of job profile
Objective	Integrating the elements necessary to know the true and proven ability of an employee to carry out an activity or task correctly
Instructions	<ol style="list-style-type: none"> 1. Build a list indicating where the actions will be performed with by persons assigned to the post 2. Establish requirements that employees must follow 3. Conduct a competency analysis 4. Specify performance criteria
Measure	<ol style="list-style-type: none"> 1. Use all instruments to measure the experience 2. Use only one instrument 3. There are no instruments to measure the experience
Format	See Appendix B

The following instruments were identified to evaluate the execution factor:

Table 5-7 Instruments used for the execution factor

Instrument #1	Interviews and tests
Objective	Measure the profile of leadership, abilities and skills of employees regarding work performance in relation to the work required
Instructions	<ol style="list-style-type: none"> 1. Depth interviews are carried out by the head of the area. 2. Diagnosis of personal skills
Measure	<ol style="list-style-type: none"> 1. Tool is known and used 2. Tool is known but not used 3. Tool is not used or known
Instrument #2	Performance
Objective	Evaluate and compare the performance of each employee
Instructions	<ol style="list-style-type: none"> 1. Comparative method It consists of making a list of personnel to evaluate and compare each employee with each member of the group and determine who has better performance 2. Scales method The worker is assessed on a scale that is determined in advance to evaluate each factor based on maximum and minimum levels
Measure	<ol style="list-style-type: none"> 1. Tool is known and used 2. Tool is known but not used 3. Tool is not used or known

Format	See Appendix B
Instrument #3	MBO (Management by Objectives)
Objective:	Analysis of job performance
Instructions	<ol style="list-style-type: none"> 1. Set the job's responsibilities, project or assignment 2. Set a time period to implement the actions set 3. Conduct the assessment 4. Establish new goals
Measure	<ol style="list-style-type: none"> 1. Tool is known and used 2. Tool is known but not used 3. Tool is not used or known
Format	See Appendix B
Instrument #4	Checklist
Objective	Measure performance in relation to work performed
Instructions	Assign a range between 25 to 100 with a different value, then the points are added, scored, and compared with other employees to determine the effectiveness of each
Measure	<ol style="list-style-type: none"> 1. Tool is known and used 2. Tool is known but not used 3. Tool is not used or known

The following instruments were identified to evaluate the training factor:

Table 5-8 Instruments used for the training factor

Instrument #1	Ability
Objective	Measure abilities and skills of employees regarding work performance in relation to the work required
Instructions	<ol style="list-style-type: none"> 1. Evaluation center This tool shows the behavior of candidates in various exercises designed for a particular purpose 2. Practice tests They are used for technical or specialized positions in which they perform a specific task sample as close to the reality to be resolved
Measure	<ol style="list-style-type: none"> 1. Tool is known and used 2. Tool is known but not used 3. Tool is not used or known

The following instruments were identified to evaluate the control factor:

Table 5-9 Instruments used for the control factor

Instrument #1	Meeting report
Objective	Document the information of department meetings
Instructions	<ol style="list-style-type: none"> 1. Must contain the date of meeting 2. The topic

	<p>3. It should include people who attended the meeting</p> <p>4. Highlight the agreements or points that remained outstanding</p>
Measure:	<p>1. Tool is known and used</p> <p>2. Tool is known but not used</p> <p>3. Tool is not used or known</p>
Instrument #2	Support tools
Objective	Assess performance in relation to work performed and exercise corrective action
Instructions	<p>Support tools:</p> <p>1. Charts</p> <p>2. Schedules</p> <p>3. Time Reports</p> <p>4. Efficiency Reports</p>
Measure:	<p>1. Tool is known and used</p> <p>2. Tool is known but not used</p> <p>3. Tool is not used or known</p>
Format*:	See Appendix B

*Formats used for the evaluation of these instruments are shown in Appendix B.

5.2.4 Application and validation

In order to apply and validate this method, a sample of 30 supervisors and managers of the Information Systems (IS) companies were given a survey to determine the importance assigned to each of the factors within that particular organization type. The use of a semantic differential scale [53] for all the items in the survey was proposed.

The first part of the validation consists in setting the limit for the factors involved, and an analysis of the survey instrument employed was made using the statistical software SPSS version 17 for Windows Vista. The interview consisted of twenty questions (see Appendix B), five of them related to each factor in random order and on a scale of 1 to 5 (1. Strongly disagree, 2. Disagree, 3. Neither agree nor disagree, 4. Agree, 5. Strongly agree) that the supervisor has with respect to each question. The survey gave the following results:

Table 5-10 Level of importance of primary dimensions of supervision work

Factor	Mean	Std Dev	Var
Planning	3.7	1.2	1.26
Question 1	4.33	1.093	1.195
Question 5	3.8	0.961	0.929
Question 12	3.9	1.213	1.472
Question 13	3.2	1.183	1.407
Question 18	3.26	1.41	1.306
Control	3.54	1.22	1.5
Question 2	3.4	1.27	1.628
Question 3	4.06	1.015	1.030
Question 6	3.06	1.311	1.72
Question 10	3.53	1.332	1.775
Question 14	3.63	1.159	1.344
Execution	3.62	1.17	1.40
Question 4	4.066	1.172	1.375

Question 8	3.3	1.512	2.286
Question 9	3.1	1.062	1.128
Question 16	3.66	0.994	0.989
Question 20	3.96	1.098	1.206
Training	3.24	1.17	1.37
Question 7	3.93	1.093	1.195
Question 11	3.66	0.961	0.929
Question 15	3.86	1.213	1.472
Question 17	1.9	1.183	1.407
Question 19	2.83	1.41	1.306
Scale 1-5 (1.Strongly disagree, 2. Disagree, 3. Neither agree nor disagree, 4. Agree, 5. Strongly agree)			
Question's 1-20: see Appendix B			

Applying linearity to the value found for average for each of the factors and assuming that the sum of the limits must be less than or equal to 1, the limit for each of the factors identified were obtained. The previous quantification shows that the factors have different hierarchy between each of them, where the planning and execution were of significant importance of 0.26, the control factor limit was 0.25 and a training factor of 0.23. The following table shows the values obtained for each factor:

Table 5-11 Quantitative limit of each factor

Factor	mean	%	limit
Planning	3.7	26.24	0.26
Control	3.54	25.11	0.25
Training	3.24	22.98	0.23
Execution	3.62	25.67	0.26
Sum	14.10	100	1

The next step for validation purposes was to measure the utilization of each of the instruments identified in Table 5-4. To validate this information, a study with 20 randomly selected supervisors was conducted. The analysis was carried out to identify how activities are performed for each factor using the formats identified for the same (tables 5-5, 5-6, 5-7, 5-8). A standard measurement of 1 to 3 for all the instruments was used, where 1 is the full utilization of the instrument, and 3 when the instrument is not known. As noted in the factors for the level 0, the analysis of the supervisor factors must be made by the next person higher in command to avoid, at least to the level $n-1$, a self-assessment process.

The objective is to assess the number of subordinates that should be expected (on average). The study consisted of an interview with each supervisor where the evaluator collected data regarding the use and knowledge that each supervisor has in each of the formats identified before. Scales were constructed to define the variables operationally. These measured the degree of a particular factor present by linking together a number of items that could be used to measure that factor. The results obtained in these 20 evaluations are shown in the following table:

Table 5-12 Evaluation of primary dimensions of supervision work

Factor	Mean	Std Dev	Var
Planning	1.2		
SWOT Matrix	1.47	0.640	.410
Project letter	1.13	0.352	.124
Action plan	1.07	0.258	.067
Job description	1.13	0.352	.124
Control	1.36		
Meeting report	1.27	.704	.495
Scale method	1.00	0	0
Execution analysis	1.00	0	0

Performance	1.27	.458	.210
Support tools	2.27	.884	.781
Execution	1.28		
Interview	1.53	0.834	.695
Tech int	1.40	0.737	.543
Eval center	1.00	0	0
Pract test	1.00	0	0
Scale method	1.00	0	0
Work exec	1.33	.617	.381
Perform analysis	1.67	.816	.667
Training	1.40		
Interview	1.53	.834	.695
Tec Int	1.40	.737	.543
Eval center	1	0	0
Prac test	2.07	.704	.495
Scale 1-3 (1. Tool is known but not used, 2. Tool is known and used, 3. Tool is not used or known)			

Once the limit of the factors and the evaluation of each supervisor have been established, the next step was to determine the value of each factor that can be assigned to the supervisor. Applying linearity and using the following rule the value of the coefficient for each factor was calculated.

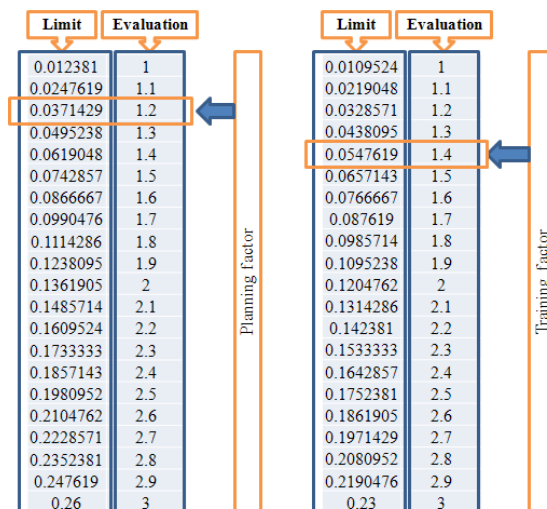


Figure 5-9 Evaluation of planning and training factor

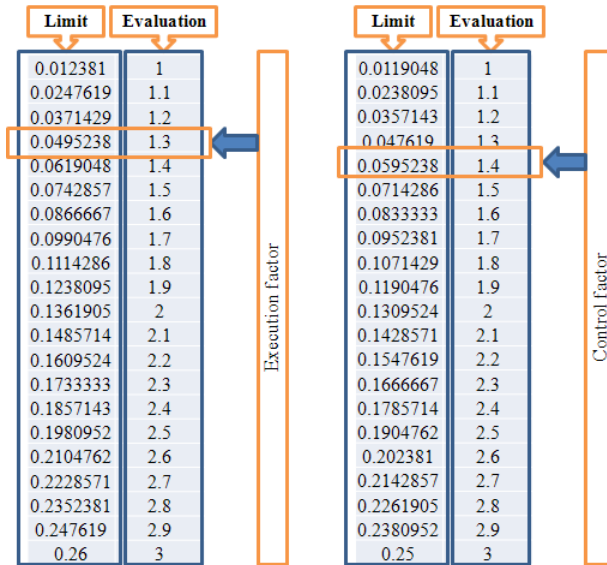


Figure 5-10 Evaluation of training and control factor.

With the data found on figures 4-3 and 4-4, and replaced in the following constraint:

$$\sum_j r_{oj} y_{0jp_0} (\rho_{ij} + c_{ij} + t_{ij} + e_{ij}) \leq \sum_j f_{ij} y_{ijp_1}$$

for an 8-hour workday, the constraint results in the following equation:

$$8 \text{ hrs} * (.037 + .059 + .054 + .049) = 8 \text{ hrs} * (0.199) = 1.592 \text{ hrs.}$$

This relationship indicates that the working group only requires 1.59 hours of supervision. Supervisor factors that affect each level must be calculated for each working group independently. If the working groups were homogeneous (same factor value for each group), then in the previous constraint the number of groups directed by each supervisor can be calculated with the following equation:

$$n_{\text{groups}} (8) (0.199) \leq 8(1)_{\text{supervisor}}$$

$$n_{\text{groups}} \leq \frac{8(1)_{\text{supervisor}}}{(8) (0.199)} \sim 5 \text{ groups}$$

Under these conditions, one supervisor could manage up to five working groups (workers). In the extreme case, where a supervisor obtains an assessment of one (perfect score) in all measuring instruments, it involves the following equation:

$$n_{\text{groups}} (8) (.045) \leq 8(1)_{\text{supervisor}}$$

$$n_{\text{groups}} \leq \frac{8(1)_{\text{supervisor}}}{(8) (0.045)} \sim 20 \text{ groups}$$

It is important to note that these results and tables of weights are relative to a particular type of organization and cannot be taken as the basis for any company. In any new design, the method must begin with the evaluation of the limit that the factors could take in order of the relevance of each factor related to the company and also the assessment made in measuring instruments.

The following figure shows the effect of the evaluation of the supervisor factors (planning, training, execution, and control) with respect to the number of subordinates who could be under each supervisor. The following figure shows that the minimum number of subordinates would be 1 and maximum 21:

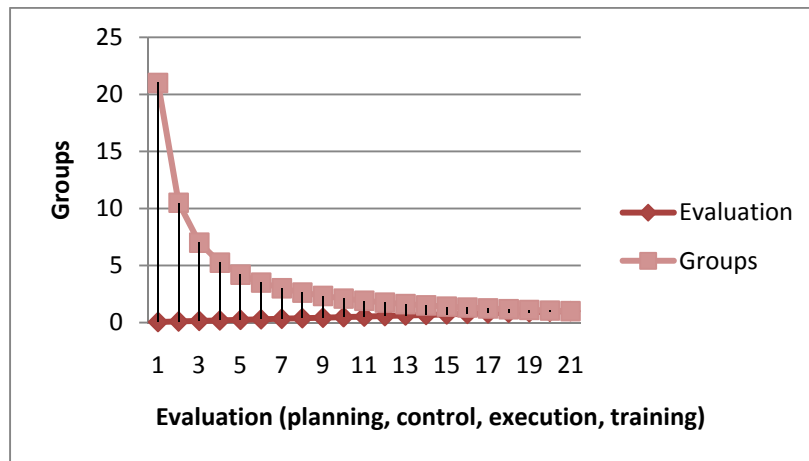


Figure 5-11 Relationship between groups and supervisor factors

5.3 Conclusions

From an examination of the literature of Organization Theory, four primary dimensions of supervision work were defined: planning, control, execution, and training. These concepts were then operated by generating 14 instruments measuring various aspects of the primary dimensions.

The results found in this chapter, regarding the span of control within the Information Systems (IS) companies, were consistent with the practice that those companies currently have between four and eight operators in charge for each supervisor and also the span of control found in the literature review. This method can give some guidelines to determine how a supervisor may be responsible for more working groups. The evaluation of each factor must be analyzed, and from this assessment, it must be determined how it can improve the performance for each factor involved. The span of control can also identify managers who need further training by measuring their performance. One of the main contributions of this quantification of factors was to determine an appropriate method to increase the number of subordinates (and minimize the span of control) considering the analysis and study of the factors involved.

Regarding the results about of the span of control in the literature review, they are consistent with the number of subordinates founded in the work of Van Fleet et al. [1] with an average SOC of five and also Entwisle and Walton [18] (average between 5 and 7). The only author that mentions a maximum SOC is Udell [19] with 30 subordinates. For this particular application case, the SOC suggested is (around) five with a maximum SOC of 21.

6. PROPOSED HEURISTIC APPROACH

The model under study is NP-hard, which means that the computational effort required to solve this problem increases exponentially with the problem size. When a commercial solver (such as LINGO) was used to solve the mathematical formulations presented in Chapter 4-1, it requires prohibitively a long run (or CPU) time. Consequently, a heuristic approach is presented in this chapter. Although this approach does not guarantee an optimal solution, it may require lesser computational time to find good solutions. In order to construct a feasible solution for the problem under study, two interdependent decisions must be made: (1) assign jobs to workers (level 0), and (2) assign the span of control for levels 1 to n. For each of the decisions, a two phase approach is proposed, one that first assigns jobs to workers (and workers to supervisors and so on) with a heuristic, and then optimizes the results with a meta-heuristic.

6.1 Introduction of the proposed heuristic approach

Basically the characteristics that the model has in order to consider for constructing the heuristic are: (1) the ability to divide the model into two sub-problems, (2) the formulation is a combinatorial integer programming (binary) model in which an optimal solution is sought over a discrete search-space, and (3) the possibility of constructing a "greedy algorithm" based on previous algorithms that have been successfully tested.

6.2 Problem decomposition and design framework

Once the actual problem is identified (see Chapter 4 for problem description), the next step was to decompose the problem into smaller distinct elements. The intent is to gain additional insights into the problem, and subsequently, a better understanding of the

solution approach. The proposed approach decomposes the binary integer model into two sub-problems: one assigns jobs to working groups, while the other focuses on the higher levels of managerial hierarchy. Using a hierarchical decomposition approach, the goal here is to divide the problem into independent smaller sub-problems: sub-problem one assigning jobs to workers at level 0, and sub-problem two assigning workers to supervisors at the next level (1 to n). Even though sub-problem two needs a feasible solution from sub-problem one, from this feasible solution at level 0, sub-problem two becomes independent in order to find a feasible solution for the whole problem.

Decomposition consists of groups of data: input and output. For level 0, input data required by the model consist of processing times according to complexity and compatibility data of jobs with workers, and capacity of workers. Output data for this problem consist of the assignment of jobs to workers. For the next level (1 to n) input data required by the model consist of number of workers selected in level 0, capacity, and the cost of supervisors.

Decomposition for this problem is described in following figure:

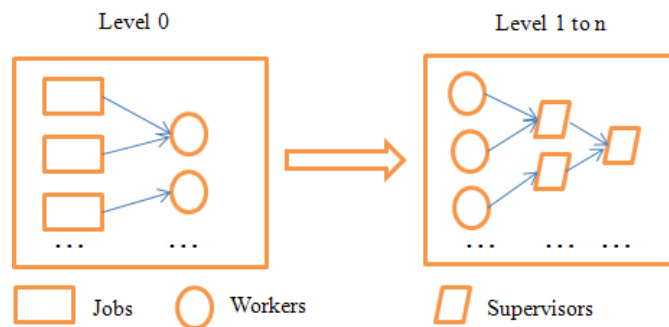


Figure 6-1 Problem decomposition

In this decomposition process, sub-problem 1 requires input 1 and 2, and sub-problem 2 requires output 1 (as input 3) as in the following figure:

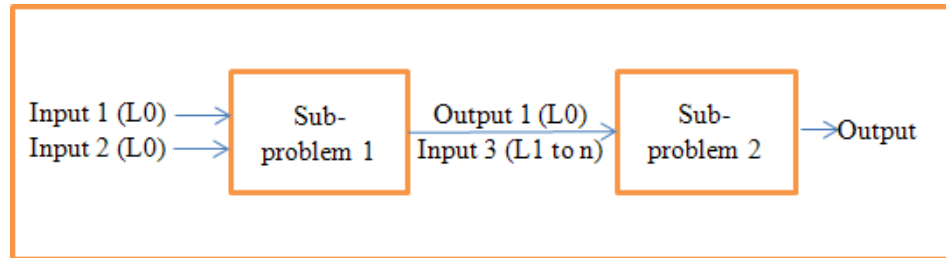


Figure 6-2 Input/output for sub-problems

Inputs and outputs required and obtained in this decomposition process are described as follows:

Sub-problem 1): a set of jobs (J) (input 1) and a set of different processing workstations (G) (input 2) are given. Each job (J) is described by its processing time for the jobs to be completed, the skills needed for the job, and the level of each skill (factor complexity) and compatibility according to the information given to the workstations. The objective is to find a set of workstations (G) to schedule the jobs (J) in such a way that the number of workstations can be minimized (output 1/input 3).

Sub-problem 2: Find the span of control needed for the Work Breakdown Structure (WBS), while maintaining the minimum cost of supervision. The number of span of control formed depends upon the number of workstations (G) considered in an instance, and also depends on the factors identified for supervision effort: planning, control, execution, and training (output).

The design framework for the proposed solution approach can be described in the following figure:

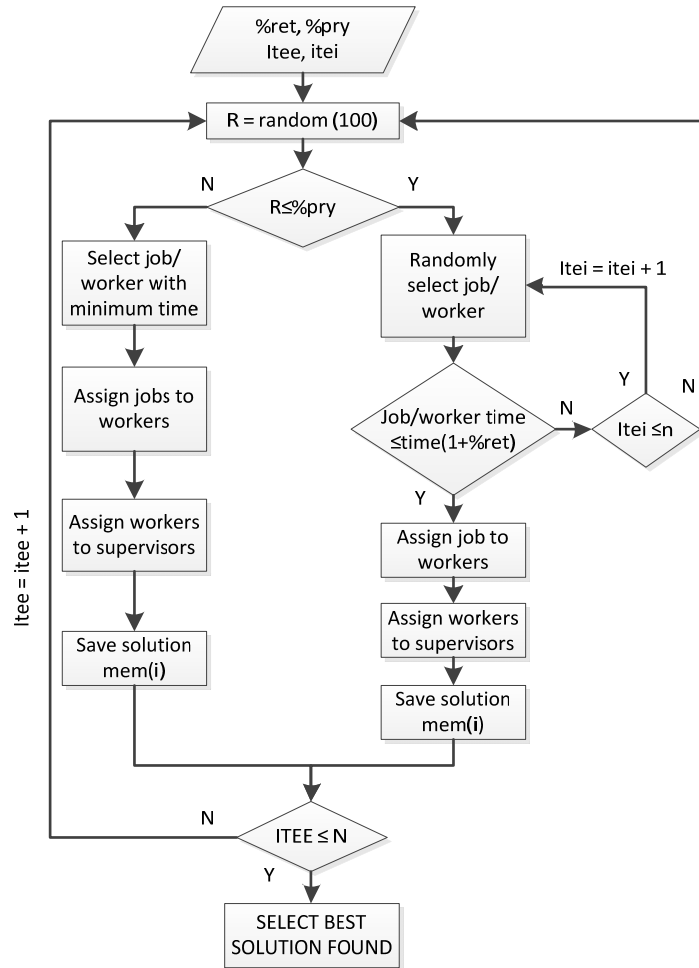


Figure 6-3 Meta-heuristic framework

The design framework is described below:

- Step 1: Select initial set of parameters ($\%ret$, $\%pry$, $itee$, $itei$)
- Step 2: Select random number R ($1 \leq R \leq 100$)
- Step 3: If $R \leq \%pry$, select job/worker with minimum processing time
- Step 4: Assign jobs to workers
- Step 5: Assign workers to supervisors

Step 6: Save solution in memory

Step 7: Repeat the process

Step 8: Randomly select jobs/workers

Step 9: If selection is between %ret, assign jobs and workers

Step 10: If selection is out of %ret, repeat the process until itei reaches n value

Step 11: Save solution in memory

Step 12: Select best solution found

6.3 Proposed heuristic approach

The structure of the problem consists of m constraints that will bound the construction of any feasible solution. When a job is chosen to be in the feasible solution, the addition of the new job cannot exceed the capacity constraints, e.g., for the first sub-problem:

$$\text{Min } \sum_{j=1}^N Y_{0j} \quad (\forall j \in J)$$

Subject to:

$$\sum_j Z_{ij} = 1 \quad (\forall i=1 \dots m)$$

and

$$\sum_i \beta_i \alpha_{ij} z_{ij} / n_{ij} \leq r_{0j} y_{0j} p_0 \quad (\forall j=1 \dots n)$$

The theme of greedy algorithm is to make the choices in sequence, such that each individual choice is the best according to some short-term or local criterion. Any choice can be made that seems best at the moment and then solve the sub-problems that arise later. The choice made by a greedy algorithm may depend on choices made so far, but not on future choices or all the solutions to the sub-problem. It iteratively makes one greedy choice after another, reducing each given problem into a smaller one.

Lagoudakis (L) (2001) [41] proposed, for a problem:

$$\text{Min } \sum_{j=1}^N c_j x_j \quad (\forall j \in J),$$

Subject to:

$$\sum_{i=1}^m a_{ij} x_i \leq b_j \quad (\forall j=1 \dots m)$$

and x binary

The following algorithm:

1) Order the items so that

$$c_i/a_i \geq c_{i+1}/a_{i+1} \quad \text{for } i = 1 \dots n-1$$

2) $CW = 0$

3) If $CW + a_i \leq b_j, j = 1$

Then $CW = CW + a_i; x_i = 1$

Else $x_i = 0$

The modified version of the Lagoudakis (ML) algorithm [41] that is proposed for the sub-problem one encompasses the following steps at level 0:

5.3.1 Sub-problem 1: Level 0

Min

$$\sum_{j=1}^N Y_{0j}$$

Subject to:

$$\sum_j Z_{ij} = 1$$

$$\sum_i \beta_i \alpha_{ij} z_{ij} / n_{ij} \leq r_{0j} Y_{0j} p_0$$

- Step 1) Initialization

$b_j = r_{0j}$ and Y_{0j} = decision variable

- Step 2) Order the workers (workstations) in descending according to the capacity of each station
- Step 3) Compute for all processing time a_{ij} the ratio $a_{ij} = \beta_i \alpha_{ij} z_{ij} / n_{ij}$
- Step 4) Find minimum a_{ij} ($j = n$); If the task does not exceeds the capacity of b_j , select the task for the workstation b_j then assign to a_{ij} a maximum value ($> any a_{ij}$) for all j . If the next (minimum) a_{ij} ($j = n$) exceeds the capacity of b_j , move to $j = n+1$.
- Step 5) Repeat step 3 until all the tasks have been assigned to a workstation

The procedure of the greedy algorithm at level 0 is shown in the following figure:

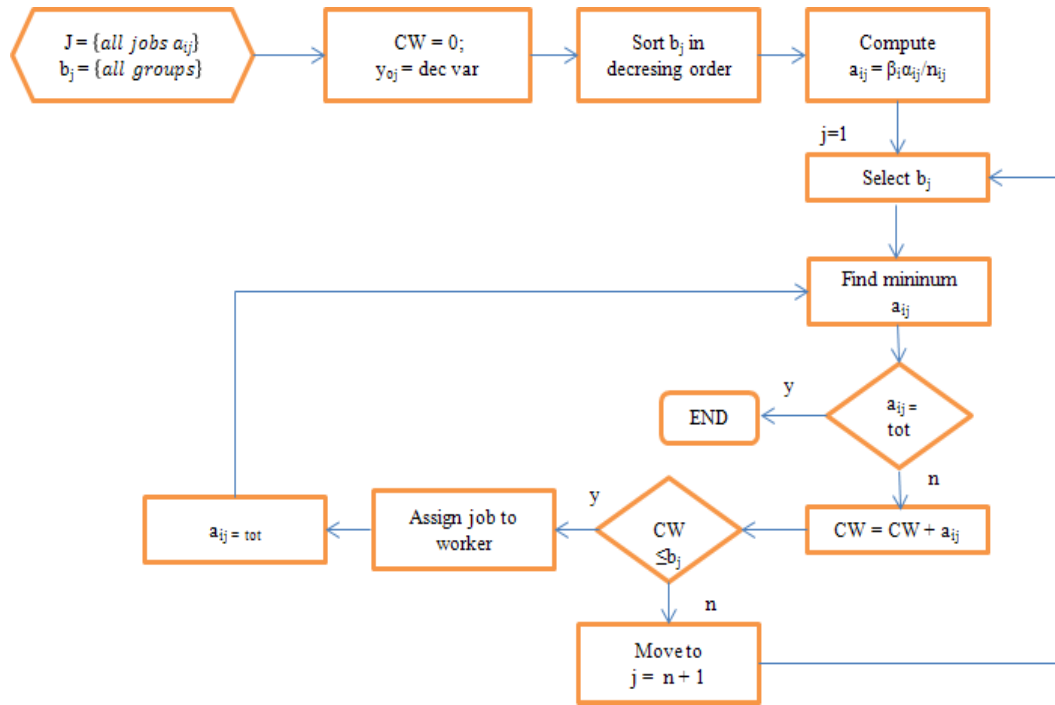


Figure 6-4 Procedure of the proposed solution approach (level 0)

The pseudo code for the modified Lagoudakis heuristic [41], at level 0, is illustrated in figure below.

Begin with

Let J be the set of workstations, and let a_{ij} be the modified processing time of task i with respect to workstation j (including complexity and compatibility factors).

Let n, m be the number of tasks and the number of workstations.

Let tot be the maximum processing time for a_{ij} once the task is assigned to a workstation and $b(j)$ the time capacity of workstation j .

Initialization

Compute matrix $a(i,j)$

Set n,m,tot

Reset cw

Rank workstation in descending order $b(j)$

For all workstations $b(j)$

For all activities $a(i,j)$

Find minimum $a(i,j)$

$cw = cw + a(i,j)$

If capacity of workstation $b(j)$ is not exceeded

Assign activity $a(i,j)$ to workstation $b(j)$

Assign to activity $a(i,j)$ a maximum value for all workstations

Else

End if

Endfor

Endfor

Return matrix $y(i,j)$ with the allocation of the activities to workstations (workers)

Figure 6-5 Procedure for the proposed heuristic approach

6.3.2 Sub-problem 2: Level 1 to n

$$\text{Min } \sum_{n=1}^N \sum_{j=1}^J w_{nj} f_{nj} y_{nj} p_n$$

Subject to:

$$\sum_j r_{oj} y_{oj} p_0 (\rho_{ij} + c_{ij} + t_{ij} + e_{ij}) \leq \sum_j f_{ij} y_{ij} p_1$$

$$y_{ojp_0} \leq \sum_{p1} y_{1p_0p1}$$

$$\sum_j f_{n-1,j} y_{n-1,jp_{n-1}} (\rho_{ij} + c_{ij} + t_{ij} + e_{ij}) \leq \sum_j f_{nj} y_{npjn}$$

$$y_{n-1,jp_{n-1}} \leq \sum_{pn} y_{npjn}$$

The procedure of the greedy algorithm at level 1 to n is shown in Figure 6-6

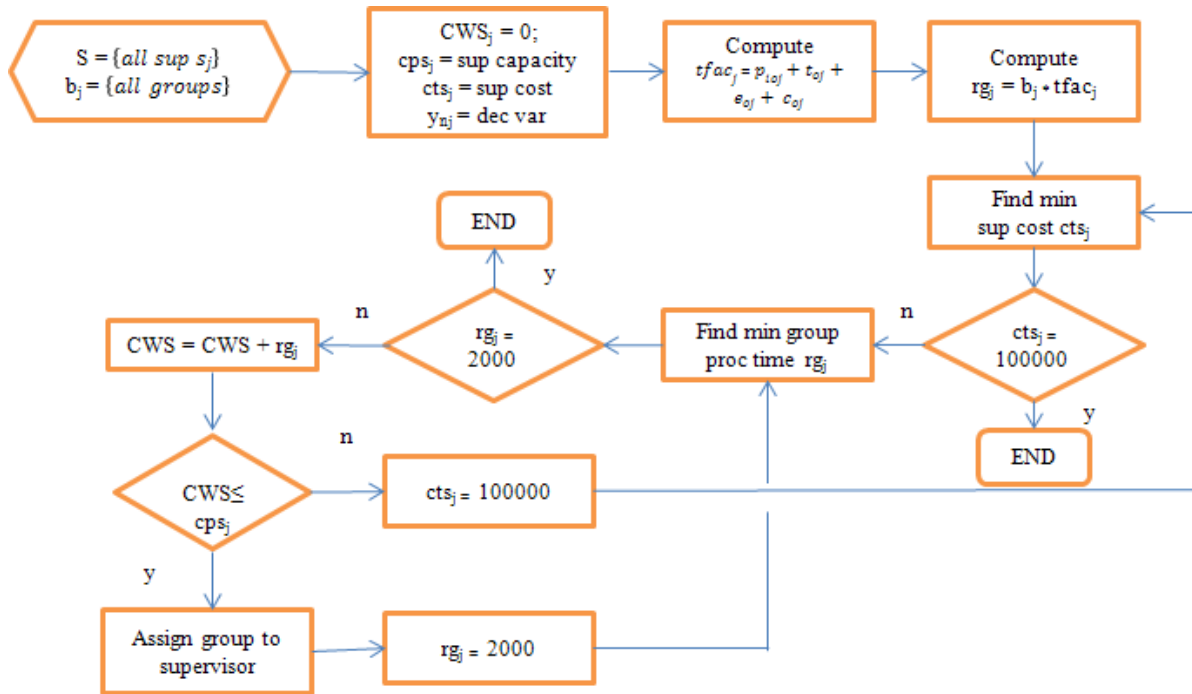


Figure 6-6 Procedure of the heuristic approach (level 1 to n)

The modified version of the Lagoudakis (ML) [41] algorithm that is proposed for the sub-problem two encompasses the following steps at level 1 to n:

- Step 1) Initialization
 Y_{nj} = decision variable
- Step 2) Compute for all workers $tfac_j = p_{ioj} + t_{oj} + e_{oj} + c_{oj}$
- Step 3) Compute value $rg(j) = b(j) * tfac(j)$ for all workers assigned at level 0.
- Step 4) Find supervisor with minimum cost

- Step 5) Find worker with minimum processing time rg_j ; if the worker does not exceed the capacity of supervisor j, select the worker for the supervisor, then assign to worker a constant value (2000). If the next (minimum) processing time exceeds the capacity of supervisor, move to $j = n+1$.
- Step 6) Repeat step 3 until all the tasks has been assigned to a workstation.

The pseudo code for the modified Lagoudakis [41] heuristic is illustrated in Figure 6-7

```

Begin with
  Let tfac be the supervision factor of supervisor j with each one of the
  workers i (including planning, execution, training, and control factors).
  Let ns, m be the number of supervisors and the number of workers.
  Let cps, cpt be the capacity and cost of each supervisor j.
Initialization
  Compute matrix tfac (j)
  Set ns,m
  Reset cws
For all workers
  | If the worker has been assigned at level 0
  |   Compute value  $rgt(j) = b(j) * tfac(j)$ 
  | Else
  |   Value  $rgt(j) = 2000$ 
  |   For all supervisors at level j
  |     Assign a value 2000 to each  $y(j,k)$ 
  |   Endfor
  | Endif
Endfor
For all supervisors j
  | Find supervisor with minimum cost
  | For all workers i
  |   Find minimum procesing time (rgt)
  |    $cws = cws + \text{minim } rgt(i)$ 
  |   If capacity of supervisor j is not exceeded
  |     Assign worker i to supervisor j
  |     Assign to worker  $a(i,j)$  a procesing time equal to 2000
  |   Else
  |     Assign cost of supervisor j equal to 100000
  |   End if
  | Endfor
Endfor
Return matrix  $yspt(i,j)$  with the allocation of the supervisors to workers

```

Figure 6-7 Procedure for the modified Lagoudakis greedy algorithm

6.4 A numerical example

The modified Lagoudakis heuristic [41] is illustrated through the following problem instance that includes four workstations, eight jobs, three supervisors, and two managers.

6.4.1 Sub-problem 1: Level 0

Table 6-1 Data of the numerical example

		GROUPS (WORKERS)							
		G1		G2		G3		G4	
CAPACITY		2		1		1		1	
TIME		JOBS							
		J1	J2	J3	J4	J5	J6	J7	J8
		0.2	0.3	0.5	1	0.8	0.75	0.1	0.15
GROUPS		COMPLEXITY FACTOR (JOBS)							
G1		0.6	0.6	0.6	1.00	1.00	1.20	1.20	1.20
G2		0.6	0.7	0.8	0.90	1.00	1.10	1.20	1.30
G3		1.5	2.00	1.5	1.00	1.00	0.60	0.70	0.60
G4		1	0.6	0.7	0.60	0.70	0.80	1.00	1.00
GROUPS		COMPATIBILITY FACTOR (JOBS)							
G1		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.01
G2		1.00	1.00	1.00	1.00	1.00	0.01	1.00	1.00
G3		0.01	1.00	1.00	1.00	1.00	1.00	0.01	0.01
G4		0.01	0.01	1.00	1.00	1.00	1.00	1.00	0.01

- Step 1) Enter the capacity available in hours for the workstations, the time required to accomplish each job, and the complexity and compatibility factors for each job with respect to each workstation, according to previous table

- Step 2) Sort the capacity of the workstations in descending order [$b(j) = 2, 1, 1, 1$]
- Step 3) Compute for all jobs the ratio $a_{ij} = \beta_i * \alpha_{ij} / \eta_{ij}$

Table 6-2 Processing time with complexity and compatibility factors

GROUPS	JOBS							
	J1	J2	J3	J4	J5	J6	J7	J8
G1	0.12	0.18	0.3	1	0.8	0.9	0.12	18
G2	0.12	0.21	0.4	0.9	0.8	82.5	0.12	0.195
G3	30	0.6	0.75	1	0.8	0.45	7	0.9
G4	20	18	0.35	0.6	0.56	0.6	0.1	15

- Step 4) Find minimum a_{ij} ($j = n$); if the job does not exceed the capacity of b_j , select the job for the workstation b_j , then assign to a_{ij} a maximum value ($> any a_{ij}$) for all j . If the next (minimum) a_{ij} ($j=n$) exceeds the capacity of b_j , move to $j = n+1$.

Table 6-3 Procedure for Step 4

Sp	Wk	J1	J2	J3	J4	J5	J6	J7	J8	a_{ij}	CW_i	b_j
1	G1	.12	.18	.3	1	.8	.9	.12	18	(1,1)	$CW_1=0.12$	$b_1 = 2$
2	G1	M	.18	.3	1	.8	.9	.12	18	(1,7)	0.24	$b_1 = 2$
3	G1	M	.18	.3	1	.8	.9	M	18	(1,2)	0.32	$b_1 = 2$
4	G1	M	M	.3	1	.8	.9	M	18	(1,3)	0.62	$b_1 = 2$
5	G1	M	M	M	1	.8	.9	M	18	(1,5)	1.42	$b_1 = 2$
6	G1	M	M	M	1	M	.9	M	18	(1,6)	2.32 > 2	$b_1 = 2$
7	G2	M	M	M	0.9	M	82.5	M	.195	(2,8)	$CW_2=0.195$	$b_2 = 1$

8	G2	M	M	M	0.9	M	82.5	M	M	(2,4)	1.195>1	$b_2 = 1$
9	G3	M	M	M	1	M	.45	M	M	(3,6)	CW ₃ =0.45	$b_3 = 1$
10	G3	M	M	M	1	M	M	M	M	(3,4)	1.45>1	$b_3 = 1$
11	G4	M	M	M	0.6	M	M	M	M	(4,4)	CW ₄ =0.6	$b_4 = 1$
12	G4	M	M	M	M	M	M	M	M		STOP	

The following table shows the results of the assigning problem at level 0

Table 6-4 Job allocations at level 0

GROUPS (workers)	G1	G2	G3	G4
Capacity (b _j)	2	1	1	2
Jobs	J1-J2-J3-J5-J7	J8	J6	J4

6.4.2 Sub-problem 2: Level 1 to n

Table 6-5 Data of the numerical example (level 1)

DATA OF SUPERVISORS		SUPERVISORS			
LEVEL 0 → 1		S1	S2	S3	
CAPACITY (Hrs)		3	2	1	
COST (w)		\$200	\$150	\$100	
FACTORS OF SUPERVISION EFFORT	WORKERS (GROUPS)				
	G1	G2	G3	G4	
PLANNING (ρ)		0.02	0.015	0.01	0.02

CONTROL (c)	0.03	0.1	0.14	0.11
TRAINING (t)	0.1	0.2	0.01	0.02
EXECUTION (e)	0.11	0.12	0.01	0.05

- Step 1) Initialization

Y_{0j} = decision variable

- Step 2) Compute for all group_j the ratio $r_{0j} = p_{0j} + c_{0j} + t_{0j} + e_{0j}$

Table 6-6 Supervision factors

	G1	G2	G3	G4
$r_{0j} = p_{0j} + c_{0j} + t_{0j} + e_{0j}$	0.13	0.38	0.33	0.29

- Step 3) Find supervisor with minimum cost s_j ($j = n$);

Table 6-7 Data of capacity and cost (level 1)

Supervisor	Capacity (hrs)	Cost
S1	3	200
S2	2	150
S3	1	100

- Step 4) Find group with minimum time; if the group does not exceed the capacity of supervisor f_j , select the group for the supervisor b_j then assign to a_{ij} a maximum value ($>$ any a_{ij}) for all j. If the next (minimum) a_{ij} ($j=n$) exceeds the capacity of b_j , move to $j = n+1$. If group j was not selected in level 0, assign a value of 2000.

Table 6-8 Procedure for level 1

Sp	Sup _j	r _{ojG1}	r _{ojG2}	r _{ojG3}	r _{ojG4}	Select	CWsup	Cap _{sj}	Cost _j
1	S3	.13	.38	.33	.29	(3,1)	0.13	1	100
2	S3	M	.38	.33	.29	(3,4)	0.42	1	100
3	S3	M	.38	.33	M	(3,3)	0.77	1	100
4	S3	M	.38	M	M	(3,2)	1.15>1	1	M
5	S2	M	.38	M	M	(2,2)	.38	2	150
6	S2	M	M	M	M	-----		STOP	

- Step 5) If group exceeds the capacity of supervisor j, assign a value of M to the supervisor cost j.
- Step 6) Repeat all steps for level 2 to n (until all the supervisors have been assigned to a manager).

6.5 Solution improvement by a meta-heuristic (Meta-RaPS)

A natural extension to the constructive heuristic presented in Chapter 6.1 is to develop an improve heuristic that can draw better results for the solutions found. A meta-heuristic for the Organizational Design Problem is presented in this chapter to solve the problem under study.

6.5.1 Introduction to Meta-RaPS

The application of Meta-RaPS to any combinatorial problem, according to Moraga et al. [41], as a general methodology considers the following steps:

1. Study the structure of the problem to be solved
2. Find good priority rules that construct feasible solutions
3. Modify priority rules using randomness
4. Construct feasible solutions using priority rule and randomness
5. Pick the best answer for a number of iterations
6. Improve solutions

According to DePuy [29] and Moraga et al. [54] Meta-RaPS modifies the way a greedy construction heuristic chooses the next activity to add to the solution, by occasionally choosing an activity that does not have the best priority value. Sometimes the next activity is chosen randomly from those feasible activities with good, but not the best, priority values. The heuristic approach begins with a well-defined mathematical representation of a problem. There is an objective or evaluation function that provides the value of any specific solution (values of the set of decision variables), and there are specified constraints that define the region of feasible solutions. Ideally, one would like to select the so-called optimal solution that achieves the minimum (in this case) value.

6.5.2. Modify priority rules using randomness

Once there is a priority rule that constructs feasible solutions (6.1 proposed solution approach), the next step is to use the framework of Meta-RaPS for improving the heuristic method. The greedy algorithm can be randomized altering the composition of the “next available activity” using a restriction range (r) so $R = c_i/a_{ij} [(1 - r/100), 1]$. The next variable will be picked randomly from the list of items.

Instead of selecting at each iteration the smallest a_{ij} , a wider list is chosen that contains more than one item available as the next item added to the solution. If the restriction percentage (%r) is 35%, that means the “next available activity” list is comprised of those not chosen variables whose ratios are within a distance $R = c_i/a_{ij}$ [0.65, 1]. Using the same data as the previous problem, the processing time is changed according to $a_{ij} = \beta_i \alpha_{ij} z_{ij} / n_{ij}$ as shown in table below.

Table 6-9 Processing time

GROUPS	JOBS							
	J1	J2	J3	J4	J5	J6	J7	J8
G1	0.12	0.18	0.3	1	0.8	0.9	0.12	18
G2	0.12	0.21	0.4	0.9	0.8	82.5	0.12	0.195
G3	30	0.6	0.75	1	0.8	0.45	7	0.9
G4	20	18	0.35	0.6	0.56	0.6	0.1	15

Table below shows an example of the use of the percentage restriction (%r):

Table 6-10 Candidates for “next available activity”

For j =	$R = c_i/a_{ij} [(1 - r/100), 1]$ %ret = 35%	The available list contains values [x,y] to choose randomly
1	$R' = a_{ij}[0.12, 0.15]$	[0.12, 0.12]
2	$R' = a_{ij}[0.12, 0.15]$	[0.12, 0.12]
3	$R' = a_{ij}[0.45, 0.6]$	[0.45, 0.6]
4	$R' = a_{ij}[.1, 0.13]$	[0.1]

The modified version of the algorithm (with a percentage restriction *%ret*) encompasses the following steps at level 0.

- Step 1) Initialization
 - Set $b_j = r_{0j}$ and Y_{0j} = decision variable
- Step 2) Compute for all a_{ij} the ratio $a_{ij} = \beta_i \alpha_{ij} z_{ij} / n_{ij}$
- Step 3) Use a percentage restriction (*%ret*) in order to find the list of “next available supervisor” with the minimum cost:
 - Find minimum a_{ij}
 - Pick randomly a_{ij} for $j = n$; if the task $a_{ij} \leq \text{minimum } a_{ij} * (1 + \%r)$ and does not exceed the capacity of b_j , select the task for the workstation b_j , assign to a_{ij} a maximum value ($>$ any a_{ij}) for all j . If the next task $a_{ij} \leq \text{minimum } a_{ij} * (1 + \%r)$ exceeds the capacity of b_j , move to $j = n+1$. If task $a_{ij} \geq \text{minimum } a_{ij} * (1 + \%ret)$, choose another a_{ij} until task $a_{ij} \leq \text{minimum } a_{ij} * (1 + \%ret)$ or stop at n iterations according to variable “itei”
- Step 4) Repeat Step 3 until the entire tasks has been assigned to a workstation or the algorithm determines that the problem is unfeasible after n iterations according to variable “itee”.

The modified version of the algorithm (with a percentage restriction *%ret*) encompasses the following steps at level 1 to n .

- Step 1) Initialization: Level 1 to n

- Set $b_j = r_{oj}$ and $Y_{oj} =$ decision variable
- Step 2) Compute for all workers $tfac_j = p_{oj} + t_{oj} + e_{oj} + c_{oj}$
- Step 3) Use a percentage restriction (%ret1) in order to find the list of “next available supervisor” with the minimum cost (cts).
 - Find supervisor with minimum cost (cts)
 - Pick randomly any supervisor for $j = n$; if the supervisor cost $cts \leq$ minimum $cts * (1 + \%ret1)$ and does not exceeds the capacity of cps_j , select worker with minimum processing time, assign worker to supervisor j . Assign to rgt_j a maximum value for all j . If the next supervisor cost $cts \leq$ minimum $cts_j * (1 + \%ret1)$ exceeds the capacity of cps_j , move to $j = n+1$.
 - If cost $cts_j \geq$ minimum $cts_j * (1 + \%ret1)$, choose another supervisor until cost $cts_j \leq$ minimum $cts_j * (1 + \%ret1)$ or stop at n iterations according to variable “itei”
- Step 4) Repeat Step 3 until all the workers have been assigned to a supervisor or the algorithm determines that the problem is unfeasible, after n iterations according to variable “itee”.
- Step 5) Repeat Steps 1 to 4 until the entire span of control has been assigned.

6.5.2 Composition of the “next available task”

The greedy algorithm is modified by altering the composition of the “next available task”. In this step, a priority percentage (%p) is used (as suggested by Moraga et al. [41]) in order to set a percentage of the time the greedy algorithm, and the percentage of time

the modified algorithm will be used to select the next task. If the $\%p = 20$, that means 20% of the time the greedy algorithm will be used, and 80% of the time the modified algorithm will be used to select the next task.

This procedure is run for a user-defined number of replications and iterations in the meta-heuristic, and the overall best solution in each replication can be reported which, can also be used to testify the robustness of the heuristic.

The procedure of meta-heuristic at level 0 is shown in the following figure:

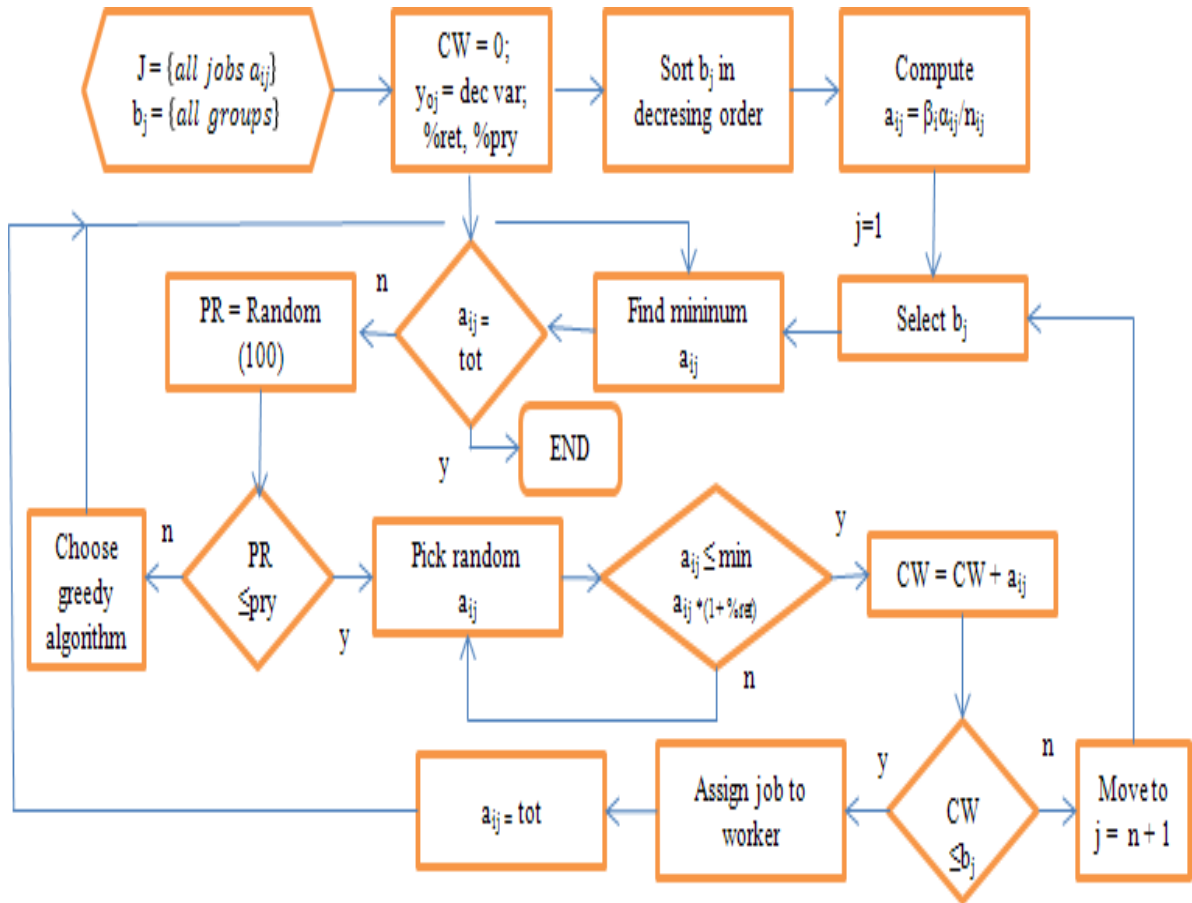


Figure 6-8 Procedure of the meta-heuristic proposed at level 0

The pseudo code for the meta-heuristic for level 0 can be illustrated in figure below:

Begin with

Let J be the set of workstations, and let a_{ij} be the modified processing time of task i with respect to workstation j (including complexity and compatibility factors).

Let n, m be the number of tasks and the number of workstations.

Let tot be the maximum processing time for a_{ij} once the task is assigned to a workstation and $b(j)$ the time capacity of workstation j .

Initialization

Compute matrix $a(i,j)$

Set parameters $\%ret, \%pry, itee, itei$.

Set n,m,tot

Rank capacity of workers in decreasing order $b(j)$

For a fixed number of iterations **DO** ($itee$)

For all workstations $b(j)$

For all activities $a(i,j)$

 Find minimum $a(i,j)$

 Generate a random number PR (1-100)

If $PR \leq \%pry$

 Choose the greedy algorithm from the available list of activities and assign to the worker

Else

For a fixed number of iterations **DO** ($itei$)

 Generate a random number $c(n)$

 Select one activity randomly

If the processing time of the activity is within the minimum processing time and $\% rest$

 Assign the activity to the worker

 Assign to activity $a(i,j)$ a maximum value for all workstations

End if

Endfor

Endif

Endfor

Endfor

Endfor

Return matrix $y(i,j)$ with the allocation of the activities to workstations (workers)

Figure 6-9 Meta-RaPS procedure for the solution approach

The procedure of meta-heuristic at level 1 to n is shown in the figure below:

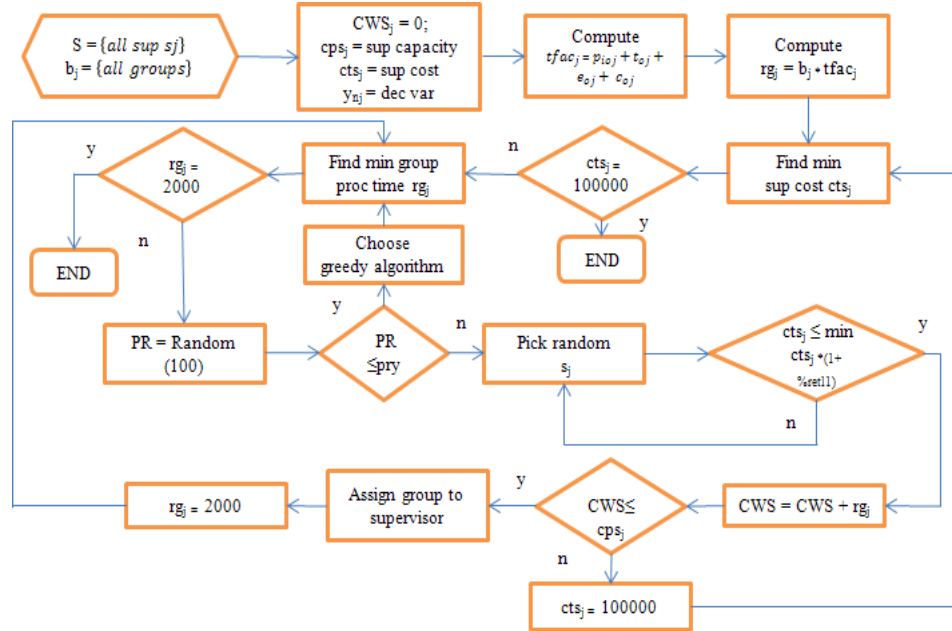


Figure 6-10 Procedure of the meta-heuristic proposed at level 1

The pseudo code for the meta-heuristic for level 1 to n can be illustrated as in figure 6-11:

```

Initialization
Read t(j) from level 1
Set parameters %ret1
Set ns, nm

For a fixed number of iterations DO (itee)
  Compute value rgt(j) if supervisor was selected at level 1
  Find manager with minimum cost
  Generate a random number
  If PR <= %ptry
    Choose the greedy algorithm from the available list of
    supervisors and assign to the workers according minimum
    processing time
  Else
    For a fixed number of iterations DO (itei)
      Generate a random number c(ns)
      Select one manager randomly
      If the cost of the manager is within the minimum
      cost and %ret1
        Assign the manager to the supervisors
      End if
    Endfor
  Endif
Endfor
Return matrix y(i,j) with the allocation of the supervisors and
workstations (workers)

```

Figure 6-11 Meta-RaPS procedure for the solution approach (level 1)

All procedures programmed in Matlab are shown in Appendix C.

6.6.3 Setting parameters

Almost all known meta-heuristic available have a number of parameters that need to be set. Combine the settings obtained in the “modify priority rules” in order to run the entire collection of restriction range. In order to construct a feasible solution, the following parameters: %ret (restriction percentage), %p (priority percentage), and the number of iterations need to be set. In order to set these parameters, this model uses some ideas reported by DePuy et al. [41] which consists in the following steps: 1) select a representative subset of problems to analyze, 2) select the appropriate parameter domain, 3) select good parameter settings for each problem, and 4) run the entire subset of problems.

The methodology to evaluate the heuristic design consists of comparing results obtained by the algorithm on test problems already solved with LINGO (with optimal values). This heuristic design is a refinement of the Meta-RaPS. The customization of the meta-heuristic is in the construction of the greedy algorithm for this particular application and adjustments made to find the optimal values of mentioned parameters. The following table shows the size of problems used to evaluate the meta-heuristic.

Table 6-11 Size of problems used to evaluate the meta-heuristic

Case (size)	m (jobs)	n (groups)	ns (sup)	nm (man)	Constraints	Variables
A	8	4	3	2	41	22
B	30	10	5	3	58	318
C	40	10	5	3	68	418

D	60	20	5	3	108	1228
E	100	40	5	3	188	4048
F	200	40	10	5	293	8055
G	300	40	10	5	393	12055

The following figure shows the number of variables and constraints for each problem size:

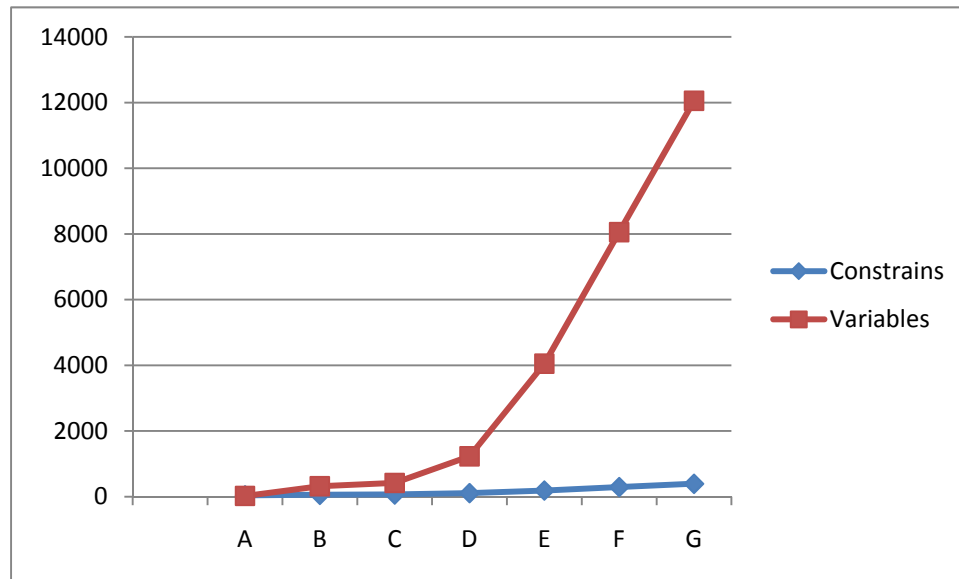


Figure 6-12 Number of variables and constraints

In order to implement the meta-heuristic, six representative problems in terms of the number of variables and constraints were selected. Table 6-12 shows the subset of problems chosen:

Table 6-12 Subset of problems selected for setting parameters

Problem	m	n	ns	nm	Optimal
(B)	(jobs)	(groups)	(sup)	(man)	value
CAS#1	30	10	5	3	4205

CAS#2	30	10	5	3	9524
CAS#3	30	10	5	3	1103
CAS#4	30	10	5	3	2103
CAS#5	30	10	5	3	1003
CAS#6	30	10	5	3	456

Using parameters of 200 iterations (itee), 20 iterations (itei) and range values of (0.1, 4), and (10, 60) arbitrarily chosen for %r and %p respectively, the subset is solved by meta-heuristic and the best combination of parameters is selected. The criterion used for level 0 is to select a good combination of pairs (%ret, %p) that produces the less deviation from optimal values. Table 6-13 shows the deviation (%) of the average six problems, i.e. with %ret= 0.1 and %p = 20, the value of 116 means that the average of the deviation from optimal value of the six problems were 116%. It is observed that the minimum deviation is with %ret = 4 and %p = 50.

Table 6-13 Deviation (%) from the optimal solution at level 0

	%p=10	20	30	40	50	60
%ret =	<i>No</i>					
0.1	<i>Feasible</i>	116	108	64	54	54
0.3	116	92	76	54	40	35
0.5	93	82	66	42	49	35
1	60	56	42	42	25	29
1.5	49	54	35	37	35	25
2	67	51	42	35	35	29
2.5	54	42	35	39	25	29
3	42	42	35	42	29	31
3.5	58	33	42	29	29	29

4	35	31	25	31	21	25
5	42	35	29	35	29	25

Using the above results (%ret = 4, %p = 50) for level 0, Table 6-14 shows pairs (%ret1, %p1) that got highest frequencies out of six problems for a deviation less or equal than 5% throughout the subset (level 2). With values of %ret1 = 1.5 or above the frequencies are 6, therefore the combination of parameters selected are %ret0 = 4, %ret1 = 1.5 and %p0 = %p1 = 50 (In level 1 and 2, the %p is irrelevant between values 10-60). Table 6-15 shows that with the parameters selected the (average) deviation from optimal values is 1.04%.

Table 6-14 Frequencies out of six problems getting a deviation of 5% or less

Level 1 to n

	%p1=10	20	30	40	50	60
%r1 = 0.1	3	3	3	3	3	3
0.3	3	3	3	3	3	3
0.5	4	4	4	4	4	4
1	4	4	4	4	4	4
1.5	6	6	6	6	6	6
2	6	6	6	6	6	6
2.5	6	6	6	6	6	6
3	6	6	6	6	6	6
3.5	6	6	6	6	6	6
4	6	6	6	6	6	6
5	6	6	6	6	6	6

Table 6-15 Deviation (%) from the optimal solution and the meta-heuristic

Level 1 to n

	%p=10	20	30	40	50	60
r = 0.1	23	23	23	23	23	23
0.3	23	23	23	23	23	23
0.5	4.29	4.29	4.29	4.29	4.29	4.29
1	2.7	2.7	2.7	2.7	2.7	2.7
1.5	1.04	1.04	1.04	1.04	1.04	1.04
2	1.04	1.04	1.04	1.04	1.04	1.04
2.5	1.04	1.04	1.04	1.04	1.04	1.04
3	1.04	1.04	1.04	1.04	1.04	1.04
3.5	1.04	1.04	1.04	1.04	1.04	1.04
4	1.04	1.04	1.04	1.04	1.04	1.04
5	1.04	1.04	1.04	1.04	1.04	1.04

6.3 Evaluation of solution quality

In table 6-17, preliminary results are shown of the meta-heuristic solving 30 cases of size A and B (commercial solver LINGO could not solve problems of size equal or larger than C).

Table 6-16 Preliminary Meta-Heuristic Results (Case A and B)

Case	Name	itee	itei	%p	%ret0	%ret1	value	%dev From optimal
1	CAS1-A	200	40	50	4	1.5	2553	0
2	CAS2-A	200	40	50	4	1.5	4603	0
3	CAS3-A	200	40	50	4	1.5	2404	0
4	CAS4-A	200	40	50	4	1.5	1603	10.39
5	CAS5-A	200	40	50	4	1.5	1453	0
6	CAS6-A	200	40	50	4	1.5	2704	0
7	CAS7-A	200	40	50	4	1.5	2364	6.77
8	CAS8-A	200	40	50	4	1.5	354	0.28
9	CAS9-A	200	40	50	4	1.5	5603	0

10	CAS10-A	200	40	50	4	1.5	603	0
11	CAS11-A	200	40	50	4	1.5	2603	0
12	CAS12-A	200	40	50	4	1.5	402	0
13	CAS13-A	200	40	50	4	1.5	503	0.19
14	CAS14-A	200	40	50	4	1.5	2652	1.92
15	CAS15-A	200	40	50	4	1.5	1704	0.05
16	CAS16-A	200	40	50	4	1.5	2003	0
17	CAS17-A	200	40	50	4	1.5	2753	0
18	CAS18-A	200	40	50	4	1.5	6402	0
19	CAS19-A	200	40	50	4	1.5	823	0
20	CAS20-A	200	40	50	4	1.5	2303	0
21	CAS1-B	200	40	50	4	1.5	4206	0.02
22	CAS2-B	200	40	50	4	1.5	9605	0.85
23	CAS3-B	200	40	50	4	1.5	1104	0.09
24	CAS4-B	200	40	50	4	1.5	2204	4.8
25	CAS5-B	200	40	50	4	1.5	605	0.33
26	CAS6-B	200	40	50	4	1.5	704	0.1422
27	CAS7-B	200	40	50	4	1.5	1004	0.09
28	CAS8-B	200	40	50	4	1.5	3404	0.029
29	CAS9-B	200	40	50	4	1.5	458	0.43
30	CAS10-B	200	40	50	4	1.5	504	0.19

Table 6-17 Average, std dev and maximum % difference from optimal

Method	Average % Difference from optimal	Std Dev % Difference from optimal	Maximum % Difference from optimal
Meta-heuristic	0.8809667	2.3372189	10.39

The method used to evaluate the meta-heuristic consists of comparing results obtained by the meta-heuristic on test problems with their optimal values; also the meta-heuristic was compared to the pure greedy algorithm results. Table 6-18 shows results of applying the meta-heuristic and the modified greedy approach (originally suggested by Lagoudakis [41]).

Table 6-18 Comparison of the meta-heuristic versus greedy algorithm

Case	Name	m	n	ns	nm	Opt Value LINGO	Greedy algorithm	Meta-heuristic	%d1 (ga) from optimal	%d2 (mh) from optimal
1	CAS1-TA	8	4	3	2	2553	2653	2553	3.91696	0
2	CAS2-TA	8	4	3	2	4603	9404	4603	104.302	0
5	CAS5-TA	8	4	3	2	1453	1604	1453	10.3923	0
10	CAS10-TA	8	4	3	2	603	754	603	25.0415	0
11	CAS11-TA	8	4	3	2	2603	5804	2603	122.973	0
14	CAS19-TA	8	4	3	2	823	824	823	0.12151	0
21	CAS1-TB	30	10	5	3	4205	4206	4206	0.02378	0.02378
24	CAS4-TB	30	10	5	3	2103	2404	2204	14.3129	4.80266
27	CAS7-TB	30	10	5	3	1003	1104	1004	10.0698	0.0997
29	CAS9-TB	30	10	5	3	456	958	458	110.088	0.4386

Another useful evaluation of the performance of the meta-heuristic is the time of response. Even though a commercial solver (LINGO) outperforms the meta-heuristic in the speed of the process, this solver can only handle problems of size (jobs x groups) until 30 jobs and 10 groups, while the meta-heuristic could solve bigger problems in a reasonable amount of time. Table 6-19 and Figure 6-11 show the performance of the meta-heuristic in each of the problems used for this evaluation.

Table 6-19 Time of response.

Case size (jobs x groups)	LINGO (seconds average)	META-HEURISTIC (seconds average)
8 x 4	00	0.46

16 x 6	01	5.56
30 x 10	01	7.76
40 x 10	**	8.97
60 x 20	**	25.33
100 x 40	**	71.33
200 x 40	**	125.4
300 x 40	**	406

Note: ** means that LINGO cannot solve problems of this size

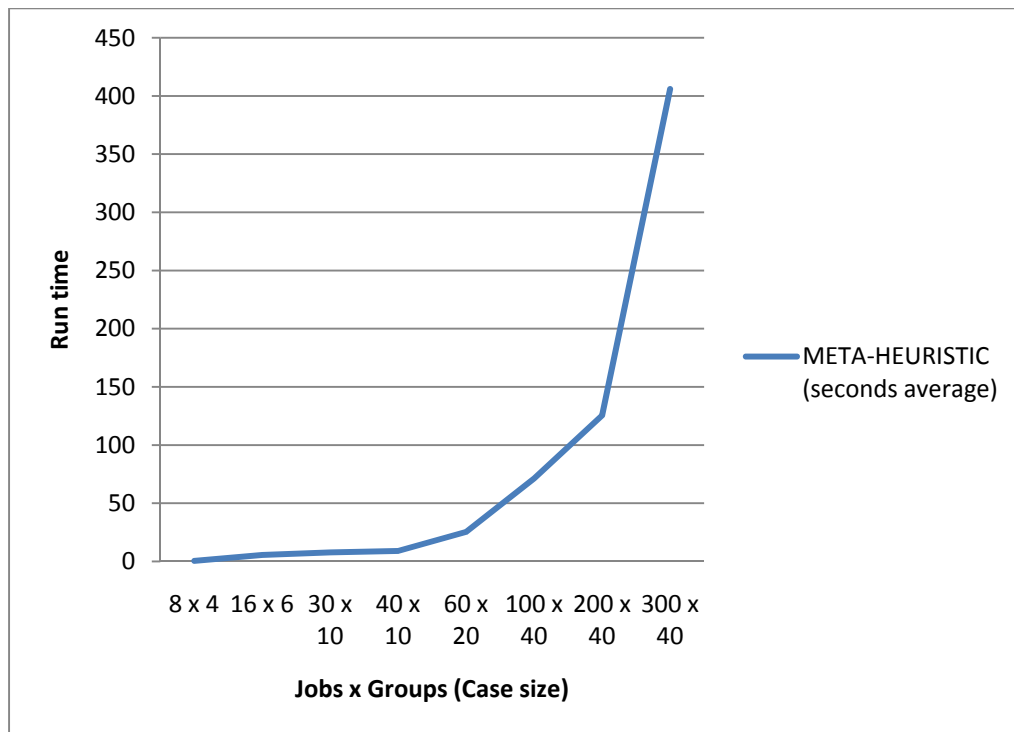


Figure 6-13 Time of response

Using these data, the computation time required by the algorithm for larger problems can be approximated. Using the number of variables (of the seven cases) a polynomial equation of degree 3 with a correlation of 0.9993 was obtained, as shown in the following figure:

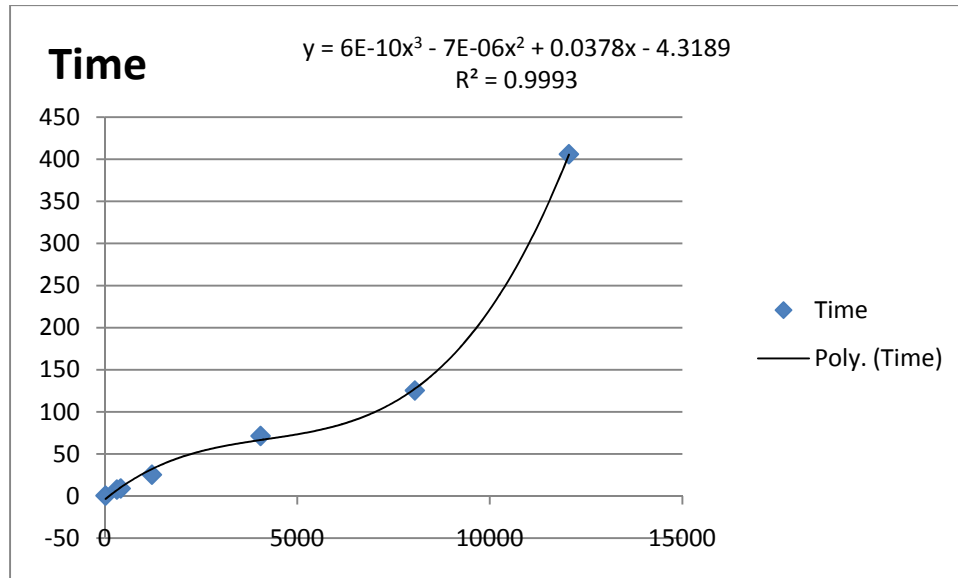


Figure 6-14 Interpolation of the number of variables

The number of variables used for each problem can be obtained with the formula:

$$\text{Vars} = (g*j) + g + s + m$$

Where:

- Vars = Number of variables
- g = number of workers (groups)
- j = number of jobs (activities)
- s = number of supervisors
- m = number of managers

Table 6-20 Approximate running time for bigger problems

Case (size)	jobs (j)	workers (g)	sup (s)	man (m)	# of vars	Aprox time in seconds	Aprox time in min	Aprox time in hrs

H	500	40	10	5	20055	2778.048	46.3008	0.77168
I	600	40	10	5	24055	5206.014	86.7669	1.446115
J	700	40	20	5	28065	8806.17	146.7695	2.446158
K	1000	50	20	5	50075	59673.98	994.5664	16.57611

6.4 Results and discussion

The topic of Chapter 6 was to apply a simple and effective meta-heuristic into the Organizational Design problem. In this research, an effective heuristic to solve the set of covering problems was designed based on the meta-heuristic called Meta-RaPS, and extensive experiments were conducted. The following conclusions can be drawn: all of the results in the meta-heuristic were obtained for each test problems using parameter values of %p=50, %ret=4, %ret1 = 1.5, itee = 200, itei = 20. The construction heuristic is efficient and solves problems with 343 constraints and 12055 variables in 406 seconds on a Dell M4400 workstation computer by using MatLab R2009b. Table 6-17 compares the average, standard deviation and maximum deviation from optimal value. These results are consistent with those reported by Song et al. [48] in the “Meta-Heuristic for the vehicle problem” based, also, on the Meta-RaPS procedure. Flexible and reconfigurable software has been implemented for this heuristic. Users can change the default values for some parameters such as %priority, %restriction (level 0 and 1 to n) and also the number of iterations (itee and itei). Users can either speed up the execution of the program for an

estimated solution, or slow down to obtain a good result, whereas the number of iterations can affect the quality of the solution obtained in this subset of problems.

In order to test the effectiveness of the meta-heuristic, 30 problems of different sizes (A and B) were solved. One of the main advantages of this meta-heuristics (as reported by Moraga et al. [41]) is the limited number of parameters compared with others, in this case five parameters (%ret, % ret1, %p, itee and itei) were used.

Table 6-21 compares the average and standard deviation from optimal values between the greedy algorithm and the meta-heuristic (based on results of 10 problems of Table 6-18). Introducing randomness into the construction heuristic greatly improves the solution quality; instead of choosing the best element as in the greedy heuristic, meta-heuristic selects the best element only during certain percentage of time, while in the remaining time, it chooses a good element from a list of candidates. For this problem, the improvement of solution quality via the randomized construction heuristic is around 80-90%.

Table 6-21 Average, STD dev and max dev. (%) between greedy and meta-heuristic

Method	Average % Difference from optimal	Standard Dev. % Difference from optimal	Maximum % Difference from optimal
Greedy heuristic	40.1241	50.638	122.973
Meta-heuristic	0.536	1.505	4.80266

Table 6-20 shows an approximation in the computation time required for problems with a thousand operations, fifty workers, twenty supervisors and five managers in approximately 16 hours, however the meta-heuristic can be stopped before this time and also get a feasible solution. With the number of variables generated by the model, the average run times for the meta-heuristic can be estimated. As shown in the following figure, when the problem size reaches 1000 jobs with 50 groups, the computational time required increases dramatically.

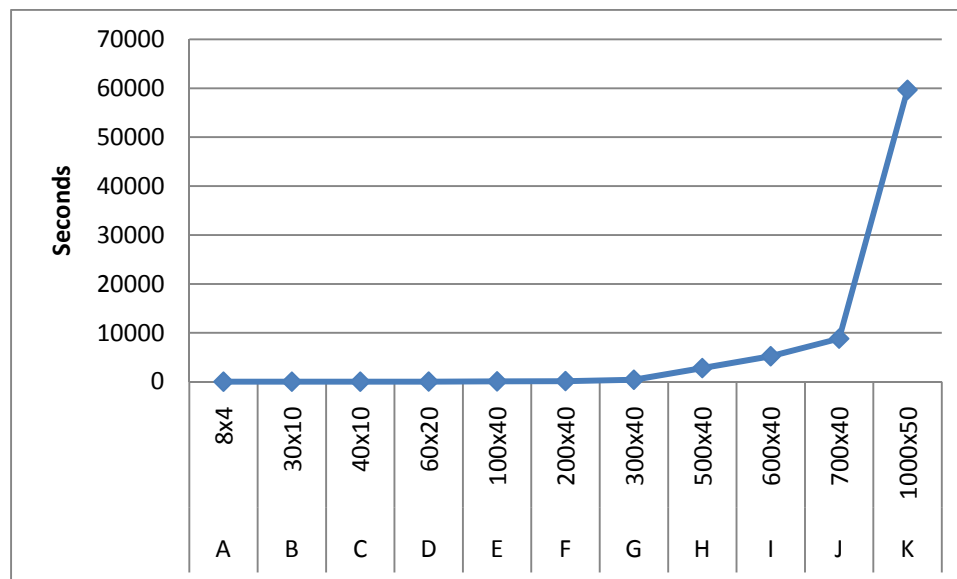


Figure 6-15 Average run times for meta-heuristic

For problems instances of size one (PS1), the meta-heuristic practically produces the same result as LINGO solver and for problem instances of size two (PS2), the percentage of deviation from optimal value is less than 5% in all cases.

Figure 6-16 shows a comparison of the percentage of deviation from optimal value with average, standard deviation and maximum value of the solution quality of the two heuristic approaches after solving 30 cases.

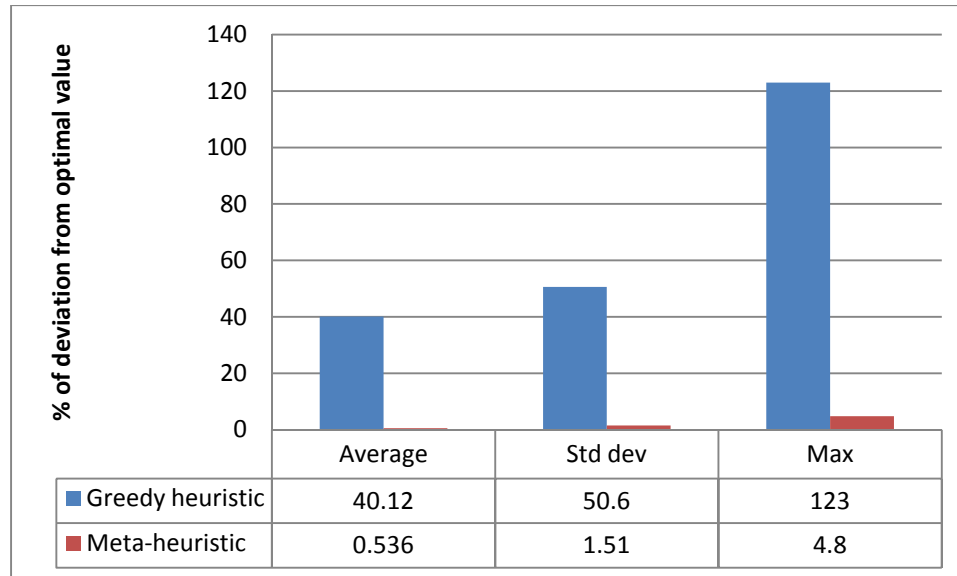


Figure 6-16 Average, std dev and max percentage of solution

As shown in previous results, the proposed meta-heuristic algorithm (based on Meta-RaPS) effectively improves the solution quality of the problem under study. The meta-heuristic also has the following advantages:

1. The results from this experimentation (Table 6-21) conclude that the meta-heuristic is able to introduce randomness and enhance the solution quality of the greedy algorithm.
2. The meta-heuristic generates a feasible route at every iteration; the algorithm can be stopped at any time and a feasible solution is available.
3. The average percentage difference from optimal is consistent with other values found in the literature for meta-heuristic approaches. Therefore, this meta-heuristic is a good method for this combinatorial problem.
4. The proposed solution approach gives a reasonable (good) run time for problems involving up to 700 jobs and 40 groups.

7. CONCLUSIONS AND FUTURE WORK

This chapter discusses the results of this research according to the original questions proposed in the Research Plan (Chapter 3), presents the conclusions, and finally proposes possible future directions for the research.

7.1 Summary

There exist abundant reports in literature on the subject of organization theories, organizational structure design, and the span of control problem. This latter alone has attracted voluminous reports in the literature. Most of these efforts are empirical in nature and are all qualitative. Few mathematical models were found in the literature and they were developed taking into account specific parameters for certain types of organizations, such as: relative wage rates between supervisors and subordinates, wage scales and communication time ratios (between and within hierarchies). These studies did not allow their use for general design of an organizational structure. This dissertation aims to study the following issues, and to reduce the research gap in this field:

1. The feasibility of optimal design (or re-design) of the organizational structure based on a quantitative model
2. The impact of organizational decisions on operation costs
3. The customization of the proposed model for different industrial domains of application, with the instruments developed for measuring the parameters in the proposed quantitative model
4. The development of a practical solution approach for large-size industrial problems

To address these issues, a novel mathematical formulation was developed to model the problem of interest. This model consists of constraints that govern the decision process and allows optimizing the span of control decision, from the allocation of jobs to workers to the management layers of the corporate structure. The proposed model optimizes the entire organizational hierarchy by eliminating unnecessary organizational units, instead of optimizing local spans of control.

The model aims at minimizing the cost of work and supervision by eliminating layers and units of the organization that are not needed based on the allocation of jobs to worker groups and the span of control needed for management. This model assigns jobs to workers, such that the total span of control costs is minimized, helping the decision maker to determine the possible impact on organization costs.

Customization of the organizational structure design model has been made through the analysis of the factors of supervision effort in the context of Information System (IS) companies. Therefore, an implementation of this model was made with a particular group of companies resulting in the selection of a feasible span of control for this type of organizations.

A commercial software system, LINGO, was used to help validate the proposed binary integer model, study its behavior, and explore its computational limits. Extensive experiments were conducted to validate the model with practical problems and evaluate the impact of changing the complexity and compatibility parameters at level 0 and managerial-related parameters for planning, training, execution, and control at the higher

levels. This experimentation showed the commercial tool (LINGO) was limited to solving small-size problems. The system parameters, including the number of jobs, worker groups, supervisors and managers, all have significant impact on the computational efficiency and the computational time grows exponentially with the number of variables in the system. That is, increasing the number of jobs and groups dramatically increases computational requirements.

A heuristic and meta-heuristic were proposed for solving large-size integer programming problems in a practical computational time. The binary integer model was decomposed into two sub-problems. One assigns jobs to working groups, while the other focuses on the higher levels of managerial hierarchy. The proposed solution approach was based on the concept of greedy algorithms and the Meta-RaPS algorithm. In this research, for each of the decisions, a two phase approach was proposed that first assigns jobs to workers with a heuristic, and then optimizes the results with a meta-heuristic.

For evaluation purposes, two-size problems were tested: problem size one (PS1) with 8 jobs, 4 groups, 3 supervisors and 2 managers, and problem size two (PS2) with 30 jobs, 10 groups, 5 supervisors and 3 managers. An optimal value was required to serve as the benchmark for the heuristic performance evaluation. The solution of the meta-heuristic was compared with the optimal solution solved by LINGO. The measurement was the percentage of deviation from the optimal value found on the solver. The results show that the quality of the solution found using this meta-heuristic was a good choice for this type of problem, and the computational time required was not excessively high when the size of the problem was not more than 300 activities with 40 working groups. Computational

time for problems with more variables begins to increase dramatically. Test of the heuristic showed that it generates good solutions in an efficient manner for applications of large-size.

7.2 Future work

A roadmap for future research is outlined in Figure 7.1. This research could be extended by improving the decision model of its decision factors, job allocations, and supervisory efforts. Coefficient evaluation is needed for application to other industries, such as construction, manufacturing, education, and so on. The proposed solution approach could also be enhanced by improving the meta-heuristic framework.

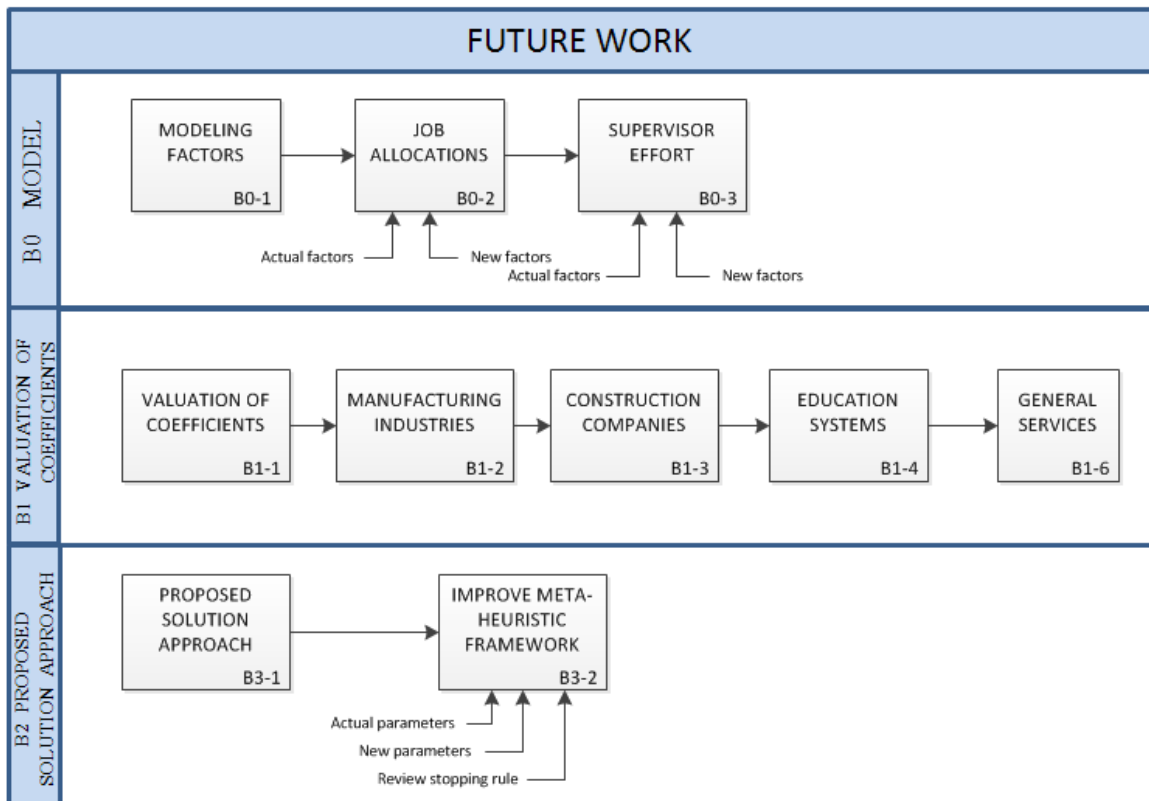


Figure 7-1 Future work

More specifically, this research could be extended to the following three major areas:

- Exploring additional factors regarding job allocations and supervisory efforts
- Developing evaluation methods to assess supervisor efforts for various industries
- Further improving performance of the proposed solution approach

In the first area, the following questions are relevant: besides complexity and compatibility, are there any other factors regarding the feasibility of job allocations? Are there other factors affecting supervision efforts besides the activities related with planning, control, execution, and training?

In the second area, future research could focus on developing an evaluation method, based on the one proposed in this research, of the factors found for the supervision efforts, to determine their coefficient values in other type of industries, such as manufacturing, construction, education systems or general services. These new developments will allow continuing the customization of the general organizational structure design model proposed in this research.

For the third area, the reason for selecting the method Meta-RaPS above other meta-heuristics was due to the results reported in recent research on the solution quality and the ease of implementation for this type of problem. It could be fruitful to continue exploring the structure of the meta-heuristic in order to improve its performance. Relevant questions regarding the meta-heuristic are: is there other stopping rule in order to improve run time? Are there others parameters besides %ret and %pry in order to improve the performance of the heuristic? Also it is worth examining other meta-heuristics to review the possibility of improving the computation time without affecting the quality of the solution found.

LIST OF REFERENCES

- [1] Van Fleet, D., and Bedeian, A. "A History of the Span of Management". The Academy of Management Review. Vol. 2, No. 3. (1977): 356-372.
- [2] Meier, K., and Bohte, J. "Span of Control and Public Organizations: Implementing Luther Gulick's Research Design". Public Administration Review. Vol. 63, No. 1, Jan/Feb 2003.
- [3] Theobald, N. A., and Nicholson-Crotty, S. "The many faces of Span of Control: Organizational Structure Across Multiple Goal". Administration & Society. 36, 6 (2005): 648-657.
- [4] Smith, Adam. The Wealth of Nations. Simon and Brown: Publishers, 2010.
- [5] Taylor, F. W. The Principles of scientific management. New York and London: Harper Brothers Publishers, 1911.
- [6] Fayol, Henri, et al. General and Industrial Management. Pitman Publishing, 1972.
- [7] Blau, P. M., and Scott, W. R. "Formal Organizations: A Corporative Approach". The American Journal of Sociology. Stanford Business Books, 1968.
- [8] Kulik, Brian and Timothy Baker. "Putting the organization back into computational organization theory: a complex Perrowian model of organizational action". Computational and Mathematical Organization Theory. Vol. 14, Iss. 2. Boston, 2008.
- [9] Woodward, J. Industrial Organization: Theory and Practice. 2nd Edition. New York: Oxford University Press, 1980.
- [10] Choo, C. W. "The knowing organization: how organizations use knowledge to construct meaning". International Journal of Information Management. 16, 5 (1996): 329-340.
- [11] Price, Alan. Human Resource Management in a Business Context. Cengage Learning Business Press, 2007.
- [12] Burton, R. M. and B. Obel. Strategic Organizational Diagnosis and Design: The Dynamics of Fit. Boston, MA: Kluwer, 2004.

- [13] Lim, M., et al. "Organizational structure for the twenty-first century". The Institute for Operations Research and The Management Sciences. Presented at the annual meeting, Austin, 2010.
- [14] Mintzberg, H. Structure in Fives: Designing Effective Organizations. Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [15] Pavett, Cynthia M., and Alan W Lau. "Managerial work: The influence of hierarchical level and functional specialty". Academy of Management Journal. 26, 1986.
- [16] Davis, J. H. "Group decision and social interaction: Theory of social decision schemes". Psychological Review. 80, (1973): 97-125.
- [17] Katz, R., and M. Tushman. "Communication patterns, project performance and task characteristics: An empirical evaluation and integration in an R&D setting". Organ. Behavior and Human Performance 23 (1979): 139–162.
- [18] Entwisle, D., and Walton, J. "Observations on the Span of Control". Administrative Science Quarterly, Vol. 5, No. 4, 1961.
- [19] Udell, J.G. "An Empirical Test of Hypotheses Relating to Span of Control". Administrative Science Quarterly, Vol. 12, No. 3. (1967): 420-439.
- [20] Bell, G. "Determinants of Span of Control." American Journal of Sociology. Vol. 37, No. 1, July 1967.
- [21] Ouchi, W., and Dowling, J. "Defining the Span of Control". Administrative Science Quarterly, Vol. 12, No. 3, 1974.
- [22] Meier, K., and Bohte, J. "Ode to Luther Gulick: Span of Control and Organizational Performance," Administration and Society, 32, 2 (2000): 115-137.
- [23] Nickols, F. "The Span of Control and the Formulas of V.A. Graicunas". Distance Consulting. 2003. <<http://www.nickols.us/graicunas.pdf>>.
- [24] Urwick, L. "The Manager's Span of Control". Harvard Business Review. (1956): 34:39-47.
- [25] Mackenzie, K. "Measuring a person's capacity for Interaction in a Problem Solving Group". Organizational Behavior and Human Performance. 12 (1974): 149-169.

- [26] Keren, M., and D Levhari. "The Optimal Span of Control in a Pure Hierarchy". Management Science. Vol. 25, No. 11, (1979): 1162-1172.
- [27] Nasrallah, W., and R Levitt. "An Interaction value Perspective on Firms and Differing Size". Computational and mathematical Organizational Theory. 7 (2001): 113-144.
- [28] Yassine, Ali, et al. "Simple Models of Hierarchical Organizations". IlliGAL Report No. 2005015, University of Illinois at Urbana-Champaign, 2005 <<http://www-illigal.ge.uiuc.edu>>.
- [29] Depuy, Gail, et al. "Workforce training schedule for logistic skills". IIE Annual Conference. Proceedings. Norcross: 2006.
- [30] LeBlanc, L.J., et al. "Heery International's spreadsheet optimization model for assigning managers to construction projects," Interfaces. 30, 6 (2000): 95-106.
- [31] Awad, R.M. and J.W. Chinneck. "Proctor assignment at Carleton University," Interfaces, 28, 2 (1998): 58-71.
- [32] Foulds, LR. "The heuristic problem-solving approach". Journal of the Operational Research Society 34 (1983): 927-934.
- [33] Il'ev, V. P. "Performance guarantees of greedy algorithms for the problems on hereditary systems". Journal of Applied and Industrial Mathematics. Dordrecht: Vol. 3, Iss. 1; p. 68, 2009.
- [34] MCGovern, S. M., and S. M. Gupta. "Combinatorial optimization analysis of the unary NP-complete disassembly line balancing problem". International Journal of Production Research. Vol. 45, Iss. 18/19; pg. 4485, London, 2007.
- [35] Agnihotri, Ameya R. Combinatorial optimization techniques for VLSI placement. State University of New York at Binghamton: 2007.
- [36] Angelopoulos, Spyridon. The approximation power of priority algorithms. University of Toronto (Canada): 2006.
- [37] Haouari, M. and J. S. Chaouachi. "A probabilistic greedy search algorithm for combinatorial optimisation with application to the set covering problem". The Journal of the Operational Research Society. Vol. 53, Iss. 7. Oxford, 2002.

- [38] Ahuja, Ravindra, et al. "A greedy genetic algorithm for the quadratic assignment problem". Computers and Operations Research. Vol. 27, Iss. 10. New York, 2000.
- [39] Thompson, G.M. "Assigning telephone operators to shifts at New Brunswick Telephone Company". Interfaces. 27, 4 (1997): 1-11.
- [40] Glover F, and M. Laguna. Tabu Search. Kluwer Academic Publishers, 1997.
- [41] DePuy et al. "Using The Meta-Raps Approach to Solve Combinatorial Problems". Proceedings of the 2002 Industrial Engineering Research Conference, May 19-21, Orlando, Florida, 2002.
- [42] DePuy, Gail and G E Whitehouse. "Applying the COMSOAL computer heuristic to the constrained resource allocation problem". Computers & Industrial Engineering. New York., Vol. 38, Iss. 3, 2000.
- [43] DePuy, Gail, et al. "Using The Meta-Raps Approach To Solve Combinatorial Problems". IIE Annual Conference. Proceedings. Norcross: 2002.
- [44] Hepdogan, S et al. "A Metaheuristic for the Early/Tardy Single Machine Scheduling Problem". IIE Annual Conference. Proceedings. Norcross: 2006.
- [45] Mora, Rubén D., et al. "Applying Multiple Heuristics to the QAP". IIE Annual Conference. Proceedings. Norcross: 2006.
- [46] Hepdogan, Seyhun, et al. "Solving the 0-1 Multidimensional Knapsack Problem". IIE Annual Conference. Proceedings. Norcross: 2006.
- [47] Keith Sinkhorn E., and Xiao Wang. "The Savings Priority Rule in the Meta-RaPS Traveling Salesman Problem Solution". IIE Annual Conference. Proceedings. Norcross: 2006.
- [48] Song, J. et al. "A Meta-heuristic for the multi-Vehicle Dial-A-Ride Problem with Time Windows". IIE Annual Conference. Proceedings. Norcross: 2004.
- [49] Moraga, Reinaldo, et al. "A Meta-Heuristic and Oscillation Improvement Strategy for the 0-1 Multidimensional Knapsack Problem". IIE Annual Conference. Proceedings. Norcross: 2004.
- [50] İyigün, M. "A Decision Support System for R&D Project Selection and Resource Allocation Under Uncertainty". Project Management Journal 24, 4, 1993.

- [51] Raisch, S. "Balanced structures: Designing organizations for profitable growth". Long Range Planning 41, 5 (2008):483–508.
- [52] Anand, N. and Daft Richard L. "What is the Right Organization Design?". Social Science Research Network. December 7, 2006 <<http://ssrn.com/abstract=961013>>.
- [53] Al-Hindawe, J. "Considerations when constructing a semantic differential scale". La Trobe Papers in Linguistics. La Trobe University, 1996.
- [54] Moraga, Reinaldo, et al. "Meta-RaPS approach for the 0-1 Multidimensional Knapsack Problem". Computers & Industrial Engineering. New York: Vol. 48, Iss. 1, 2005.
- [55] Hales. C.P. "What do managers do? A critical review of the evidence". Journal of Management Studies. (1986): 23: 88–115.
- [56] Pugh, D. S., ed. Organization Theory: Selected Readings. Harmondsworth: Penguin, 1990.
- [57] Armandi, B.R., and Wills, E.W. "Organizational Size, Structure, and Efficiency: A Test of a Blau-Hage Model". American Journal of Economics and Sociology. Vol. 41, No. 1 (1982): 43-60.
- [58] Khaled, El Emam, and Nazim H. Madhavji. "The reliability of measuring organizational maturity". Software Process Improvement and Practice, John Wiley & Sons, 1995.
- [59] Crowley, Martha et al. "Neo-Taylorism at Work: Occupational Change in the Post-Fordist Era". Social Problems. Vol. 57, Iss. 3, Berkeley (2010): 421, 27.
- [60] Klein, E. "Using information technology to eliminate layers of bureaucracy," The National Public Accountant. Vol. 46, Iss. 4 (2001): 46-56.
- [61] Qian, Y. "Incentives and Loss of Control in an Optimal Hierarchy". Review of Economic Studies. 61, 3 (1994): 527-544.
- [62] Beckmann, M. Tinbergen Notes on Organization Theory. Berlin: Springer-Verlag, 1988.
- [63] Jonker, R. and Volgenant, A. "A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems". Computing, 38, 4 (1987): 325-340.

- [64] Jackson, E, et al. “Logistics Skills Management Heuristics”. Proceedings of the 2008 Industrial Engineering Research Conference, J Fowler and S. Mason, eds., 2008.
- [65] Tang, Jiafu et al. “Heuristics-Based Integrated Decisions for Logistics Network Systems”, Journal of Manufacturing Systems. 23, 1, 2004.
- [66] Phelps, Selcen and Murat Koksalan. “An Interactive Evolutionary Metaheuristic for Multiobjective Combinatorial Optimization”; Management Science; 49,12, 2003.
- [67] Gamvros, Ioannis, et al. “The Multilevel Capacitated Minimum Spanning Tree Problem”. INFORMS Journal on Computing; Vol 18, No 3 (2006): 348-365.
- [68] Silver, EA. “An overview of heuristic solution methods”. Journal of the Operational Research Society 55 (2004): 936–956.
- [69] French, Alan, and John Wilson. “Heuristic Solution Methods for the Multilevel Generalized Assignment Problem”. Journal of Heuristics. 8, 2, 2002.
- [70] Glover, F., et al. “Improved Computer-Based Planning Techniques – Part II”. Interfaces 9, 4 (1979): 12-20.
- [71] Boehm, B. W. Software Engineering Economics. Prentice-Hall, 1981.
- [72] Nemetz, P.L. and Fry, L.W. “Flexible Manufacturing Organizations: Implications for Strategy Formulation and Organization Design”. The Academy of Management Review. Vol. 13, No. 4. (1988): 627-638.
- [73] Hareton, K.N., et al. “Automated support of software quality improvement”. The International Journal of Quality & Reliability Management. Vol. 24, Iss. 3; p. 230. Bradford: 2007.
- [74] Humphrey, Watts S. “Using a defined and measured personal software process”. IEEE Software. Vol. 13, Iss. 3; p. 77. Los Alamitos: 1996.
- [75] Kennington, J. and Z. Wang. “A Shortest Augmenting Path Algorithm for the Semi-Assignment Problem”. Operations Research, Vol. 40, No. 1 (1992): 178-187.
- [76] Donaldson, L. The Contingency Theory of Organizations, Thousand Oaks, CA: SAGE Publications, 2001.
- [77] Claessens, B., et al. “Planning behavior and perceived control of time at work”. Journal of Organization Behavior , 93, 2004.

- [78] Cullen, J.B. and Baker, D.D. "Administration size and organization size: an examination of the lag structure". The Academy of Management Journal, Vol. 27 (1984): 644-653.
- [79] Garicano, L. "Hierarchies and the organization of knowledge in production". Journal of Political Economy. (2000): 874–904.
- [80] Salas, E., et al. "Does Crew Resource Management Training Work? An Update, an Extension, and Some Critical Needs". Journal of Human Factors. (2006): 392-403.
- [81] Akella, Devi. "A reflection on critical management studies". Journal of Management and Organization (2008): 100–110.
- [82] Carroll, S. J., and D. J. Gillen. "Are classical management functions useful in describing managerial work". Academy of Management Review. (1987): 38–51.
- [83] Fournier, V and C. Grey. "At the critical moment: Conditions and prospects for critical management studies". SAGE Journal Human Relations 53, 1 (2000):7–32.
- [84] Jinghong Liang, Pierre, et al. "Optimal Team Size and Monitoring in Organizations". The accounting review. Vol. 83, No. 3, (2008): 789–822.
- [85] Davison, B. "Management span of control: how wide is too wide?". Journal of Business Strategy. (1993): 24, 22-29.
- [86] Hoffer Gittell, Jody. "Supervisory Span, Relational Coordination and Flight Departure Performance: A Reassessment of Post-bureaucracy Theory". Organization Science. Vol. 12, No. 4, (2001): 468–483.
- [87] DePuy, Gail, et al. "Skills Management Assignment Problem Heuristics". IIE Annual Conference. Proceedings. Norcross: 2009.

APPENDICES

APPENDIX A: Parameters (data), formulations in LINGO, and results of computational experiments of Chapter 4.

APPENDIX B: Formats used in the application and validations process of the valuation of factors of Chapter 5.

APPENDIX C: Complete algorithm scripts in MatLab for the meta-heuristic and heuristic of the proposed solution approach developed for Chapter 6.

APPENDIX A

Table A.1 Parameters for data for supervisors (levels 0 → 1) of case one

DATA OF SUPERVISORS		SUPERVISORS		
LEVEL 0 → 1		S1	S2	S3
CAPACITY (Hrs)		3	2	1
COST (w)		\$200	\$150	\$100
FACTORS OF SUPERVISION EFFORT	WORKERS (GROUPS)			
	G1	G2	G3	G4
PLANNING (ρ)	0.02	0.03	0.1	0.11
CONTROL (c)	0.015	0.1	0.2	0.12
TRAINING (t)	0.01	0.14	0.01	0.01
EXECUTION (e)	0.02	0.11	0.02	0.05

Table A.2 Parameters for data for levels 1 → 2 of case one

DATA OF MANAGERS		MANAGERS	
LEVEL 1 → 2		M1	M2
CAPACITY (Hrs)		5	1
COST (w)		\$450	\$500
FACTORS OF MANAGERS EFFORT	SUPERVISORS		
	S1	S2	S3
PLANNING (ρ)	0.25	0.16	0.24

CONTROL (c)	0.2	0.17	0.25
TRAINING (t)	0.15	0.15	0.2
EXECUTION (e)	0.25	0.25	0.2

Example of formulation in LINGO Software (Case 1)

$$\text{Min : } \sum_{n=1}^N \sum_{j=1}^J w_{nj} f_{nj} y_{njp_n}$$

$$\text{MIN } 2200 Y11 + 1704 Y12 + 3000 Y13 + 2504 Y14 + 3750 Y15 + 2250 Y16 + 5400 Y17 + 11500 Y21 + 12000 Y22 + 16000 Y23 + Y01 + Y02 + Y03 + Y04$$

Constraints

SUBJECT TO

- 1) Each job (operation type) is assigned to only one direct labor group

$$\sum_j z_{ij} = 1$$

$$Z11 + Z12 + Z13 + Z14 = 1$$

$$Z21 + Z22 + Z23 + Z24 = 1$$

$$Z31 + Z32 + Z33 + Z34 = 1$$

$$Z41 + Z42 + Z43 + Z44 = 1$$

- 2) Each direct labor group has a finite capacity, all job assignments are limited by the capacity of each labor group.

$$\sum_i \beta_i \alpha_{ij} \eta_{ij} z_{ij} \leq r_{0j} y_{0jp_0}$$

$$0.4 Z_{11} + 1.28 Z_{21} + 1.6 Z_{31} + 1.6 Z_{41} - 1.3 Y_{01} \leq 0$$

$$0.5 Z_{12} + 0.4 Z_{22} + Z_{32} + Z_{42} - 4.8 Y_{02} \leq 0$$

$$0.7 Z_{13} + 1.12 Z_{23} + 0.7 Z_{33} + 1.4 Z_{43} - 4.3 Y_{03} \leq 0$$

$$0.7 Z_{14} + 1.12 Z_{24} + 1.4 Z_{34} + 0.7 Z_{44} - 3.8 Y_{04} \leq 0$$

3 and 4) The total managerial work requirement of its child units cannot exceed the finite capacity imposed on the parent unit.

$$\sum_j r_{0j} y_{0jp_0} \alpha_{0j} (\rho_{1j} + c_{1j} + t_{1j} + e_{1j}) / \omega_{0j} \leq \sum_j f_{1j} y_{1jp_1}$$

$$1.04 Y_{01} + 1.5 Y_{02} + 1.505 Y_{03} + 1.995 Y_{04} - 11 Y_{11} - 8 Y_{12} - 12 Y_{13} - 8 Y_{14} - 10 Y_{15} - 6 Y_{16} - 12 Y_{17} \leq 0$$

$$\sum_j f_{n-1,j} y_{n-1,jp_{n-1}} (\rho_{nj} + c_{nj} + t_{nj} + e_{nj}) / \omega_{nj} \leq \sum_j f_{nj} y_{njp_n}$$

$$11 Y_{11} + 8 Y_{12} + 12 Y_{13} + 8 Y_{14} + 10 Y_{15} + 6 Y_{16} + 12 Y_{17} - 20 Y_{21} - 20 Y_{22} - 25 Y_{23} \leq 0$$

5 and 6) Each child unit must have a parent unit.

$$y_{0jp_0} \leq \sum_{p_1} y_{1p_0p_1}$$

$$Y_{01} - Y_{11} - Y_{12} - Y_{13} - Y_{14} - Y_{15} - Y_{16} - Y_{17} \leq 0$$

$$Y_{02} - Y_{11} - Y_{12} - Y_{13} - Y_{14} - Y_{15} - Y_{16} - Y_{17} \leq 0$$

$$Y_{03} - Y_{11} - Y_{12} - Y_{13} - Y_{14} - Y_{15} - Y_{16} - Y_{17} \leq 0$$

$$Y_{04} - Y_{11} - Y_{12} - Y_{13} - Y_{14} - Y_{15} - Y_{16} - Y_{17} \leq 0$$

$$y_{n-1,jp_{n-1}} \leq \sum_{p_n} y_{np_{n-1}p_n}$$

$$Y_{11} - Y_{21} - Y_{22} - Y_{23} \leq 0$$

$$Y12 - Y21 - Y22 - Y23 \leq 0$$

$$Y13 - Y21 - Y22 - Y23 \leq 0$$

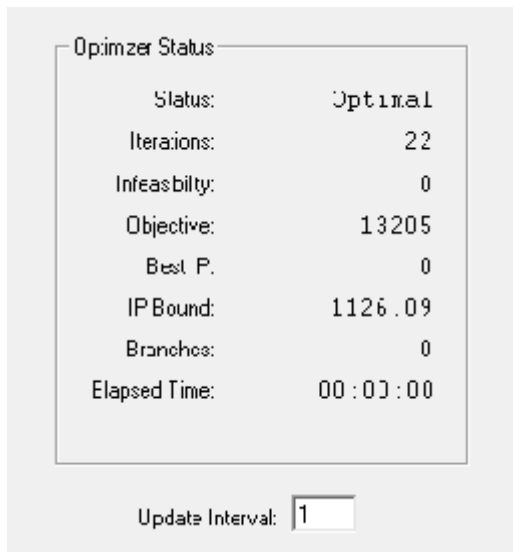
$$Y14 - Y21 - Y22 - Y23 \leq 0$$

$$Y15 - Y21 - Y22 - Y23 \leq 0$$

$$Y16 - Y21 - Y22 - Y23 \leq 0$$

$$Y17 - Y21 - Y22 - Y23 \leq 0$$

Results from LINGO Software.



Results		
INT Z11	INT Z32	INT Y13
INT Z12	INT Z33	INT Y14
INT Z13	INT Z34	INT Y21
INT Z14	INT Z41	INT Y22
INT Z21	INT Z42	INT Y23
INT Z22	INT Z43	INT Y01
INT Z23	INT Z44	INT Y02
INT Z24	INT Y11	INT Y03
INT Z31	INT Y12	INT Y04

Example of formulation in LINGO Software (Case 2)

Objective function

$$\text{Min} \quad \sum_{n=1}^N \sum_{j=1}^J w_{nj} f_{nj} y_{njp_n}$$

Min

$$72000 Y11 + 67200 Y12 + 72000 Y13 + 172800 Y21 + 156000 Y22 + 216000 Y23 + Y01 + Y02 + Y03 + Y04$$

Constraints

$$1. \quad \sum_j z_{ij} = 1 \quad \underline{i \in I}$$

$$Z11 + Z12 + Z13 + Z14 = 1$$

$$Z21 + Z22 + Z23 + Z24 = 1$$

$$Z31 + Z32 + Z33 + Z34 = 1$$

$$Z41 + Z42 + Z43 + Z44 = 1$$

$$Z51 + Z52 + Z53 + Z54 = 1$$

$$Z61 + Z62 + Z63 + Z64 = 1$$

$$Z71 + Z72 + Z73 + Z74 = 1$$

$$Z81 + Z82 + Z83 + Z84 = 1$$

$$2. \quad \sum_i \beta_i \alpha_{ij} z_{ij} / \eta_{ij} \leq r_{0j} y_{0jp_0} \quad \underline{j \in J}$$

$$\frac{\beta_1 \alpha_{11}}{\eta_{11} z_{11}} + \frac{\beta_2 \alpha_{21}}{\eta_{21} z_{21}} + \frac{\beta_3 \alpha_{31}}{\eta_{31} z_{31}} + \frac{\beta_4 \alpha_{41}}{\eta_{41} z_{41}} \leq r_{01} y_{01}$$

$$\frac{\beta_1 \alpha_{12}}{\eta_{12} z_{12}} + \frac{\beta_2 \alpha_{22}}{\eta_{22} z_{22}} + \frac{\beta_3 \alpha_{32}}{\eta_{32} z_{32}} + \frac{\beta_4 \alpha_{42}}{\eta_{42} z_{42}} \leq r_{02} y_{02}$$

$$\frac{\beta_1 \alpha_{13}}{\eta_{13} z_{13}} + \frac{\beta_2 \alpha_{23}}{\eta_{23} z_{23}} + \frac{\beta_3 \alpha_{33}}{\eta_{33} z_{33}} + \frac{\beta_4 \alpha_{43}}{\eta_{43} z_{43}} \leq r_{03} y_{03}$$

$$\frac{\beta_1 \alpha_{14}}{\eta_{14} z_{14}} + \frac{\beta_2 \alpha_{24}}{\eta_{24} z_{24}} + \frac{\beta_3 \alpha_{34}}{\eta_{34} z_{34}} + \frac{\beta_4 \alpha_{44}}{\eta_{44} z_{44}} \leq r_{04} y_{04}$$

$$0.2 Z11 + 0.3 Z21 + 0.5 Z31 + 0.2 Z41 + 0.8 Z51 + 0.75 Z61 + 0.1 Z71 + 0.15 Z81 - 1 Y01 \leq 0$$

$$0.16 Z12 + 0.24 Z22 + 40 Z32 + 0.8 Z42 + 0.64 Z52 + 60 Z62 + 0.8 Z72 + 0.12 Z82 - 1 Y02 \leq 0$$

$$18 Z13 + 0.27 Z23 + 45 Z33 + 0.9 Z43 + 0.72 Z53 + 0.68 Z63 + 9 Z73 + 0.14 Z83 - 1 Y03 \leq 0$$

$$0.2 Z14 + 30 Z24 + 0.5 Z34 + 1 Z44 + 0.8 Z54 + 0.75 Z64 + 0.1 Z74 + 0.1 Z84 - 1 Y04 \leq 0$$

$$3. \quad \sum_j r_{0j} \alpha_{ij} (\rho_{1j} + c_{1j} + t_{1j} + e_{1j}) y_{0jp_0} / \omega_{0j} \leq \sum_j f_{1j} y_{1jp_1}$$

$$\frac{r_{01}y_{01}\alpha_{01}(\rho_{11} + c_{11} + t_{11} + e_{11})}{\omega_{01}} + \frac{r_{02}y_{02}\alpha_{02}(\rho_{12} + c_{12} + t_{12} + e_{12})}{\omega_{02}} + \frac{r_{03}y_{03}\alpha_{03}(\rho_{13} + c_{13} + t_{13} + e_{13})}{\omega_{03}} + \frac{r_{04}y_{04}\alpha_{04}(\rho_{14} + c_{14} + t_{14} + e_{14})}{\omega_{04}} \leq f_{11}y_{11} + f_{12}y_{12} + f_{13}y_{13}$$

$$0.55 Y01 + 8 Y02 + 27 Y03 + 26 Y04 - 360 Y11 - 240 Y12 - 240 Y13 \leq 0$$

$$4. y_{0jp_0} \leq \sum y_{1p_0p_1}$$

$$Y01 - Y11 - Y12 - Y13 \leq 0$$

$$Y02 - Y11 - Y12 - Y13 \leq 0$$

$$Y03 - Y11 - Y12 - Y13 \leq 0$$

$$Y04 - Y11 - Y12 - Y13 \leq 0$$

$$5. \sum_j f_{n-1,j}(\rho_{nj} + c_{nj} + t_{nj} + e_{nj})y_{n-1,jp_{n-1}} / \omega_{ij} \leq \sum_j f_{nj}y_{njp_n}$$

$$\frac{f_{11}y_{11}(\rho_{21} + c_{21} + t_{21} + e_{21})}{\omega_{21}} + \frac{f_{12}y_{12}(\rho_{22} + c_{22} + t_{22} + e_{22})}{\omega_{22}} + \frac{f_{13}y_{13}(\rho_{23} + c_{23} + t_{23} + e_{23})}{\omega_{23}} + \frac{f_{14}y_{14}(\rho_{24} + c_{24} + t_{24} + e_{24})}{\omega_{24}} \leq f_{21}y_{21} + f_{22}y_{22} + f_{23}y_{23}$$

$$18 Y11 + 240 Y12 + 540 Y13 - 360 Y21 - 240 Y22 - 240 Y23 \leq 0$$

$$6. y_{n-1,jp_{n-1}} \leq \sum_{P_n} y_{np_{n-1}p_n}$$

$$Y11 - Y21 - Y22 - Y23 \leq 0$$

$$Y12 - Y21 - Y22 - Y23 \leq 0$$

$$Y13 - Y21 - Y22 - Y23 \leq 0$$

CAPACITY (r=HOURS AVAILABLE)	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5
GROUPS (j)	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
CAPACITY (f)				COST (W)						
SUPERVISOR	HRS			SUPERVISOR	\$					
S1	8			S1	50					
S2	8			S2	100					
S3	8			S3	150					
S4	8			S4	200					
S5	8			S5	250					
COEFFICIENTS										
	PLANNING (p)	CONTROL (c)	TRAINING (t)	EXECUTION (e)						
S* ==> G1	0.2	0.25	0.18	0.25	0.88					
S* ==> G2	0.25	0.19	0.2	0.22	0.86					
S* ==> G3	0.19	0.22	0.23	0.18	0.82					
S* ==> G4	0.23	0.23	0.19	0.2	0.85					
S* ==> G5	0.2	0.25	0.18	0.25	0.88					
S* ==> G6	0.25	0.19	0.2	0.22	0.86					
S* ==> G7	0.19	0.22	0.23	0.18	0.82					
S* ==> G8	0.23	0.23	0.19	0.2	0.85					
S* ==> G9	0.2	0.25	0.18	0.25	0.88					
S* ==> G10	0.25	0.19	0.2	0.22	0.86					
RESTRICCION 2										
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
roj (poj + coj + toj + eoj)	2.2	2.15	2.05	2.125	2.2	2.15	2.05	2.125	2.2	2.15
FUN OBJETIVO										
WnjFnj	400									
WnjFnj	800									
WnjFnj	1200									
WnjFnj	1600									
WnjFnj	2000									

Table A.5 Level 2 (DATA)

CAPACITY (h)					
SUPERVISOR	HRS				
S1	8				
S2	8				
S3	8				
S4	8				
S5	8				
MANAGER	2				
CAPACITY		COST (w)			
MANAGER	HRS	MANAGER	\$		
M1	10	M1	100		
M2	10	M2	200		
M3	10	M3	300		
COEFFICIENTS					
	PLANNING (p)	CONTROL (c)	TRAINING (t)	EXECUTION (e)	
M*=> S1	0.25	0.2	0.15	0.25	0.85
M*=> S2	0.16	0.17	0.15	0.25	0.73
M*=> S3	0.24	0.25	0.2	0.2	0.89
M*=> S4	0.21	0.26	0.22	0.18	0.863333
M*=> S5	0.20	0.28	0.24	0.16	0.883333
REST 2 (COEFF)					
flj(pij+cij+tj+ej)	6.8				
flj(pij+cij+tj+ej)	5.84				
flj(pij+cij+tj+ej)	7.12				
flj(pij+cij+tj+ej)	6.906666667				
flj(pij+cij+tj+ej)	7.066666667				
FUN OBJE					
WnjFnj	1000				
WnjFnj	2000				
WnjFnj	3000				

Table A.6 Results from LINGO

The screenshot displays the LINGO software interface with three windows open:

- LINGO Solver Status [cas30-i]**: Shows the following information:
 - Variables: Total: 318, Nonlinear: 0, Integers: 318
 - Constraints: Total: 58, Nonlinear: 0
 - Optimizer Status: State: Global Optimum, Iterations: 1459, Infeasibility: 0, Objective: 4205, Best IP: 4205, IP Bound: 4205
 - Nonzeros: Total: 756, Nonlinear: 0
 - Generator Memory Used (K): 128
 - Elapsed Runtime (hh:mm:ss): 00:00:00
- Solution Report - cas30-i**: Shows the following information:
 - Rows= 58 Vars= 318 No. integer vars= 318
 - Nonzeros= 756 Constraint nonz= 708 (318)
 - Smallest and largest elements in abs value= No. < : 27 No. =: 30 No. > : 0, Obj=MIN, Single cols= 0
 - Global optimal solution found at step:
 - Objective value:
 - Branch count:
 - Variable Value: Y11 1.00000, Y12 1.00000, Y13 0.00000
- LINGO Model - cas30-i**: Shows the MODEL: section.

APPENDIX B

Table B.1 SWOT Matrix format

	Strengths (S) 1. 2. 3.	Weaknesses (W) 1. 2. 3.
Opportunities (O) 1. 2. 3.	Strategies(SO) 1. 2. 3.	Strategies(SO) 1. 2. 3.
Threats 1. 2. 3.	Strategies(ST) 1. 2. 3.	Strategies(ST) 1. 2. 3.

Table B.2 Project Action Plan format

Project Action Plan						
Area	No	Activities	Responsible	Date		Comments
				Start	End	
				Developed		

	by	
	Supervisor	

Table B.3 Description of job profile format

Job	
Area	
General Description	
Specific Description	
1) Daily activities	
2) Periodic activities	
Responsibilities	

Table B.4 Performance format

Worker's name		Job:	
---------------	--	------	--

Evaluator's name				Date:	
	Excelent	Very Good	Good	Regular	Bad
Knowledge of the job					
Quality of work					
Quantity of work					
Working speed					
Commitment to work					
Comments					

Table B.5 Analysis of the performance of work format

Worker's name	
Evaluator's name	
Job:	

Date:	
Instructions: Check with YES/NO statements.	
<ol style="list-style-type: none"> 1. Has the ability to respond to an emergency situation 2. Usually offers new ideas 3. Usually proposes and achieves goals 4. Stays calm in difficult situations 	
Comments:	

Table B.6 Performance in relation to work performed format

Worker's name:	
Evaluator's name:	
Job:	
Date:	
Instructions: Below is a list of statements about job performance. Please check with an X all that applies.	
<ol style="list-style-type: none"> 1. Efficient use of working area 2. Is very careful about cleaning the equipment 3. Their work is specifically good 4. Working below standards 5. Changes to other activities quickly 	
Comments:	

--	--

Table B.7 Questions for supervisor's survey

<i>Question Type</i>	Question	1	2	3	4	5
<i>p</i>	I need to have my work goals and objectives defined in advance					
<i>c</i>	It is important to keep all the rules laid down in the organization					
<i>c</i>	In my work I need to take care of the details of each activity					
<i>e</i>	I need to be part of a team in order to do my work					
<i>p</i>	I need to analyze different alternatives for the same scenario					
<i>c</i>	It is very important to continuously check the appropriate course of activities					
<i>t</i>	It is important to know what other skills the people have					
<i>e</i>	In order to do my job, I need to follow specific instructions					
<i>e</i>	It is important to have experience in order to perform this job properly					
<i>c</i>	I need to make judgments on the activities of the company					
<i>t</i>	It is important to increase the learning capacity of people					
<i>p</i>	It is important to have flexibly with the working group					
<i>p</i>	In order to do my work it is necessary to update the programs continuously					
<i>c</i>	I need to evaluate the performance of people who are					

	under my charge					
<i>t</i>	It is important for the company to have training					
<i>e</i>	It is important to know the job formats established by the company					
<i>t</i>	It is important to learn new ways to perform an activity					
<i>p</i>	I need to manage the resources of my office					
<i>p</i>	It is necessary to strengthen the activities of employees					
<i>e</i>	I need to put into practice new ideas when I do my work					

APPENDIX C

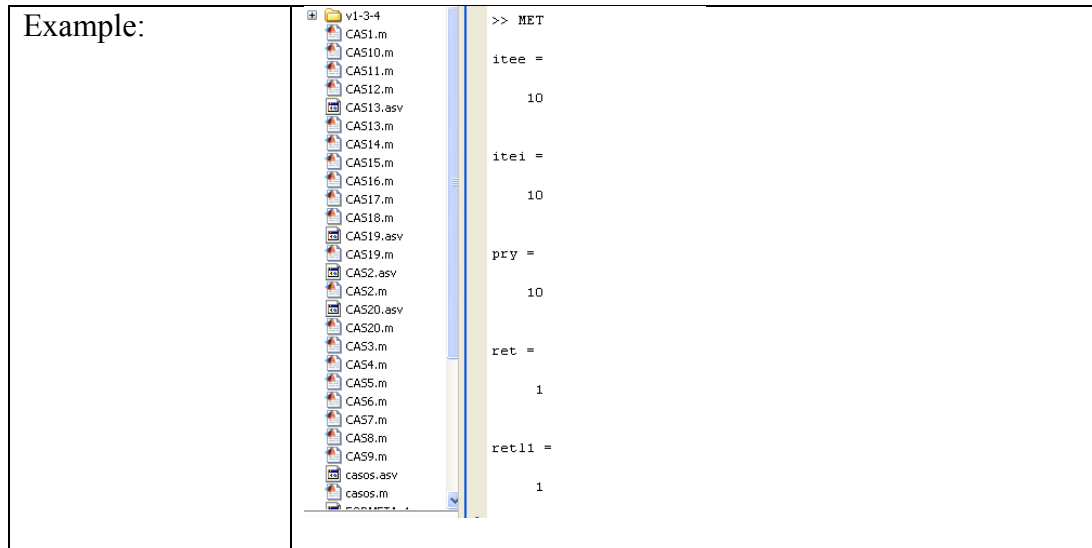
C.1 Meta-Heuristic Procedure

The meta-heuristic proposed is summarized as follows:

1. Run script MET (MET contains the parameters of the Meta-Heuristic algorithm) (see files in CD attach).

Table C.1 Scrip for MET (parameters)

Parameters:	Description
Itee	Number of iterations (repeats the modified and original greedy algorithm for a user-defined number of iterations).
Itei	Number of iterations (internal) (select an activity in order to find the activity that has a minimum value within restriction ret and ret1).
pry	Priority percentage (if p= 10%, it means that 10% of the time the greedy algorithm will be used and 90% of the time the modified algorithm will be used to select the next item)
ret	Restriction percentage to reduce the magnitude of the maximum rate in order to choose the next available activities (the variable will be picked randomly) at level 0
ret1	Restriction percentage to reduce the magnitude of the maximum rate in order to choose the next available activities (the variable will be picked randomly) at level 1 to n



2. Run Script CAS# (CAS# contains the data for cas # n)

Table C.2 Scrip for CAS (data)

Data	Description
aor	Table of time of each activity with complexity and compatibility factors $aor[i,j] \rightarrow i= \text{activity}, j= \text{group}$
b	Capacity of worker j
n	Number of groups
m	Number of activities
ns	Number of supervisors
nm	Number of managers
ctors	Cost of supervisor [1 ... ns]
cpsor	Capacity of supervisor [1...ns]
tfac	Value of factors (planning, execution, training, control) of supervisors for each worker [1 ... n]
ctmor	Cost of manager [1 ... nm]

cpmor	Capacity of manager [1 ... nm]
tfacm	Value of factors (planning, execution, training, control) of managers for each supervisor [1 ... ns]
Example:	<pre> >> CAS1 aor = 0.1200 0.1200 30.0000 20.0000 0.1800 0.2100 0.6000 18.0000 0.3000 0.4000 0.7500 0.3500 1.0000 0.9000 1.0000 0.6000 0.8000 0.8000 0.8000 0.5600 0.9000 83.0000 0.4500 0.6000 0.1200 0.1200 7.0000 0.1000 18.0000 0.1950 0.9000 15.0000 b = 1 2 1 1 ns = 3 ctsor = 200 150 100 cpsor = 3 2 1 </pre>

3. Run script L0 (level 0)

Table C.3 Scrip for L0 (level 0)

Step:	Script:
Reset variables yopt(i,j) = Assign activities (i) to groups (j) cw = assign time to each group (y2, u, gopt) : auxiliary variables for the meta-heuristic	<pre> for j=1:m for i=1:n yopt(i,j)=0; end end </pre>

	end
Set iteration k Itee = number of iterations	for k=1:itee [...] end
Find activity with minimum processing time	for i= 1:n if a(z,j)>a(i,j) z = i; else end end men = a(z,j);
Generate a random number between 1-100	PR = randi(100)
If PR <= pry, the solution approach will choose the greedy algorithm from the available list of activities (this list contains all activities that can be scheduled for any group j)	if PR<=pry % heuristic solution cw = cw + a(z,j); if cw <= b(j) y2(z,j) = 1; for x=1:m a(z,x)=tot; end else end

<p>Otherwise the solution approach will determine the next activity to be scheduled by selecting an activity at random from the available list and comparing it to the minimum value modified by %r. If the time is according to this rule, assign the activities to the group, then assign to the activities a value = tot (maximum time). Variable “cw” reduces the number of units of resource available for group resource j. This part of the algorithm uses an iteration “itei” in order to find an activity whose value is within the minimum value and the %rest.</p>	<pre> for f=1:itei c=randi(n); if a(c,j)<=a(z,j)*(1+ret) cw=cw+a(c,j); if cw<=b(j) y2(c,j)=1; % assigned value tot if the activity is selected for x=1:m a(c,x)=tot; end else end else end end end end </pre>
<p>Check if the solution is feasible with variable “fac”.</p> <p>Number of activities assigned to each group (variable “ti”)</p> <p>Verify if the group has been assigned to the solution with variable “ta”</p>	<pre> % look if the solution is factible with variable fac for j= 1:m for i= 1:n fac(k) = fac(k) + y2(i,j); end end % look if the group has one or more </pre>

	<pre> activities assigned for j=1:m ti=0; for i=1:n ti=ti+y2(i,j); end %if the group has one or more activities assigned the %value ta is equal to 1 if ti>0 ta(j)=1; else end end end </pre>
<p>Compute how many groups were assigned with variable “opt”</p>	<pre> for j=1:m opt(k) = opt(k)+ta(j); end </pre>
<p>If the solution is feasible, variable “yopt” will have the assigned activities with their group.</p> <p>Assign the number of groups selected (if the solution is feasible) to variable “gopt”.</p> <p>These processes will choose the minimum groups assigned in order to minimize the objective function.</p>	<pre> %is solution is feasible, then assign yopt, gopt in mem =k if fac(k)==n u=u+1; if u==1 yopt=y2; </pre>

	<pre> gopt=opt(k); mem=k; else % choose iteration with minimum opt if opt(k)<=opt(mem) yopt=y2; gopt=opt(k); mem=k; else end end else end </pre>
<p>Variable “tb” has the “best solution” within the iterations k</p>	<pre> % look for the best solution within the iterations for j=1:m tb(j)=0; end for j=1:m ti2=0; for i=1:n ti2=ti2+yopt(i,j); end </pre>

	<pre> if ti2>0 tb(j)=1; end end </pre>
<p>Example: (after running scrip MET and CAS1)</p> <p>gopt = number of groups selected in the problem (3)</p> <p>yopt = activities assigned to groups a(1,1) = 1 (activity 1 assigned to group 1) a(2,1) = 1 (activity 2 assigned to group 1) a(3,1) = 1 (activity 3 assigned to group 1) a(4,1) = 0 ... A(8,2) = 1(activity 8 assigned to group 2) ...</p> <p>tb = (group 1 assigned, group 2 assigned, group 3 assigned and group 4 unassigned).</p>	<pre> gopt = 3 yopt = 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 ta = 1 1 1 0 tb = 1 1 1 0 </pre>

4. Run Script L1 & L2

Table C.4 Scrip for L1 to Ln (level 1 to n)

Step:	Script:
Reset variables	

<p>yopcts(j) = time assigned to each supervisor cws = assign time to each supervisor ysptot(I,j) = assigned groups to supervisors</p>	<pre>for j0=1:ns yopcts(j0)=0 for i0=1:m ysptot(i0,j0)=0 end end</pre>
<p>Set iteration kt Itee = number of iterations</p>	<pre>for kt=1:itee/2 [...] end</pre>
<p>% yspt = assigning supervisors to groups % calculate rgt and rg2t % if group is not assigned, then rgt goes to 2000 % assign to each yspt the same value</p>	<pre>for j2=1:m if tb(j2)==1 rgt(j2)=b(j2)*tfac(j2) rg2t(j2)=b(j2)*tfac(j2) else rgt(j2)=2000 for k2=1:ns yspt(j2,k2)=2000 end end end</pre>
<p>Find supervisor with minimum cost</p>	<p>% the algorithm is going to choose between the assign of</p>

	<pre> % groups based on minimum cost or values between factor ret for d=1:ns wt=0 cwst=0 gs=1 for j3=1:ns if cts(gs)>cts(j3) gs=j3 end end mencts = cts(gs) </pre>
<p>Generate a random number between 1-100</p>	<pre>PRT = randi(100)</pre>
<p>If $PR \leq pry$, the solution approach will choose the greedy algorithm from the available list of supervisors. In order to assign groups to supervisors the greedy algorithm will look for the minimum processing time.</p>	<pre> if PRT<=pry % look for min group time for hx= 1:m gr=1 for j4=1:m if rgt(gr)>rgt(j4) gr=j4 end end menrg=rgt(gr) </pre>

	<pre> % assign group only if their value is minus than 2000 % groups with value of 2000 are already assigned cwst = cwst + rgt(gr) if cwst <= cps(gs) yspt(gr,gs)=1 rgt(gr)=2000 else cts(gs)=100000 end end [...]</pre>
<p>Otherwise the solution approach will determine the next supervisor to be scheduled by selecting a supervisor at random from the available list and compare with the minimum value modified by %r1. If the cost is according to this rule assign the group to supervisor, then assign to the group a value = 2000 (maximum time). Variable “cw” reduces the number of units of resource available for group resource j.</p> <p>This part of the algorithm uses an iteration “itei” in order to find a supervisor whose value is within the minimum value and the %rest1</p>	<pre> Else cf = gs for kc=1:itei/2 ct=randi(ns) if cts(ct)<=cts(gs)*(1+retl1) cf=ct end end for hx= 1:m gr=1 for j5=1:m if rgt(gr)>rgt(j5)</pre>

	<pre> gr=j5 end end menrg=rgt(gr) % assign group only if their value is < 2000 % groups with value of 2000 are already assigned cwst = cwst + rgt(gr) if cwst <= cps(cf) yspt(gr,cf)=1 rgt(gr)=2000 else cts(cf)=100000 end end end [...]</pre>
<p>Compute the time needed of supervision for each group (variable “yhrt”)</p> <p>Compute the cost related to the supervisors assigned (variable “costt”)</p>	<pre> % calculate the time needed of supervision for j11=1:ns for j21=1:m if yspt(j21,j11)==1 yhrt(j11)=yhrt(j11)+rg2t(j21); end end end</pre>

	<pre> % calculate the cost related to the supervisors assigned costt=0; for jhr=1:ns if yhrt(jhr)>0 costt=costt+(cpsor(jhr)*ctsor(jhr)); end end </pre>
<p>Verify feasibility of the solution at this level (variable “rgtot”)</p>	<pre> % if the solution is not feasible assigned a high cost rgtot=0 for j6=1:m rgtot= rgtot + rgt(j6) end if rgtot == 2000*m solf = 'FACTIBLE LEVEL' wt=0 else solf = 'NO FACTIBLE LEVEL1' wt=1 costt=100000 end </pre>

Look for the “best solution” within the iterations kt

```
ut = ut +1
if ut == 1
    yopcts = yhrt
    costot = cosot (kt)
    ysptot = yspt
    memt = kt
else
    if cosot(kt) < cosot(memt)
        yopcts = yhrt
        costot = cosot (kt)
        ysptot = yspt
        memt = kt
    end
end
```

Example L1:
 (after running scrip MET and CAS1
 and L0)

costot = minimum cost after kt
 iterations for this level

yopcts = number of hours assigned to
 the supervisor

numst = number of groups (workers)
 assigned to each supervisor

ysptot = assigned groups to
 supervisors

y(1,1) = 0
 ...
 y(1,2) = 1 (group 1 assigned to
 supervisor 2)
 y(4,2) = 2000 (group 4 was not
 selected at level 0)
 ...
 y(1,3) = 0
 ...

costot =			
	300		
yopcts =			
	0	1.1550	0
numst =			
	0	3	0
ysptot =			
	0	1	0
	0	1	0
	0	1	0
	2000	2000	2000

<p>Example L2: (after running scrip MET, CAS1, L0 y L1)</p> <p>yopcts2 = number of hours assigned to the manager</p> <p>numst = number of supervisors assigned to each manager</p> <p>ysptot = assigned supervisors to managers</p> <p>y(1,1) = 2000 (supervisor 1 was not selected at level 1)</p> <p>...</p> <p>y(2,2) = 1 (supervisor 2 assigned to manager 2)</p> <p>...</p> <p>supst2 = number of managers required at level 2</p> <p>costot2 = minimum cost after kt iterations for this level (2)</p> <p>costcas = minimum cost of the entire problem (level 0 to n)</p> <p>Ltime = seconds required to run the problem</p>	<pre> yopcts2 = 0 1.4600 numst2 = 0 1 ysptot2 = 2000 2000 0 1 2000 2000 supst2 = 1 costot2 = 2000 </pre>
--	---


```
costcas =  
    2303  
Elapsed time is 0.309355 seconds.  
L2tim =  
    0  
L2tim =  
    0.3094  
Ltime =  
    0.9231
```

C.2 Script (matlab) for heuristic (greedy algorithm) at level 0

Let J be the set of workstations, and let a_{ij} be the modified processing time of task i with respect to workstation j (including complexity and compatibility factors).

Let n , m be the number of tasks and the number of workstations.

Let tot be the maximum processing time for a_{ij} once the task is assigned to a workstation.

```
Read matrix  $a_{ij}$  (= a)
n= 8
m= 4
d=a
cw = 0
tot=200
    for j=1:m           // for all workstations reset decision variable  $y_{ij}$ 
        for i=1:n
            y(i,j)=0
        end
    end

    for j=1:m           // for all workstations
        cw=0
        for q = 1:n     // find minimum  $a_{ij}$ 
            z = 1
            for i= 1:n
                if a(z,j)>a(i,j)
                    z = i
                else
                    end
            end
            men = a(z,j) // minimum value
            cw = cw + a(z,j)
            if cw <= b(j) // if capacity it is not exceeds
                y(z,j) = 1 // assign  $y_{zj}$  task to workstation  $b_j$ 
                for x=1:m
                    a(z,x)=tot // assign to  $a_{ij}$  a maximum value for all j
                end
            else
                end
            end
        end
    end
Return matrix  $y_{ij}$ 
```

C.3 Script (matlab) for heuristic (greedy algorithm) at level 1 to n

```
% DATA ns (# supervisors) tfac = supervisor factors
% ctsor = cost/supervisor cpsor = capacity/supervisor
% START algorithm based on minimum cost CTS
% yspt = assigning supervisors to groups
% if group is not assigned, and then rgt goes to 2000
% assigns to each yspt the same value
for j2=1:m
    if tb(j2)==1
        rgt(j2)=b(j2)*tfac(j2)
        rg2t(j2)=b(j2)*tfac(j2)
    else
        rgt(j2)=2000
        for k2=1:ns
            yspt(j2,k2)=2000
        end
    end
end
% the algorithm is going to choose between the assign of
% groups based on minimum cost
for d=1:ns
    wt=0
    cwst=0
    gs=1
    for j3=1:ns
        if cts(gs)>cts(j3)
            gs=j3
        end
    end
    mencts = cts(gs) //minimum cost
    % look for min group time
    for hx= 1:m
        gr=1
        for j4=1:m
            if rgt(gr)>rgt(j4)
                gr=j4
            end
        end
        menrg=rgt(gr) //minimum time
        % assign group only if their value is minus than 2000
        % groups with value of 2000 are already assigned
        cwst = cwst + rgt(gr)
        if cwst <= cps(gs)
            yspt(gr,gs)=1
            rgt(gr)=2000
        else
            cts(gs)=100000
        end
    end
end
end
```

VITA

JESÚS A. MENA

Born, Chihuahua, Chih, México

- | | |
|-----------|--|
| 1984-1988 | B.A., Industrial Engineering
Chihuahua Institute of Technology
Chihuahua, Chih, México |
| 1989-1990 | M.S., Manufacturing Systems
Monterrey Institute of Technology
Monterrey, NL, México |
| 1991-2011 | Full Time Professor
Monterrey Institute of Technology
Campus Chihuahua
Chihuahua, Chih, México |
| 2005-2007 | Teaching Assistant
Florida International University
Miami, Florida |
| 2011 | Doctoral Candidate in Industrial and Systems Engineering
Florida International University
Miami, Florida |