A mathematical model of operation allocation and materials handling system selection problems in a flexible manufacturing system.

Jorge. Paulo
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation
https://scholar.uwindsor.ca/etd/653
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®
A Mathematical Model of Operation Allocation and Materials Handling System Selection Problems in a Flexible Manufacturing System

By

Jorge Paulo

A Thesis Submitted to the College of Graduate Studies and Research through the Industrial and Manufacturing Systems Engineering Program in Partial Fulfillment of the Requirements for the Degree of Master of Applied Science at the University of Windsor

Windsor, Ontario, Canada

© 2000, Jorge Paulo
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

0-612-52629-1
Abstract

Materials handling systems are an integrating component of a manufacturing system and as such must be considered within an integrated approach to manufacturing systems design. This work proposes to integrate the operation allocation and the materials handling system selection problems in a flexible manufacturing system by extending the operation allocation model to include some aspects of materials handling system design. The objective of the operation allocation model is to select a group of machines where the operations of the part types will be performed and then to assign those operations to the selected machines. The operation allocation model interfaces with the materials handling system selection model by providing input data in the form of the manufacturing operations to be performed at each machining center. The selection of the materials handling system is centered on the matching of the parts visiting a machining center to perform a manufacturing operation and the abilities of the handling equipment to perform the required materials handling functions of those part types. The objective is to select an optimal group of materials handling equipment to be assigned to a cell. A computer program was developed to greatly automate the process of solving the models. This allows the program to be used as a rapid modeling tool.
Dedication

This work is dedicated to my parents, Albano and Luzia Paulo, who through personal sacrifices permitted me to successfully complete this Master’s thesis. Their relentless encouragement and support will never be forgotten.
Acknowledgements

I would like to acknowledge and thank Dr. R.S. Lashkari, Dr. S.P Dutta and Dr. A. Atmani for their guidance and complete support of this work.

I would also like to thank Mrs. Mummery, Mr. Barakat and Mr. McKenzie for their technical expertise and support.
Table of Contents

ABSTRACT ............................................................................................................. III
DEDICATION ......................................................................................................... IV
ACKNOWLEDGEMENTS ...................................................................................... V
LIST OF TABLES ................................................................................................... IX
LIST OF FIGURES ................................................................................................ X
NOTATION ............................................................................................................. XI

CHAPTER 1. INTRODUCTION ............................................................................. 1
  1.1. Motivation for the Proposed Research ..................................................... 2
  1.2. Organization of the Proposal .................................................................... 4

CHAPTER 2. LITERATURE SURVEY ................................................................ 6
  2.1. Operation Allocation Literature ............................................................. 6
  2.1. Materials Handling System Selection Literature ..................................... 9
  2.2. Commentary on Previous Work ............................................................... 12

CHAPTER 3. THE MATHEMATICAL MODELS ............................................. 13
  3.1. Sub-Model 1. Operation Allocation ....................................................... 13
      3.1.1. Mathematical Formulation .............................................................. 14
  3.2. Sub-model 2. Materials Handling System Selection ................................ 18
      3.2.1. Mathematical Formulation .............................................................. 19

CHAPTER 4. CASE STUDY EXAMPLE ......................................................... 26
  4.1. The Operation Allocation Problem ....................................................... 26
  4.2. The Materials Handling System Selection Problem .............................. 35

CHAPTER 5. SENSITIVITY ANALYSIS ....................................................... 44
  5.1. Sensitivity Analysis of \( \bar{W}_{it} \) ............................................................... 45
      5.1.1. Detailed Sensitivity Analysis of \( \bar{W}_{in} \) ....................................... 49
  5.2. Sensitivity Analysis of \( W_{et} \) ................................................................. 50
List of Tables

Table 1. Rating Scale for $W_{he}$  ................................................................. 22
Table 2. Rating Scale for $W_{et}$  ................................................................. 22
Table 3. Rating Scale for $W_{it}$  ................................................................. 22
Table 4. Manufacturing operation times $t_{ij}(ip)$ and costs $OC_{ij}(ip)$, part type demands $d_i$ and machine setup costs $SC_j$ ................................................................. 27
Table 5. Cost of moving a unit of part type $i$ from machine $j$ to machine $j$, $T_{ij} (i = 1, \ldots, 14)$  ................................................................. 32
Table 6. Machine selection and operation assignments for the case study  ................................................................. 34
Table 7. Machine selection and operation assignments for case with reduced machine availability ................................................................. 34
Table 8. $W_{it}$ table for the case study example  ................................................................. 35
Table 9. $W_{et}$ table for case study example  ................................................................. 36
Table 10. $W_{he}$ table for the case study  ................................................................. 38
Table 11. Materials handling times, $t_{he}$  ................................................................. 38
Table 12. Materials handling requirements, $a_{hei}$  ................................................................. 39
Table 13. Solution to the materials handling system selection problem  ................................................................. 40
Table 14. Materials handling equipment utilization  ................................................................. 42
Table 15. $W_{it}$ sensitivity analysis for part type 2 - assignments  ................................................................. 46
Table 16. $W_{it}$ ratings for part type $i = 2$  ................................................................. 49
Table 17. Sensitivity analysis results for $W_{et} = W_{11}$  ................................................................. 53
Table 18. Sensitivity analysis results for $W_{et} = W_{12}$  ................................................................. 56
Table 19. Sensitivity analysis results for $W_{he} = W_{11}$  ................................................................. 59
Table 20. Sensitivity analysis results for $W_{he} = W_{12}$  ................................................................. 62
Table 21. Data menu entries and associated database tables  ................................................................. 67
Table 22. List of abbreviations used in sensitivity analysis of $W_{it}$  ................................................................. 89
Table 23. Results of sensitivity analysis for $W_{it}$ ratings - part type 2  ................................................................. 90
Table 24. Description of the tables in the database $OA.MDB$  ................................................................. 93
Table 25. $OA.MDB$ Database - description of the table: Parts  ................................................................. 94
Table 26. $OA.MDB$ Database - description of the table: Machines  ................................................................. 94
Table 27. $OA.MDB$ Database - description of the table: Plans  ................................................................. 94
Table 28. $OA.MDB$ Database - description of the table: MfgOps  ................................................................. 95
Table 29. $OA.MDB$ Database - description of the table: PartPlan  ................................................................. 95
Table 30. $OA.MDB$ Database - description of the table: PartPlanOp  ................................................................. 95
Table 31. $OA.MDB$ Database - description of the table: PartPlanOpMC  ................................................................. 95
Table 32. $OA.MDB$ Database - description of the table: MHC  ................................................................. 96
Table 33. $OA.MDB$ Database - description of the table: A  ................................................................. 96
Table 34. $OA.MDB$ Database - description of the table: MH_Equip  ................................................................. 96
Table 35. $OA.MDB$ Database - description of the table: MH_Op  ................................................................. 96
Table 36. $OA.MDB$ Database - description of the table: MH_Subop  ................................................................. 97
Table 37. $OA.MDB$ Database - description of the table: Tech  ................................................................. 97
Table 38. $OA.MDB$ Database - description of the table: MH_Req  ................................................................. 97
Table 39. $OA.MDB$ Database - description of the table: W_it  ................................................................. 98
Table 40. $OA.MDB$ Database - description of the table: W_et  ................................................................. 98
Table 41. $OA.MDB$ Database - description of the table: W_hhe-time  ................................................................. 98
Table 42. Source code files in the Solver program  ................................................................. 101
List of Figures

Figure 1. Main form of the Solver program ......................................................... 64
Figure 2. The File menu entries ........................................................................... 65
Figure 3. The Data menu entries .......................................................................... 66
Figure 4. Table for the Machines menu entry ..................................................... 68
Figure 5. The Reports menu entries ................................................................... 69
Figure 6. Operation Allocation report window .................................................. 70
Figure 7. The Help menu entries ........................................................................ 70
Figure 8. Dialog box illustrating the commands passed to Lingo ...................... 71
Figure 9. Dialog box informing the user about Lingo's success in solving the model ........................................................................ 72
Figure 10. Main window after both models were solved .................................... 73
Figure 11. The operation allocation problem must be solved first ..................... 73
Notation

Sub-model 1. Operation Allocation

Index Sets

\[ i \in \{1,2,...,n\} \]  
\text{part types}

\[ p \in \{1,2,...,P(i)\} \]  
\text{process plans for part type } i

\[ s \in \{1,2,...,S(ip)\} \]  
\text{operations for part type } i \text{ under process plan } p

\[ j \in J_{ip} \subset \{1,2,...,m\} \]  
\text{set of machines that can perform operation } s \text{ of part type } i \text{ under process plan } p

\[ \hat{j} \in J_{ip(s+1)} \subset \{1,2,...,m\} \]  
\text{Set of machines that can perform operation } (s+1) \text{ of part type } i \text{ under process plan } p

Decision Variables

\[ X_{sj}(ip) = \begin{cases} 1 & \text{if operation } s \text{ is processed on machine } j \text{ for the combination } (ip) \\ 0 & \text{otherwise} \end{cases} \]

\[ L_{ij}(ip) = \begin{cases} 1 & \text{if part type } i \text{ moves to machine } j \text{ to perform operation } (s+1) \text{ after performing} \\ 0 & \text{otherwise} \end{cases} \]

\[ Z(ip) = \begin{cases} 1 & \text{if part type } i \text{ is processed under process plan } p \\ 0 & \text{otherwise} \end{cases} \]

\[ M_{ij} = \begin{cases} 1 & \text{if machine } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \]

Parameters

\[ b_j \text{ = amount of time available on machine } j \]
\[ OC_{sj}(ip) \text{ = cost of performing operation } s \text{ on machine } j \text{ for the combination } (ip) \]
\[ d_i \text{ = demand for part type } i \]
\[ SC_j \text{ = setup cost of machine } j \]
\[ t_{sj}(ip) \text{ = time for performing operation } s \text{ on machine } j \text{ for the combination } (ip) \]
\[ T_{ij} \text{ = cost of moving a unit of part type } i \text{ from machine } j \text{ to machine } j \]
Index Sets

\[ i \in \{1,2,\ldots,n\} \quad \text{part types} \]
\[ s \in \{1,2,\ldots,S(i)\} \quad \text{manufacturing operations for part type } i \]
\[ h \in \{1,2,\ldots,H\} \quad \text{major materials handling operations} \]
\[ \hat{h} \in \{1,2,\ldots,\hat{H}\} \quad \text{materials handling sub-operations} \]
\[ t \in \{1,2,\ldots,T\} \quad \text{key product variables} \]
\[ e \in \{1,2,\ldots,E\} \quad \text{materials handling equipment} \]
\[ \hat{e} \in \{1,2,\ldots,\hat{E}\} \quad \text{materials handling equipment} \]
\[ j \in J_{it} \subset \{1,2,\ldots,m\} \quad \text{set of machines that can perform operation } s \text{ of part type } i \]

Decision Variables

\[ Y_{hkeit}(i) = \begin{cases} 1 & \text{if the materials handling operation/sub-operation } hh \text{ require materials handling equipment} \\ e \text{ at machine } j \text{ where manufacturing operation } s \text{ of part type } i \text{ is performed} \\ 0 & \text{otherwise} \end{cases} \]

\[ D_c = \begin{cases} 1 & \text{if materials handling equipment } e \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \]

\[ D_{\hat{c}} = \begin{cases} 1 & \text{if materials handling equipment } \hat{e} \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \]

Parameters

\[ T_e = \text{amount of time available on materials handling equipment } e \]
\[ A_{sj} = \text{manufacturing operation assignments where an instance of the parameter indicates that} \]
\[ \text{manufacturing operation } s \text{ of part type } i \text{ is performed at machine } j \]
\[ d_i = \text{demand for part type } i \]
\[ \alpha_{hkeit} = \text{materials handling operation/sub-operation requirements, where an instance of the parameter indicates that} \]
\[ \text{the material handling function } (hh) \text{ is required at machine } j \text{ where manufacturing} \]
\[ \text{operation } s \text{ of part } i \text{ is performed} \]
\[ t_{hke} = \text{time for equipment } e \text{ to perform materials handling operation/sub-operation } hh \]
\[ W_{it} = \text{relative weight of the product variable } t \text{ on part type } i \]
\[ W_{et} = \text{relative weight of the product variable } t \text{ on materials handling equipment } e \]
\[ W_{hke} = \text{relative degree of capability of materials handling equipment } e \text{ to perform the operation/sub-} \]
\[ \text{operation combination } hh \]
\[ C_{ei} = \text{compatibility factor between a piece of materials handling equipment } e \text{ and a part type } i \]
CHAPTER 1. INTRODUCTION

The environment under which manufacturing organizations operate is changing rapidly. The emerging post-industrial environment is characterized by fierce global competition, rapid market changes, shorter product life cycles and advances in both manufacturing and information technology (Vonderembse et al., 1997). These environmental changes have a direct effect on the design of manufacturing systems spawning significant changes to that design.

The widening of competition to a global realm, made increasingly possible by the emergence of various international trade agreements and facilitated by the recent advances in information technologies, places an added demand on the manufacturing systems to produce “agile” products. These products must be continuously produced to closer tolerances, with higher reliability and from an expanding variety of materials.

The shorter product life cycles and rapid market changes lead to a proliferation in the number and variety of products that must be produced within a manufacturing facility. The emphasis has shifted to producing this large array of new and improved products in smaller volumes as rapidly and economically as possible.

Other changes include the recognition of the cost of materials, materials handling and energy as significant components of the total manufacturing cost. This is especially true for materials handling. Tompkins and White (1984) estimated that between 20-50% of the total operating cost in manufacturing can be attributed to the various materials handling functions.
Advances in manufacturing and information technology make these trends seemingly feasible to accommodate. Technological advances in CAD/CAM/CIM and flexible manufacturing systems or cells can, at least in theory, provide the flexibility and efficiency required to successfully cope with the changes outlined above. However, in many cases, this has not proved to be true, or at least not to the extent that one would expect. While these advanced technologies have been successfully exploited for specific tasks, often the required system-wide benefits have not been achieved.

A manufacturing system is a complex of smaller interacting subsystems. The changes and trends discussed previously affect these subsystems and therefore the manufacturing capability of the whole system. Due to the relationships among these subsystems, it is necessary to consider the manufacturing system from an integrated standpoint. The focus on integration guarantees that critical information exchange between the subsystems is considered. This is of paramount importance to a manufacturing system that must produce a large variety of low-medium volume products in a reliable, time efficient and resource efficient manner.

1.1. Motivation for the Proposed Research

Integration should be the focus of a post-industrial manufacturing firm. As such, the manufacturing system must be viewed from a holistic perspective as an integrated whole. This “whole” is composed of a number of interacting and integrated subsystems. Each subsystem has certain objectives that it must meet; however, its operations must be fine-tuned to optimize the “whole”.
Flexible manufacturing systems or cells have the potential to provide the flexibility and efficiency required by the post-industrial firm. This has been widely recognized and has been the motivating factor behind a large volume of research in this area. This is to be expected since flexible manufacturing systems require careful planning to ensure that resources are not underutilized, flexibility is maximized, and set-up costs are minimized. One of the principal tasks to be performed is the optimal selection of machines and the allocation of part operations to the selected machines. This task is usually performed while trying to meet a number of objectives. These include minimization of operation, machine set-up and materials handling costs, minimization of total processing times, maximization of machine utilization, etc.

The materials handling system can be thought of as the basic integrator of a manufacturing system. After all, the materials handling system connects and traverses all departments and cost centers. Because of its integrating nature, all complexity inherent in a manufacturing problem is passed on to the materials handling system (Noble and Tanchoco, 1994). The high complexity of a materials handling system makes the selection of an appropriate system a non-trivial problem. Even for a modest size manufacturing system, the design details increase rapidly. There is a large variety of standard and specialty equipment, each with different characteristics that can be difficult to quantify for inclusion in a model. Further, there can be multiple objectives to consider for any specific situation. Computationally, the size of the problem grows very rapidly as each piece of equipment can perform a number of operations at a machining center or between machining centers. Finally, it can be difficult to estimate some parameters such as cost and time before the system is in operation.
Despite the complexity involved in selecting a materials handling system, it is extremely important that the problem be carefully considered. The magnitude of the cost impact of materials handling justifies such an effort. Past research on the design and selection of a materials handling system has focused primarily on the optimization of materials handling equipment. More recent work has considered the materials handling system as a “whole”, rather than just individual elements of that system. This is of great importance since the objective is to optimize the whole.

At the center of a materials handling system is the material itself. Therefore, it is necessary to start the design cycle by considering the material, its transformation processes and the resultant products. The logistics design needs to be undertaken concurrently with the materials handling system design. Traditionally this has not been the case. Rather, it is only considered once the products have been engineered and bills of materials have been produced. By that time, considerable product lead time has been lost.

As a result of the reasons described above, it is proposed that an integrated approach to manufacturing system design be developed. More specifically, it is proposed that the machine selection and operation allocation problem and the materials handling system selection problem be simultaneously considered to generate a “reasonably optimal” choice of design features.

1.2. **Organization of the Proposal**

The report is divided into eight chapters. This first chapter is an introduction where some background on the current problems and the motivation for this work is
explained. Chapter 2 presents a literature survey of previous research work on the
operation allocation and materials handling system selection problems. The research
objectives of this work are further defined at the end of that chapter. The operation
allocation and materials handling system selection models are then developed and
presented in Chapter 3. Chapter 4 presents a case study that is solved with the use of the
developed models and the solution to the case study is briefly discussed. In Chapter 5,
the sensitivity analysis of various rating factors used in the materials handling system
selection model is presented. The sixth chapter presents the Solver program that was
developed to simplify the solution process to both problems. Chapter 7 discusses the
work presented and offers some ideas for further development of the models. Lastly,
Chapter 8 finalizes with concluding remarks about this research work.
CHAPTER 2. LITERATURE SURVEY

The literature survey is divided into two sections. The first section reviews selected literature for the operation allocation problem. The second section reviews the literature for the materials handling system selection problem.

2.1. Operation Allocation Literature

The operation allocation problem has received considerable attention from researchers in the past. Several approaches have been proposed to solve the problem. For the survey of this work, only the mathematical programming approaches are reviewed. The emphasis is placed on those models that were developed primarily for flexible manufacturing systems.

Stecke (1983) formulated the loading problem in a flexible manufacturing system as a nonlinear 0-1 mixed integer program. Because the best loading procedure is problem dependent, Stecke proposes six different formulations corresponding to six different loading objectives. These are:

1. Balance the assigned machine processing times.
2. Minimize the number of movements from machine to machine.
3. Balance the workload per machine for a system of groups of pooled machines of equal sizes.
4. Unbalance the workload per machine for a system of groups of pooled machines of unequal sizes.
5. Fill the tool magazines as densely as possible.

6. Maximize the sum of operation priorities.

The formulations are solved by various linearization methods.

A branch and bound algorithm was then developed by Berrada and Stecke (1986) to solve the loading problem more expediently than by means of the linearized formulations proposed earlier. However, in this paper, the only loading objective considered was that of balancing the workload on all machines. This is done by minimizing the workload of the bottleneck machine. A sequence of sub-problems is defined and each sub-problem is solved by a branch and bound procedure. The procedure first solves a simple, relaxed assignment problem, next checks for feasibility and then modifies the assignment to correct the violated constraints.

Lashkari et al. (1987) extended the formulation of the operation allocation problem to include the planning aspects of refixturing and limited tool availability. The problem is formulated as a non-linear 0-1 integer programming formulation with two objective functions. The objective functions consist of minimization of transport load and minimization of refixturing activities.

Wilson (1989) proposed an alternative formulation to the one proposed by Lashkari et al. (1987). The planning aspects of refixturing and limited tool availability were also considered. The alternative formulation avoids the non-linearities of the previous approach and instead proposes an integer programming formulation. Only the problem of minimization of transport load is formulated.

Damodaran et al. (1992) presented a mathematical model for the operation allocation problem in multi-machine and multiple cell environments. The mixed integer
linear model considers the trade-off between refixturing and materials handling. The authors argue that this trade-off influences operation allocation. The objective function minimizes the refixturing and materials handling costs as well as the processing costs.

Taboun and Ulger (1992) developed a multi-objective modeling of operation allocation in flexible manufacturing systems. The 0-1 integer programming formulation considers multiple objectives such as the minimization of processing, handling, tool set-up, fixturing/refixturing and penalty costs of under-utilization and over-loading of machining centers.

Ribeiro and Pradin (1993) presented a methodology for cellular manufacturing design that consisted of two stages. In the first stage, the job shop machines are selected and parts are assigned to those machines. In the second stage, the operation allocation to selected machines is realized with the intent of minimizing inter-cell moves. The second phase is developed so that the operator can iteratively repeat it until a certain level of inter cell moves is achieved.

Atmani et al. (1995) developed a mathematical programming approach to joint cell formation and operation allocation in cellular manufacturing. The model introduced was a 0-1 integer program that simultaneously considered the cell formation and operation allocation problems while minimizing the sum of the operation costs, refixturing costs and transportation costs.

Atmani (1995) proposed a flexible planning model for flexible manufacturing systems with setup cost consideration. The 0-1 integer programming model minimizes the joint operation, materials handling and setup costs. This model was applicable to a single cell.
Atmani and Lashkari (1998) presented a linear, 0-1 integer programming model of the machine tool assignment and operation allocation in a flexible manufacturing system. An optimal plan is obtained by minimizing the total costs of operations, materials handling (transportation) and machine set-ups.

Guerrero et al. (1999) presented a mixed integer linear programming model of machine loading and part type selection in flexible manufacturing systems. The model considers the existence of alternative routes for each part type. The objective is to balance the machine workloads.

2.1. **Materials Handling System Selection Literature**

The materials handling system selection problem has also received considerable attention from researchers. Many different approaches have been proposed in past and recent literature.

Chu et al. (1995) as well as Kim and Eom (1997) have proposed expert based system approaches. A closely related approach is that of knowledge based solutions. Some work in this area has been conducted by Matson et al. (1990), and Welgama and Gibson (1995). Another popular approach is simulation. Some researchers that have worked in this area include Nadoli and Rangaswami (1993) as well as Harit and Taylor (1995).

Very few mathematical programming models have been proposed. This is not surprising given the complexity of the problem both from a conceptual and computational point of view. Most of the research in the area of mathematical programming of
materials handling systems has focused on optimizing materials handling equipment and not the whole materials handling system. A good review of such practices is provided by Rembold and Tanchoe (1994).

An example of an algorithm for the selection and assignment of materials handling equipment is that of Hassan et al. (1985). The problem is formulated as an integer programming model with the objective of minimizing the total operating and investment costs of the selected equipment. The model only explores the assignment of equipment for departmental moves (i.e., transportation). Other materials handling operations are not considered.

Recent work has made substantial strides in developing models that deal with the materials handling system as a whole rather than with individual pieces of equipment. Gupta and Dutta (1994) developed a methodology for linking product design with manufacturing logistics design activities. In particular, it was proposed that materials handling needs be considered within the manufacturing systems design through concurrent engineering.

The authors argued that “manufacturing logistics design should be undertaken as soon as the product design stage is started”. Five key product variables are used to describe a product’s characteristics: complexity, precision, diversity, batch size/lot size/run-length and mass or linear dimension. These characteristics are defined in terms of the choice of manufacturing technology associated with the product’s conception, realization, disposition and disposal. Each materials handling system contains within it five subsystems: loading/unloading at the workplace, transportation, handling/rehandling away from the workplace, inspection module and storage/retrieval device. Each of these
subsystems is composed of a number of sub-operations such as orientation change, position change, quantity change, sequence change and timing change. The attributes of the product/process combination, i.e. choice of technology, are related to the materials handling system requirements through a five-point rating scale. The appropriate materials handling system is then selected through a weighted rating method.

This concept was expanded by Atmani and Dutta (1996) into a mathematical model. A 0-1 integer programming model was proposed to maximize the ‘adaptability factor’. The ‘adaptability factor’ maps the product/process design onto the materials handling system design. It was formally defined by Gupta and Dutta (1994) as “the ratio of basic motions/movements that are required by a new product to those available in the current manufacturing logistics system”. The methodology outlined previously by these authors formed the base for the mathematical model.

Other integrated approaches have been proposed. Noble et al. (1998) developed a model that integrates materials handling equipment selection and specification (materials handling interface equipment included) and path/load dependent unit load size. The formulation attempts to minimize the operating and capital cost of materials handling and necessary interface equipment. Although the materials handling operations are not defined, the model appears to only consider transportation between workplaces. Paulo et al. (1999) presented a methodology to consider the operation allocation and materials handling system selection problems jointly. The work represents an important first step towards developing an integrated approach to solving these two manufacturing problems.
2.2. Commentary on Previous Work

The recent work on the development of more integrated models in operations allocation and materials handling system selection is indicative of a trend to model more realistic problems in a manufacturing system. These problems are undoubtedly more complex and difficult to model and solve. This is part of the reason why most past work has dealt with individual problems. However, the realization of the benefits of an integrated approach warrants new research of models that are integrative in nature.

In the past, the operation allocation and materials handling system selection problems have been considered separately. However, it is obvious that the solution to one of these problems is influenced by the solution to the other problem. This has been evidenced by the attempt of previous researchers to include the materials handling costs in their operation allocation models. In addition, a necessary input to most materials handling system selection models is the operations assignments that result from the solution to the operation allocation problem. It is obvious that these two problems are both members of a larger system affected by various interrelationships. This work attempts to bridge these two problems by explicitly considering both of them and their interactions. The work presented here encompasses and enlarges the methodology proposed by Paulo et al. (1999).
CHAPTER 3. THE MATHEMATICAL MODELS

The work builds upon two existing models: the model of the joint cell formation and operation allocation in a cellular manufacturing by Atmani et al. (1995); and the materials handling system selection model by Atmani and Dutta (1996). These models, which form the foundation of this work, were revised and suitably modified for the scope of this work. In the case of the materials handling system selection model, the changes were substantial, resulting in a model that is distinctively different from the previous one. Some of the changes were introduced to allow the integration of the two models.

3.1. Sub-Model 1. Operation Allocation

The model of the joint cell formation and operation allocation in a cellular manufacturing system, proposed by Atmani et al. (1995), was simplified. Specifically, the model was reduced to an operation allocation model. The cell formation component was eliminated in an attempt to reduce the complexity inherent in bridging the two models. As such, the constraint that ensured that a specific machine was only assigned to one specific cell was eliminated.

There were further modifications in the objective function and constraints. One modification was the addition of the machine setup cost to the objective function. This ensures that we now have a more complete description of the total cost. The unit refixturing cost present in the original model was combined with the unit operation cost since these constituted sums over identical sets of indices. As such, the new unit
operation cost includes both the unit manufacturing cost and the unit refixturing cost. The same is true for the constraint that ensures that the total time of the operations assigned to a certain machine does not load the machine beyond its capacity. The unit operation time now includes the unit manufacturing time and the unit refixturing and defixturing times. This modified model is essentially the same as that proposed by Atmani (1995).

The modified 0-1 integer programming model is developed below.

3.1.1. Mathematical Formulation

Assume that there is a set of \( n \) part types labeled with the indices \( i = 1, \ldots, n \), where the part type \( i \) has the known and uniform demand \( d_i \) over the planning period. A part type \( i \) can be processed under the different process plans \( p = 1, \ldots, P(i) \). For a process plan \( p \in P(i) \) of a part type \( i \), the manufacturing operations are represented by the indices \( s = 1, \ldots, S(ip) \). There is a set of \( m \) machines labeled with the indices \( j = 1, \ldots, m \). The set of machines that can perform manufacturing operation \( s \) of a part type \( i \) under process plan \( p \) is given by \( J_{ips} \).

The operation allocation model involves the assignment of operations of each part type to appropriate machines to minimize the total costs of manufacturing operations, machine setups and materials handling. The 0-1 decision variables are denoted by \( X_{sj(ip)} \), where \( X_{sj(ip)} = 1 \) if operation \( s \) of a part type \( i \) is processed under plan \( p \) on machine \( j \) and zero otherwise.

The operation cost is given by \( E_i(X_{sj(ip)}) \):
\[ E_1(X_{n_i}(ip)) = \sum_{i=1}^{n} d_i \sum_{p=1}^{P_{i}(ip)} \sum_{i=1}^{S_{i}(p)} \sum_{j \in J} O_{CS_{i}(ip)} X_{n_i}(ip) X_{n_j}(ip) \]  

where \( O_{CS_{i}(ip)} \) is the known cost of operation \( s \) of a unit of part type \( i \) on machine \( j \) under process plan \( p \). This includes both the manufacturing cost and the refixturing cost.

The machine setup cost is given by \( E_2(M) \):

\[ E_2(M) = \sum_{j=1}^{r} SC_j, M \]

where \( SC_j \) is the known setup cost for machine \( j \) and the auxiliary variable \( M \) takes the value of one if machine \( j \) is selected and zero otherwise. Only one setup cost per planning period is assumed. Since these machines are operated under a flexible manufacturing system, it is assumed that the only significant setup cost is the initial one where the machines must be programmed and equipped with the required tools.

Finally, the materials handling cost is given by \( \bar{E}_3(X_{SF}(ip)) \):

\[ \bar{E}_3(X_{SF}(ip)) = \sum_{i=1}^{n} d_i \sum_{p=1}^{P_{i}(ip)} \sum_{i=1}^{S_{i}(p)} \sum_{j \in J} T_{ij} X_{n_i}(ip) X_{(s+1)j}(ip) \]

where \( T_{ij} \) is the cost of moving a unit of part type \( i \) from machine \( j \) to machine \( j \) for the next operation. Since \( \bar{E}_3(X_{SF}(ip)) \) is a nonlinear function, the linearization technique given in Taha (1992) is applied. This prescribes replacing \( \bar{E}_3(X_{SF}(ip)) \) with:

\[ E_3(L_{ij}(ip)) = \sum_{i=1}^{n} d_i \sum_{p=1}^{P_{i}(ip)} \sum_{i=1}^{S_{i}(p)} \sum_{j \in J} T_{ij} L_{ij}(ip) \]

where \( L_{ij}(ip) \) is a new 0-1 variable satisfying the following two sets of constraints:

\[ X_{n_i}(ip) + X_{(s+1)j}(ip) - 2 L_{ij}(ip) \geq 0 \quad \forall i, p, s \in \{1, 2, \ldots, S(ip) - 1\}, j \in J_{qa}, \tilde{j} \in J_{qa(i+1)} \]  

\[ X_{n_i}(ip) + X_{(s+1)j}(ip) - L_{ij}(ip) \leq 1 \quad \forall i, p, s \in \{1, 2, \ldots, S(ip) - 1\}, j \in J_{qa}, \tilde{j} \in J_{qa(i+1)} \]
Note how the above set of constraints ensure that $L_{ij}(ip)$ takes the value of one if and only if a unit of part type $i$ is moved to machine $j$ for operation $(s+1)$ under plan $p$, after performing operation $s$ on machine $j$ under plan $p$.

The objective of the model is therefore to determine the values of $X_{ij}(ip)$ and $L_{ij}(ip)$ that will minimize the total operating, machine set-up and materials handling costs. This is mathematically expressed as:

$$\min \ E_1(X_n(ip)) + E_2(M) + E_3(L_{i}(ip))$$  \hspace{1cm} (7)

The constraints are developed next. The first constraint set ensures that each part type is processed under a single process plan, and it is given by:

$$\sum_{p=1}^{P(i)} Z(ip) = 1 \hspace{1cm} \forall i$$  \hspace{1cm} (8)

where $Z(ip) = 1$ if part type $i$ is processed under process plan $p$ and zero otherwise.

The next set of constraints ensures that once a process plan is selected for a part type, each operation in that plan is processed on only one of the available machines. It is given by:

$$\sum_{p=1}^{P(i)} X_{ij}(ip) = Z(ip) \hspace{1cm} \forall i, p, s$$  \hspace{1cm} (9)

The third set of constraints ensures that if machine $j$ is selected then at least one operation must be assigned to it. It is given by:

$$\sum_{i=1}^{n} \sum_{p=1}^{P(i)} \sum_{s=1}^{S(ip)} X_{ij}(ip) \geq M_j \hspace{1cm} \forall j$$  \hspace{1cm} (10)
The next set of constraints ensures that the total time required by the operations allocated to a machine \(j\), once it is selected, does not exceed the machine's known capacity. It is given by:

\[
\sum_{i=1}^{n} d_i \sum_{p=1}^{P(i)} \sum_{s=1}^{S(ip)} t_{sj}(ip) X_{sj}(ip) \leq b_j M_j \quad \forall j
\]  

where \(t_{sj}(ip)\) is the time to perform operation \(s\) of part type \(i\) on machine \(j\) under process plan \(p\). The two sets of constraints, (10) and (11), ensure consistency between the allocation of operations of the part types to machines and the selection of machines.

Assembling the above, we get the following complete statement of our 0-1 mathematical programming model of the operation allocation model, which is designated as P(OA).

**P(OA): Minimize**

\[
\sum_{i=1}^{n} d_i \sum_{p=1}^{P(i)} \sum_{s=1}^{S(ip)} O_{sj}(ip) X_{sj}(ip) + \sum_{j=1}^{m} S_{ij} M_j
\]

\[
+ \sum_{i=1}^{n} d_i \sum_{p=1}^{P(i)} \sum_{s=1}^{S(ip)} \sum_{j \in F_{p}} \sum_{i \in F_{q+1}} T_{ij} L_{sji}(ip)
\]  

Subject to

\[
\sum_{p=1}^{P(i)} Z(ip) = 1 \quad \forall i
\]  

\[
\sum_{j \in F_{p}} X_{sj}(ip) = Z(ip) \quad \forall i, p, s
\]  

\[
\sum_{i=1}^{n} \sum_{p=1}^{P(i)} \sum_{s=1}^{S(ip)} X_{sj}(ip) \geq M_j \quad \forall j
\]  

\[
\sum_{i=1}^{n} d_i \sum_{p=1}^{P(i)} \sum_{s=1}^{S(ip)} t_{sj}(ip) X_{sj}(ip) \leq b_j M_j \quad \forall j
\]
\[
X_{sq}(ip) + X_{(s+1)j}(ip) - 2L_{sij}(ip) \geq 0 \quad \forall i, p, s \in \{1, 2, ..., S(ip) - 1\}, j \in J_{ip}, \hat{j} \in J_{ip(s-1)} \tag{17}
\]

\[
X_{sq}(ip) + X_{(s+1)j}(ip) - L_{sij}(ip) \leq 1 \quad \forall i, p, s \in \{1, 2, ..., S(ip) - 1\}, j \in J_{ip}, \hat{j} \in J_{ip(s-1)} \tag{18}
\]

\[
[L_{sij}(ip), X_{sq}(ip), Z(ip), M_{j}] \in \{0, 1\} \quad \forall i, p, s, j \in J_{ip}, \hat{j} \in J_{ip(s-1)} \tag{19}
\]

The assignments determined by the model (i.e., the \(X_{sq}(ip)\)) are then summarized in the matrix \(A_{sij}\), where an element of \(A_{sij}\) is equal to one if operation \(s\) of part type \(i\) is to be performed at machine \(j\), and zero otherwise. This matrix serves as an input to the materials handling system selection model.

### 3.2. Sub-model 2. Materials Handling System Selection

The current implementation of the materials handling system selection model is significantly different from the original model by Atmani and Dutta (1996). The original model was used as the basis for the current model. However, a number of significant modifications were introduced. The model developed in this section shares much of the conceptual framework of the previous model, but is implemented much differently. The changes derive from an evolution in the model development as well as from the goal to integrate the operation allocation model with the materials handling system selection model.

The notation was modified to accommodate the interfacing of this model with the operation allocation model. It is necessary that common elements such as part types and machines be represented by the same indices to facilitate communication between the two models.
As it was the case with the operation allocation model, the cell component of the materials handling system selection formulation was removed. Therefore, the constraint limiting the number of pieces of materials handling equipment assigned to each cell was also removed.

The new implementation of the 0-1 integer programming model is developed below.

3.2.1. Mathematical Formulation

Assume the same set of \( n \) part types as defined in the operation allocation model, labeled with the indices \( i = 1, \ldots, n \), where the part type \( i \) has the known and uniform demand \( d_i \) over the planning period. A part type \( i \) can be described in terms of five key product variables that define the choices of manufacturing technology associated with its conception, realization, disposition and disposal. These are labeled by indices \( t = 1, \ldots, T(i) \). In our case \( T(i) = 5 \) and the key product variables are complexity, precision, lot or batch size, diversity and mass or linear dimension. Most of these are easily measured parameters to manufacturing engineers and managers. For example, precision can be readily measured in terms of tolerances. However, complexity can be difficult to measure. For the purpose of this thesis, it is assumed that the information embodied in the shapes of part types is the measure of complexity. More precisely, complexity is measured based on information theory. It measures the minimum number of bits of information needed to describe a part. In most situations this includes geometrical and dimensional information. Details of its quantification are presented by Ayres (1988).
There is a set of $m$ machines labeled with the indices $j = 1, \ldots, m$. At these machines, the required manufacturing operations for a part type $i$ are represented by the indices $s = 1, \ldots, S(i)$. The process plan index $p$ is dropped from this model since a plan has already been selected by the previous model.

The major materials handling operations, $h = 1, \ldots, H$, where in our case $H = 5$, are load/unload at the machine, transportation between manufacturing processes or machines, handling/rehandling away from the workplace, inspection and storage and retrieval. Each such operation is associated with a number of sub-operations labeled as $\hat{h} = 1, \ldots, \hat{H}$, where in our case $\hat{H} = 6$. These are orientation change, position change, quantity change, sequence change, timing change and "no change". Simply explained, an orientation change involves rotational change or movement while a position change involves linear change or movement. Sequence change occurs when the order of part types is modified. Timing change involves a change in cycle time between processing of part types. A number of materials handling equipment are available to perform these materials handling operation/sub-operation combinations. These are labeled as $e = 1, \ldots, \hat{E}$.

The 0-1 decision variables are denoted by $Y_{hhejs}(i)$, where $Y_{hhejs}(i) = 1$ if materials handling operation $h$, which is associated with sub-operation $\hat{h}$, requires materials handling equipment $e$ at machine $j$ where manufacturing operation $s$ of part type $i$ is performed.

The objective is to generate the optimal materials handling system selection for a given mix of part types based on the part/process combination and the choices of materials handling equipment available. Thus, we would like to maximize:
\[
\sum_{e=1}^{E} \sum_{h=1}^{H} \sum_{k=1}^{K} W_{hk} \sum_{i=1}^{n} C_{ei} \sum_{s=1}^{S} \sum_{j=1}^{J} A_{sij} \alpha_{kjs} \gamma_{kxe} (i)
\]

where \( C_{ei} = 1 - \frac{\sum_{t=1}^{T} |W_{et} - \hat{W}_{it}|}{4T} \)

where \( W_{hk} \) is a measure of the ability of a materials handling piece of equipment to handle a certain operation/sub-operation combination while \( W_{et} \) and \( \hat{W}_{it} \) relate the key product variables to the materials handling equipment and the part type, respectively.

The parameter \( C_{ei} \) is a measure of the compatibility of a piece of equipment and a part type. The fraction part of the expression evaluates to a value between 0 and 1. The maximum allowed value for the difference between each individual pair of \( W_{et} \) and \( \hat{W}_{it} \) is 4 units. Since the absolute value of this difference is summed over \( t = 1, \ldots, T \), where \( T = 5 \), the maximum value that the fraction numerator can exhibit is 20. The denominator always evaluates to 20. If the values of \( W_{et} \) and \( \hat{W}_{it} \) are as far apart as possible, the value of the fraction will be 20/20 = 1. However, when this is the case, we want to eliminate the materials handling equipment/part type combination from consideration. Since we are maximizing the objective function, what we really want is for the value of \( C_{ei} \) to be as small as possible for this situation. Therefore, the fraction is subtracted from 1 to obtain the value of \( C_{ei} \). As such, this parameter evaluates to a number between 0 and 1, where 0 indicates incompatibility and 1 indicates complete compatibility. Incompatibility would occur for a situation where the values of \( W_{et} \) and \( \hat{W}_{it} \) are as far apart as possible, which signals that the materials handling equipment is unsuitable to handle that part type. Complete compatibility would occur when the sum of the differences between \( W_{et} \) and \( \hat{W}_{it} \) evaluate to 0. In this case, \( C_{ei} \) evaluates to a value of 1, which affords the greatest maximization of the objective function. In essence, since the objective function is to be
maximized, the model forces $C_{ei}$ to be as close as possible to a value of 1. In turn, this forces the $W_{et}$ and $\hat{W}_{it}$ ratings to be as close as possible in values. This insures that the model attempts to match as well as possible the materials handling equipment and the part types it will service. For most situations, the parameter evaluates to a numerical value between 0 and 1, indicating some degree of compatibility.

Integer scales are used to assign values to the $W$ parameters. The rating scales range from 0 to 5 for $W_{hhe}$ and $W_{et}$, and 1 to 5 for $\hat{W}_{it}$. The interpretation of the values is given in Tables 1, 2 and 3.

<table>
<thead>
<tr>
<th>Rating</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Excellent at performing the operation/sub-operation combination</td>
</tr>
<tr>
<td>4</td>
<td>Very capable of performing the operation/sub-operation combination</td>
</tr>
<tr>
<td>3</td>
<td>Satisfactorily capable of performing the operation/sub-operation combination</td>
</tr>
<tr>
<td>2</td>
<td>Poor, but capable of performing the operation/sub-operation combination</td>
</tr>
<tr>
<td>1</td>
<td>Very poor, but minimally capable of performing the operation/sub-operation combination</td>
</tr>
<tr>
<td>0</td>
<td>Incapable of performing the operation/sub-operation combination</td>
</tr>
</tbody>
</table>

Table 1. Rating Scale for $W_{hhe}$

<table>
<thead>
<tr>
<th>Rating</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Piece of equipment best suited to handle parts with a very high rating of product variable $t$</td>
</tr>
<tr>
<td>4</td>
<td>Piece of equipment best suited to handle parts with a high rating of product variable $t$</td>
</tr>
<tr>
<td>3</td>
<td>Piece of equipment best suited to handle parts with a moderate rating of product variable $t$</td>
</tr>
<tr>
<td>2</td>
<td>Piece of equipment best suited to handle parts with a low rating of product variable $t$</td>
</tr>
<tr>
<td>1</td>
<td>Piece of equipment best suited to handle parts with a very low rating of product variable $t$</td>
</tr>
<tr>
<td>0</td>
<td>Do not allow this piece of equipment to handle parts with product variable $t$</td>
</tr>
</tbody>
</table>

Table 2. Rating Scale for $W_{et}$

<table>
<thead>
<tr>
<th>Rating</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Part type exhibits a very high level of the key product variable $t$</td>
</tr>
<tr>
<td>4</td>
<td>Part type exhibits a high level of the key product variable $t$</td>
</tr>
<tr>
<td>3</td>
<td>Part type exhibits a moderate level of the key product variable $t$</td>
</tr>
<tr>
<td>2</td>
<td>Part type exhibits a low level of the key product variable $t$</td>
</tr>
<tr>
<td>1</td>
<td>Part type exhibits a very low level of the key product variable $t$</td>
</tr>
</tbody>
</table>

Table 3. Rating Scale for $\hat{W}_{it}$
The three rating factors \((W_{hic}, W_{et} \text{ and } \hat{W}_{ir})\) are largely subjective and a formalized approach to their quantification has not been developed. Details of the quantification of the key product variables are given in a paper by Ayres (1988).

The matrix \(A_{sij}\) is derived from the output of the operation allocation model where any element from that matrix equals one if a manufacturing operation \(s\) for part type \(i\) is performed at machine \(j\) and zero otherwise. This parameter is the link between the two models. It provides a necessary input to the materials handling system selection model by describing the manufacturing operation assignments for each part type.

The parameter \(a_{hshij}\) equals one if materials handling operation \(h\), which is followed by sub-operation \(\hat{h}\), is required at machine \(j\) where manufacturing operation \(s\) of part type \(i\) is performed and zero otherwise. This parameter is a new addition to the model, which describes the required materials handling operations/sub-operations correspondent to the manufacturing operations. This was a necessary addition since the goal of the model changed. The previous model was a stand-alone model that chose the best general materials handling system. It was unaware of part type routings and manufacturing operations. The current model is no longer stand-alone. One of its inputs is the matrix summarizing the operations allocations. As such, it is necessary to supply the model with information concerning the materials handling operations that must be performed at a machine where a manufacturing operation has been assigned. This information is inputted by the user after the machine selections and operation assignments are known.

The first set of constraints ensure that only one type of materials handling equipment is chosen to perform a materials handling operation/sub-operation
combination required for manufacturing operation $s$ of part type $i$ at machine $j$. It is expressed as:

$$
\sum_{e=1}^{E} Y_{hhejs}(i) = A_{sij} \alpha_{hheij} \quad \forall s, i, j, h, \hat{h}
$$

(21)

This constraint also ensures that a piece of materials handling equipment is only assigned when it is required. In other words, the decision variable $Y_{hhejs}(i)$ is only set equal to one when a materials handling operation/sub-operation is required.

The next set of constraints ensures that a piece of equipment $e$ is only chosen after another piece of equipment $\hat{e}$ has been assigned. This set of constraints is provided to allow precedence relationships that may exist in the assignment of materials handling equipment. It is given as:

$$
D_e \leq D_{\hat{e}} \quad \forall e, \hat{e}, e \neq \hat{e}
$$

(22)

where $D_e = 1$ if a piece of equipment $e$ has been chosen and zero otherwise. Similarly for $D_{\hat{e}}$.

The third set of constraints ensures that if a piece of materials handling equipment is chosen, then at least one materials handling operation/sub-operation combination must be performed by that equipment. It is mathematically expressed as:

$$
\sum_{s=1}^{n} \sum_{h=1}^{H} \sum_{k=1}^{K} \sum_{j=1}^{J} A_{sij} \alpha_{hheij} Y_{hhejs}(i) \geq D_e \quad \forall e
$$

(23)

The next set of constraints guarantees that the total time for all jobs assigned to a piece of materials handling equipment does not exceed the time available on that piece of equipment. It is given by:
\[
\sum_{i=1}^{n} d_i \sum_{h=1}^{H} \sum_{h' = 1}^{H'} \sum_{s=1}^{S(i)} t_{h'h'c} A_{sij} \alpha_{kh'sj} Y_{kh'sj}(i) \leq T_e D_e \quad \forall e
\]

where \( t_{h'h'c} \) is the time required by materials handling equipment \( e \) to perform the materials handling operation/sub-operation combination \( h'h \) and \( T_e \) is the time available on materials handling equipment \( e \).

Assembling the above, we get the following complete statement of our 0-1 mathematical programming model of the materials handling system selection model, which is designated as \( P(MH) \).

**P(MH): Maximize**

\[
\sum_{e=1}^{E} \sum_{h=1}^{H} \sum_{h' = 1}^{H'} \sum_{s=1}^{S(i)} C_{ei} \sum_{i=1}^{n} A_{sij} \alpha_{kh'sj} Y_{kh'sj}(i)
\]

where \( C_{ei} = 1 - \frac{\sum_{i=1}^{n} |W_{et} - \hat{W}_{et}|}{4T} \)

**Subject to**

\[
\sum_{e=1}^{E} Y_{kh'sj}(i) = A_{sij} \alpha_{kh'sj} \quad \forall s, i, j, h, \hat{h}
\]

\[
D_e \leq D_{\hat{e}} \quad \forall e, \hat{e}, \ e \neq \hat{e}
\]

\[
\sum_{i=1}^{n} \sum_{h=1}^{H} \sum_{h' = 1}^{H'} \sum_{s=1}^{S(i)} A_{sij} \alpha_{kh'sj} Y_{kh'sj}(i) \geq D_e \quad \forall e
\]

\[
\sum_{i=1}^{n} d_i \sum_{h=1}^{H} \sum_{h' = 1}^{H'} \sum_{s=1}^{S(i)} t_{h'h'c} A_{sij} \alpha_{kh'sj} Y_{kh'sj}(i) \leq T_e D_e \quad \forall e
\]

\[
[Y_{kh'sj}(i), D_e, D_{\hat{e}}] \in \{0,1\} \quad \forall i, s, h, \hat{h}, e, j \in J_{is}
\]
CHAPTER 4. CASE STUDY EXAMPLE

An example is presented to demonstrate the viability of this work and of the models developed. This randomly generated example was solved using Lingo (Lindo Systems Inc., 1999). The Lingo program files that were produced to solve this case study can be found in Appendix I.

4.1. The Operation Allocation Problem

Assume that we have \( i = 1, \ldots, 14 \) part types to be manufactured with demands as listed in Table 4. The part types have \( P(1) = 2, P(2) = 3, P(3) = 2, P(4) = 2, P(5) = 2, P(6) = 1, P(7) = 2, P(8) = 2, P(9) = 1, P(10) = 3, P(11) = 2, P(12) = 2, P(13) = 1, P(14) = 2 \) different process plans. Each process plan can include up to four manufacturing operations. There are \( j = 1, \ldots, 10 \) machines with known capacity \( b_j = 57,600 \) and setup costs as listed in Table 4. Each of these machines is capable of performing operations required by the process plans. For example: under process plan \( p = 1 \), part type \( i = 1 \) has \( S(11) = 3 \) operations with indices \( s \in \{1, 2, 3\} \). Operation \( s = 1 \) of process plan \( p = 1 \) for part type 1 can be completed on any of the machines \( j \in J_{111} = \{1, 3, 4, 7, 10\} \) while operation \( s = 2 \) can be completed at any of the machines \( j \in J_{112} = \{1, 5, 7, 8\} \) and operation \( s = 3 \) can be completed at any of the machines \( j \in J_{113} = \{3, 4, 10\} \). The information for all allowable combinations \( (ip) \) are summarized in Table 4, which contains the values of the operation times \( t_{sj}(ip) \) and costs \( OC_{sj}(ip) \).
<table>
<thead>
<tr>
<th>Machine, j</th>
<th>Part Types, i</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Process Plan, p</td>
</tr>
<tr>
<td></td>
<td>Operation, s</td>
</tr>
<tr>
<td>1</td>
<td>1  2  3</td>
</tr>
<tr>
<td>2</td>
<td>6  12</td>
</tr>
<tr>
<td>3</td>
<td>$9  $18</td>
</tr>
<tr>
<td>4</td>
<td>11  12  9</td>
</tr>
<tr>
<td>5</td>
<td>8  9  $7</td>
</tr>
<tr>
<td>6</td>
<td>5  8  $7</td>
</tr>
<tr>
<td>7</td>
<td>11  $10  8</td>
</tr>
<tr>
<td>8</td>
<td>$11  $10  8</td>
</tr>
<tr>
<td>9</td>
<td>3  2  8</td>
</tr>
<tr>
<td>10</td>
<td>$3  $10  8</td>
</tr>
</tbody>
</table>

Table 4. Manufacturing operation times $t_{ij}(ip)$ and costs $OC_{ij}(ip)$, part type demands $d_i$ and machine setup costs $SC_j$. 

27
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Operation, s</td>
<td>Operation, s</td>
<td>Operation, s</td>
<td>Operation, s</td>
<td>Operation, s</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Machine, j</td>
<td>Operation, s</td>
<td>Operation, s</td>
<td>Operation, s</td>
<td>Operation, s</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>9</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>13</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>23</td>
<td>26</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>10</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>14</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>8</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>10</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>11</td>
<td>14</td>
<td>17</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>13</td>
<td>16</td>
<td>19</td>
</tr>
</tbody>
</table>

Table 4. (cont'd 1)
<table>
<thead>
<tr>
<th>Machine, j</th>
<th>Process Plan, p</th>
<th>Part Types, i</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Operation, s</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>$6</td>
</tr>
<tr>
<td></td>
<td>$15</td>
<td>$3</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>$6</td>
</tr>
<tr>
<td></td>
<td>$14</td>
<td>$3</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>$4</td>
</tr>
<tr>
<td></td>
<td>$5</td>
<td>$7</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>$4</td>
</tr>
<tr>
<td></td>
<td>$9</td>
<td>$7</td>
</tr>
<tr>
<td></td>
<td>$12</td>
<td>$4</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>$2</td>
</tr>
<tr>
<td></td>
<td>$2</td>
<td>$8</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>$13</td>
</tr>
<tr>
<td></td>
<td>$11</td>
<td>$9</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>$6</td>
</tr>
<tr>
<td></td>
<td>$3</td>
<td>$7</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>$4</td>
</tr>
<tr>
<td></td>
<td>$10</td>
<td>$7</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>$11</td>
</tr>
<tr>
<td></td>
<td>$14</td>
<td>$6</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>$9</td>
</tr>
<tr>
<td></td>
<td>$11</td>
<td>$2</td>
</tr>
<tr>
<td>Demand, di</td>
<td>90</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 4. (cont'd 2)
<table>
<thead>
<tr>
<th>Machine, j</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>12</td>
<td>9</td>
<td></td>
<td>4</td>
<td>10</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$6</td>
<td>$18</td>
<td>$14</td>
<td></td>
<td>$6</td>
<td>$15</td>
<td></td>
<td></td>
<td>$5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>11</td>
<td></td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$14</td>
<td></td>
<td>$9</td>
<td>$6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>9</td>
<td></td>
<td>8</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$3</td>
<td>$7</td>
<td></td>
<td>$6</td>
<td>$5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>13</td>
<td></td>
<td>6</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$22</td>
<td>$26</td>
<td></td>
<td>$12</td>
<td>$8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td></td>
<td>9</td>
<td>13</td>
<td></td>
<td>8</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$3</td>
<td></td>
<td>$9</td>
<td>$13</td>
<td>$8</td>
<td>$3</td>
<td>$5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>9</td>
<td></td>
<td>4</td>
<td>3</td>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$14</td>
<td>$13</td>
<td></td>
<td>$6</td>
<td>$4</td>
<td></td>
<td>$13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>10</td>
<td></td>
<td>4</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$7</td>
<td>$6</td>
<td></td>
<td>$2</td>
<td>$4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>7</td>
<td>11</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$13</td>
<td>$11</td>
<td>$18</td>
<td>$14</td>
<td>$16</td>
<td>$18</td>
<td>$8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>6</td>
<td>12</td>
<td>2</td>
<td>13</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$13</td>
<td>$7</td>
<td>$14</td>
<td>$2</td>
<td>$16</td>
<td>$4</td>
<td>$5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. (cont'd 3)
<table>
<thead>
<tr>
<th>Machine, j</th>
<th>Part Types, i</th>
<th>Process Plan, p</th>
<th>Machine Setup Cost, SCj</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>i</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demand, di</td>
<td>40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. (cont'd 4)
To simplify this example, we assume that the cost of moving a part from one machine to the next is the same for any part type. These materials handling costs are summarized in Table 5.

<table>
<thead>
<tr>
<th>Machine, j</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>14</td>
<td>11</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>10</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>0</td>
<td>12</td>
<td>13</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>13</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>12</td>
<td>0</td>
<td>7</td>
<td>10</td>
<td>4</td>
<td>5</td>
<td>11</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>13</td>
<td>7</td>
<td>0</td>
<td>13</td>
<td>13</td>
<td>6</td>
<td>14</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>13</td>
<td>0</td>
<td>5</td>
<td>15</td>
<td>6</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>7</td>
<td>4</td>
<td>13</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>11</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>15</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>13</td>
<td>11</td>
<td>14</td>
<td>6</td>
<td>11</td>
<td>8</td>
<td>0</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td>10</td>
<td>9</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>j</td>
<td>10</td>
<td>7</td>
<td>4</td>
<td>9</td>
<td>7</td>
<td>15</td>
<td>4</td>
<td>6</td>
<td>15</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 5. Cost of moving a unit of part type $i$ from machine $j$ to machine $j$, $T_{ij}$ ($i = 1, \ldots, 14$).

The problem was solved with the aid of Lingo. The results obtained are presented in Table 6. The results indicate that machines 1 and 3-8 were selected to process the 14 part types under one of their possible process plans. For example, part type $i = 1$ is processed under plan $p = 2$ where operation $s = 1$ is performed at machine $j = 3$ and operation $s = 2$ is performed at machine $j = 1$. Similar interpretations can be made for all other part types. These results are stored in $A_{sij}$ to serve as input to the materials handling system selection model. The total cost is 7,752, which includes processing, machine setup and transportation costs.

The machine utilization percentages are quite small, indicating that the machine availability exceeds by far the time requirements of the processing operations. In a more realistic situation, the time available on the machines will not greatly exceed the time requirements of the operations.
The same example was re-solved with machine capacities that were significantly reduced. For this case, the machine availability of machines $j = 3$ and $j = 7$ were set at 480 time units, while the time available in each of the other machines was restricted to 980 time units. All the other case study data remained unchanged. Once the problem was solved, a new solution was obtained and it is presented in Table 7.

The solution to this modified case shows some interesting differences from the solution to the previous example where there was ample machine time available. The first difference that one might notice is that for some part types a different process plan was chosen. No longer is it possible for the process plan with the absolute lowest cost to be always chosen. Due to constraints in machine availability, the model is forced at times to choose process plans yielding slightly more expensive processing costs.

Another major difference and one that is expected, is that the machine utilization percentages increased. Further, in this example, all available machines were selected. Again, this was necessary to satisfy the machining time requirements. Please note that for a few of the machines, the utilization is extremely high. It is assumed that machine allowances were factored into the machine availability data used in the example. This means that if one expects a machine to have an uptime of 90% then the 980 time units made available to the model actual represent 1089 time units of real time.

A price must be paid for the time constraints on the machines. As it was discussed above, the limited machine availability forces the model to choose more costly process plans. For this example, the new value of the objective function was 11,789. This represents an increased cost of 4,037.
<table>
<thead>
<tr>
<th>Part, i (Process Plan, p)</th>
<th>Machine Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(2) 2(1) 3(1) 4(1) 5(3) 6(1) 7(1) 8(2) 9(1) 10(1) 11(1) 12(2) 13(1) 14(1)</td>
<td>1.2%</td>
</tr>
<tr>
<td>M 2</td>
<td>-</td>
</tr>
<tr>
<td>a 3 1 1</td>
<td>-</td>
</tr>
<tr>
<td>c 4 1</td>
<td>-</td>
</tr>
<tr>
<td>h 5 1</td>
<td>-</td>
</tr>
<tr>
<td>i 6 1 2</td>
<td>-</td>
</tr>
<tr>
<td>n 7 2 2 1 2,3,4 1 2</td>
<td>-</td>
</tr>
<tr>
<td>e 8 2</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
</tr>
<tr>
<td>j 10</td>
<td>-</td>
</tr>
</tbody>
</table>

*Operation 2 of part type 1 under process plan 2 is performed at machine 1.*

Table 6. Machine selection and operation assignments for the case study.

<table>
<thead>
<tr>
<th>Part, i (Process Plan, p)</th>
<th>Machine Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(2) 2(1) 3(1) 4(1) 5(2) 6(1) 7(2) 8(1) 9(1) 10(1) 11(1) 12(2) 13(1) 14(1)</td>
<td>100.0%</td>
</tr>
<tr>
<td>M 2 1</td>
<td>1 87.5%</td>
</tr>
<tr>
<td>a 3 2</td>
<td>2 93.8%</td>
</tr>
<tr>
<td>c 4 1 3</td>
<td>1 21.9%</td>
</tr>
<tr>
<td>h 5 2</td>
<td>2 89.6%</td>
</tr>
<tr>
<td>i 6 1 1</td>
<td>1 85.9%</td>
</tr>
<tr>
<td>n 7 2 2</td>
<td>1 100.0%</td>
</tr>
<tr>
<td>e 8 2</td>
<td>2 93.8%</td>
</tr>
<tr>
<td>9 1,2 1</td>
<td>1 99.0%</td>
</tr>
<tr>
<td>j 10</td>
<td>1 99.5%</td>
</tr>
</tbody>
</table>

*Operation 2 of part type 1 under process plan 2 is performed at machine 1.*

Table 7. Machine selection and operation assignments for case with reduced machine availability.
4.2. The Materials Handling System Selection Problem

As discussed previously, it is assumed that each product has \( i = 1, \ldots, 5 \) choices of manufacturing technology. These five choices of manufacturing technology describe the part types in terms of their key product variables as seen in Table 8.

<table>
<thead>
<tr>
<th>Part, ( i )</th>
<th>Complexity</th>
<th>Precision</th>
<th>Diversity</th>
<th>Batch size</th>
<th>Mass/linear Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 8. \( \hat{W}_i \) table for the case study example.

From Table 8 we can observe that, for example, part type \( i = 2 \) has a high degree of complexity and precision. This means that this part is composed of a large number of geometrical and dimensional features and it must be manufactured to close tolerances. The very low score for diversity indicates that this part type belongs to a part family composed of a small number of products. We also know that this part type is manufactured in medium-size batches and it is large in dimension/mass. A similar description of each part type can be inferred from Table 8.

The fundamental materials handling operations, \( h = 1, \ldots, 5 \), are load/unload, handling/rehandling, transportation, inspection and storage/retrieval. Each of these operations is associated with one or more sub-operations such as orientation, positioning,
quantity, sequence, timing or none. These are labeled with the indices \( \hat{h} = 1, \ldots, 6 \). For this example it is assumed that we have \( e = 1, \ldots, 9 \) different types of materials handling equipment. Each of these is available for 57600 time units. The materials handling equipment is rated against the choices of manufacturing technology as seen in Table 9.

<table>
<thead>
<tr>
<th>Equipment, ( e )</th>
<th>Complexity</th>
<th>Precision</th>
<th>Diversity</th>
<th>Batch size</th>
<th>Mass/linear Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light-load robot</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Heavy-load robot</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Human</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Powered hand truck</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Forklift truck</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Roller bed belt conveyor</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Light-duty belt conveyor</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>AGV</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>AS/RS</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 9. \( W_{\text{het}} \) table for case study example.

Table 9 indicates, for example, that the available robots are best suited to handle part types with a high degree of complexity and precision, a low degree of diversity and large batch sizes. The light-load robot is best suited to handle part types of very small mass/linear dimension while the heavy-load robot is best suited to handle large and heavier part types. In other words, the ideal part type for these robots to handle is one that is described by a large number of geometrical and dimensional features, requires close tolerances, belongs to a part family with a low number of members and is produced in large batches.

The materials handling equipment is also rated on its ability to perform the various materials handling operation/sub-operation combinations (Table 10). Again, using the robots as an example, we can see that our particular robots are capable of performing the various operations required for load/unload and handle/rehandle. The
degree to which these are able to perform the various operation/sub-operation combinations varies. For example, the robots excel at load/unload operations with the exception of those requiring a quantity change. This is understandable since robots are often fitted with grippers and unable to accommodate a varying number of part types from cycle to cycle. The same is true for handle/rehandle operations.

The unit times required by the various types of materials handling equipment to perform the materials handling operation/sub-operation combinations are given in Table 11.

The materials handling requirements are derived from the output of the operation allocation problem. Table 12 summarizes the materials handling requirements for this case study example.

The problem was formulated in Lingo and then solved. The results obtained from the Lingo solution of the materials handling system selection are summarized in Tables 13 and 14.

The results indicate that the model behaves in an expected manner, choosing sound materials handling assignments for the part types. For example, if we look carefully at the materials handling assignments for part type 2, we can reason why the chosen assignments are the correct ones. The load/unload operations were assigned to the heavy-load robot. This is not surprising since part type 2 is a heavy and large part, of very high complexity and precision. The AGV was chosen for the transportation operations. Again, this is not surprising given the characteristics of part type 2.
<table>
<thead>
<tr>
<th>Equipment, e</th>
<th>(un)Load</th>
<th>(re)Handling</th>
<th>Transportation</th>
<th>Inspection</th>
<th>Storage/Retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light-load robot</td>
<td>O 5 2 5 5 5</td>
<td>O 5 2 5 5 5</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
</tr>
<tr>
<td>Heavy-load robot</td>
<td>O 5 5 2 5 5</td>
<td>O 5 5 2 5 5</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
</tr>
<tr>
<td>Human</td>
<td>O 5 5 5 4 5</td>
<td>O 5 5 5 4 5</td>
<td>O 0 0 0 0 0</td>
<td>O 5 5 5 4 5</td>
<td>O 0 0 0 0 0</td>
</tr>
<tr>
<td>Powered hand truck</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
<td>O 2 5 4 4 3 5</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
</tr>
<tr>
<td>Forklift truck</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
<td>O 2 5 4 4 3 5</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
</tr>
<tr>
<td>Roller bed belt conveyor</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
<td>O 1 3 4 4 4 5</td>
<td>O 1 3 4 4 4 5</td>
<td>O 0 0 0 0 0</td>
</tr>
<tr>
<td>Light-duty belt conveyor</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
<td>O 1 3 4 4 4 5</td>
<td>O 1 3 4 4 4 5</td>
<td>O 0 0 0 0 0</td>
</tr>
<tr>
<td>AGV</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
<td>O 2 3 2 5 5 5</td>
<td>O 2 3 2 5 5 5</td>
<td>O 0 0 0 0 0</td>
</tr>
<tr>
<td>AS/RS</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
<td>O 0 0 0 0 0</td>
<td>O 5 5 5 5 5</td>
</tr>
</tbody>
</table>

Table 10. $W_{i,kc}$ table for the case study.

<table>
<thead>
<tr>
<th>Equipment, e</th>
<th>(un)Load</th>
<th>(re)Handling</th>
<th>Transportation</th>
<th>Inspection</th>
<th>Storage/Retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light-load robot</td>
<td>0.3 0.3 2.0 0.3 0.3 0.2</td>
<td>0.3 0.3 2.0 0.3 0.3 0.2</td>
<td>3.0 3.0 3.0 3.0 3.0 3.0</td>
<td>0.3 0.3 3.0 0.3 0.3 0.2</td>
<td>4.0 4.0 4.0 4.0 4.0 4.0</td>
</tr>
<tr>
<td>Heavy-load robot</td>
<td>0.3 0.3 2.0 0.3 0.3 0.2</td>
<td>0.3 0.3 2.0 0.3 0.3 0.2</td>
<td>3.0 3.0 3.0 3.0 3.0 3.0</td>
<td>0.3 0.3 3.0 0.3 0.3 0.2</td>
<td>4.0 4.0 4.0 4.0 4.0 4.0</td>
</tr>
<tr>
<td>Human</td>
<td>0.3 0.3 2.0 0.3 0.3 0.2</td>
<td>0.3 0.3 2.0 0.3 0.3 0.2</td>
<td>3.0 3.0 3.0 3.0 3.0 3.0</td>
<td>1.0 1.0 1.0 1.0 1.0 1.0</td>
<td>4.0 4.0 4.0 4.0 4.0 4.0</td>
</tr>
<tr>
<td>Powered hand truck</td>
<td>1.5 1.5 1.5 1.5 1.5 1.0</td>
<td>2.0 2.0 2.0 2.0 2.0 1.5</td>
<td>0.5 0.5 0.5 0.5 0.5 0.5</td>
<td>2.5 2.5 2.5 2.5 2.5 2.5</td>
<td>3.5 3.5 3.5 3.5 3.5 3.5</td>
</tr>
<tr>
<td>Forklift truck</td>
<td>1.5 1.5 1.5 1.5 1.5 1.0</td>
<td>2.0 2.0 2.0 2.0 2.0 1.5</td>
<td>0.5 0.5 0.5 0.5 0.5 0.5</td>
<td>2.5 2.5 2.5 2.5 2.5 2.5</td>
<td>3.5 3.5 3.5 3.5 3.5 3.5</td>
</tr>
<tr>
<td>Roller bed belt conveyor</td>
<td>3.5 3.5 3.5 3.5 3.5 3.0</td>
<td>3.5 3.5 3.5 3.5 3.5 3.0</td>
<td>0.3 0.3 0.3 0.3 0.3 0.3</td>
<td>0.3 0.3 0.3 0.3 0.3 0.3</td>
<td>3.0 3.0 3.0 3.0 3.0 3.0</td>
</tr>
<tr>
<td>Light-duty belt conveyor</td>
<td>3.5 3.5 3.5 3.5 3.5 3.0</td>
<td>3.5 3.5 3.5 3.5 3.5 3.0</td>
<td>0.3 0.3 0.3 0.3 0.3 0.3</td>
<td>0.3 0.3 0.3 0.3 0.3 0.3</td>
<td>3.0 3.0 3.0 3.0 3.0 3.0</td>
</tr>
<tr>
<td>AGV</td>
<td>2.0 2.0 2.0 2.0 2.0 1.5</td>
<td>0.5 0.5 0.5 0.5 0.5 0.5</td>
<td>1.5 1.5 1.5 1.5 1.5 1.5</td>
<td>2.0 2.0 2.0 2.0 2.0 2.0</td>
<td>5.0 5.0 5.0 5.0 5.0 5.0</td>
</tr>
<tr>
<td>AS/RS</td>
<td>9.0 9.0 9.0 9.0 9.0 9.0</td>
<td>9.0 9.0 9.0 9.0 9.0 9.0</td>
<td>9.0 9.0 9.0 9.0 9.0 9.0</td>
<td>9.0 9.0 9.0 9.0 9.0 9.0</td>
<td>1.5 1.5 1.5 1.5 1.5 1.5</td>
</tr>
</tbody>
</table>

Table 11. Materials handling times, $t_{i,kc}$.  

38
Table 13. Solution to the materials handling system selection problem.
### Part Type 7

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>(un)Load/None</td>
<td>Light-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/Orientation</td>
<td>Powered hand truck</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/Orientation</td>
<td>Human</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>(un)Load/Orientation</td>
<td>Light-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>Powered hand truck</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/Position</td>
<td>Light-duty conveyor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S&amp;R/None</td>
<td>AS/RS</td>
</tr>
</tbody>
</table>

### Part Type 8

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>(un)Load/Orientation</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>AGV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>7</td>
<td>(un)Load/None</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>AGV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/Orientation</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S&amp;R/None</td>
<td>AS/RS</td>
</tr>
</tbody>
</table>

### Part Type 9

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>(un)Load/None</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>Powered hand truck</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/Position</td>
<td>Human</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>(un)Load/Orientation</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>Powered hand truck</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/Orientation</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S&amp;R/None</td>
<td>AS/RS</td>
</tr>
</tbody>
</table>

### Part Type 10

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>(un)Load/None</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/Quantity</td>
<td>Powered hand truck</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>(un)Load/Orientation</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/Quantity</td>
<td>Powered hand truck</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/Quantity</td>
<td>Roller bed belt conveyor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S&amp;R/Quantity</td>
<td>AS/RS</td>
</tr>
</tbody>
</table>

### Part Type 11

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>(un)Load/None</td>
<td>Light-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>Light-duty conveyor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>(un)Load/None</td>
<td>Light-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>Light-duty conveyor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/Position</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S&amp;R/None</td>
<td>AS/RS</td>
</tr>
</tbody>
</table>

### Part Type 12

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>(un)Load/Orientation</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>AGV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>7</td>
<td>(un)Load/None</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/Position</td>
<td>AGV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/Orientation</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S&amp;R/None</td>
<td>AS/RS</td>
</tr>
</tbody>
</table>

Table 12. (cont'd 1)
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>(un)Load/None</td>
<td>Light-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(re)Handle/Position</td>
<td>Light-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>Light-duty conveyor</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>(un)Load/Orientation</td>
<td>Light-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>Light-duty conveyor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/None</td>
<td>Light-duty conveyor</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>(un)Load/Orientation</td>
<td>Light-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/Quantity</td>
<td>Light-duty conveyor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/Quantity</td>
<td>Human</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>(un)Load/None</td>
<td>Light-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>Light-duty conveyor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/None</td>
<td>Light-duty conveyor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S&amp;R/None</td>
<td>AS/RS</td>
</tr>
</tbody>
</table>

Table 12. (cont'd 2)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>(un)Load/Orientation</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/None</td>
<td>Roller bed belt conveyor</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>(un)Load/Position</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transportation/Position</td>
<td>Powered hand truck</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection/Quantity</td>
<td>Roller bed belt conveyor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S&amp;R/Quantity</td>
<td>AS/RS</td>
</tr>
</tbody>
</table>

Table 14. Materials handling equipment utilization.

<table>
<thead>
<tr>
<th>MH Equipment</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light-load robot</td>
<td>0.4%</td>
</tr>
<tr>
<td>Heavy-load robot</td>
<td>0.3%</td>
</tr>
<tr>
<td>Human</td>
<td>0.8%</td>
</tr>
<tr>
<td>Powered hand truck</td>
<td>0.3%</td>
</tr>
<tr>
<td>Forklift truck</td>
<td>0.1%</td>
</tr>
<tr>
<td>Roller bed belt conveyor</td>
<td>0.1%</td>
</tr>
<tr>
<td>Light-duty belt conveyor</td>
<td>0.4%</td>
</tr>
<tr>
<td>AGV</td>
<td>0.9%</td>
</tr>
<tr>
<td>AS/RS</td>
<td>1.7%</td>
</tr>
</tbody>
</table>
A more unexpected result is that the inspection operation was assigned to the human. One might be surprised that a heavy part was assigned to the human. This obviously raises questions regarding potential ergonomic problems. However, for this example, the human was the best-suited option for an inspection operation requiring an orientation change. No other choice of materials handling equipment comes even close. In addition, humans are well suited to handle part types of high complexity and precision. Similar interpretations can be made for the remaining assignments of the part types.

At times, the results may not be as simply justified. This is because the actual assignments are a result of three ratings. As such, it is possible that a piece of equipment, with the highest rating for a certain key product variable, is not chosen for a part type exhibiting a high degree of that key product variable. This would be the case if that piece of equipment rates poorly for the materials handling operations required by the part type.
CHAPTER 5. SENSITIVITY ANALYSIS

The three parameters, $\hat{W}_{lt}$, $W_{el}$ and $W_{hie}$ are at the heart of the materials handling system selection model. These parameters are largely subjective ratings of relationships. Therefore, it is important that a sensitivity analysis be completed so that we may more clearly understand their impact on the model’s assignments.

Parameter $\hat{W}_{lt}$ is studied in great detail since it provides the link between product design and manufacturing design. Parameter $W_{el}$ is partly handled by drawing a parallel to the insight gained by the sensitivity analysis of $\hat{W}_{lt}$. Finally, some experimentation with parameter $W_{hie}$ demonstrates its impact on the assignments chosen by the model.

One can draw a hypothesis regarding the impact of the rating factors from an elementary analysis of the model’s objective function. Since the final assignments are largely dependent on the values of the three factors, in general one may expect that a single change to one of the factors will not cause an immediate change in assignments. As an example, parameters $\hat{W}_{lt}$ and $W_{el}$ are used in the evaluation of $C_{el}$, which measures their differences for the various key product variables. As such, one will expect that a small change in one of these parameters will not greatly impact the model’s assignments. One can predict that a significant change in one of these parameters must occur before the solution is changed. In other words, one may expect the model to react to changes in the factors, but without being overly sensitive to these changes. Of course that this hypothesis is based on generalized conclusions. There are always exceptions and if the rating that is being changed has an impact on a constraint that was “tight” at the solution, then we can expect to see a change in the model’s assignments.
These hypotheses were tested and the results from the sensitivity analysis follow below.

5.1. Sensitivity Analysis of $\hat{W}_H$

A sensitivity analysis was conducted on $\hat{W}_H$ to study the response of the model in terms of materials handling assignments to the differences in $\hat{W}_H$ ratings. A large number of runs were conducted. The full listing of these can be found in Appendix II. A subset of these is presented and discussed below.

The examples presented for the sensitivity analysis are based on the case study example. Each key product variable for part type $i = 2$ was assigned a value of 2 (indicating a low value of that key product variable) or 4 (indicating a high value of the key product variable). The total number of combinations, and therefore trials, were $2^5 = 32$. The $\hat{W}_H$ ratings for every other part type remained unchanged. All other ratings and data also remained unchanged from the case study example. This meant that the only parameters that changed over the trials were those for part type $i = 2$.

Table 15 presents the results of the test runs. Please note the meaning of the following abbreviations that are used in Table 15 to conserve space:

- C Complexity
- P Precision
- D Diversity
- BS Batch Size
- L/M D Linear/Mass Dimension

Each row in Table 15 represents a trial run. The assignments determined by the model for the required materials handling operations follow the $\hat{W}_H$ values used in that run.
<table>
<thead>
<tr>
<th>Run</th>
<th>C</th>
<th>P</th>
<th>D</th>
<th>B</th>
<th>S</th>
<th>L/M/D</th>
<th>(un)Load/none</th>
<th>(un)Load/Orientation</th>
<th>Transportation/None</th>
<th>Inspection/Orientation</th>
<th>S&amp;R/Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>Human</td>
<td>Human</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Human</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>22</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>Human</td>
<td>Human</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Light robot</td>
<td>Human</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>26</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>29</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>30</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>Human</td>
<td>Human</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>31</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
<tr>
<td>32</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
</tr>
</tbody>
</table>

Table 15. $\hat{\psi}_H$ sensitivity analysis for part type 2 - assignments.
From the results obtained and presented in the Table 15, one can analyze the model’s response to changes in $\hat{W}_{ii}$ values. As such, we can take pairs of runs and compare the resulting assignments. For example, if we look at runs #1 and #2, we can observe that the only value that was changed was that for linear/mass dimension. It was changed from a value of 2 (low score) to a value of 4 (high score). From the resulting assignments, we can see that for load/unload operations, the human was replaced by the heavy-load robot. As well, for the transportation operations, the light belt conveyor was replaced by the roller-bed belt conveyor. These constitute sound assignments since our part type was changed from light and small to heavy and large. The assignments for inspection and storage and retrieval operations remained unchanged. This indicates that these assignments remain the most appropriate for the part type.

When the value for batch size was increased (comparison of runs #1 and #3) and the values of all the other key product variables remained at 2, we observe that the only change in assignments were that for the load/unload operations. In this case, the human was replaced with the light-load robot. This seems appropriate since the robot is better suited to process larger batch sizes.

A comparison of runs #1 and #5, where only the value assigned to diversity is changed, shows that the original assignments remain optimal. This indicates that the chosen assignments for this part type, when all other values for the key product variables are set to 2, are optimal for both low and high values of diversity. However, the same is not true for the case when the values of all the other key product variables are set to 4. Comparing runs #28 and #32, we can observe that one of the assignments changes. For the transportation operation, the model chose an AGV for the case where the diversity
value was low (run #28) and the roller bed belt conveyor for the case where the diversity value was high (run #32). This makes sense since an AGV will usually require fixtures for the part type and as such, it is better suited to a part type of low diversity. On the other hand, the rolled bed belt conveyor can accommodate part types of high diversity quite easily.

When we change the value of precision while keeping the values of all other key product variables constant at a value of 2 (runs #1 and #9), we observe some changes in the assignments. The human was replaced by a light-load robot for the load/unload operations and the light-duty belt conveyor was replaced with an AGV for the transportation operations. An AGV is better suited to handle part types of high precision (compared to a conveyor), even though it is hard to justify its use on a part type of low complexity and low mass and dimension. However, given the set of materials handling equipment that our model was forced to choose from, it has determined that this piece of materials handling equipment is the best choice out of the set. One should also keep in mind that the model does not consider costs and as such, the model ignores any cost considerations from the assignment choices.

Finally, we can look at the case where only the value of complexity is changed. From observing the assignments for runs #1 and #17 we can conclude that the model responded to the change in complexity by choosing a different set of assignments. The increase in complexity resulted in the replacement of the human with a light load robot for the load/unload operations and of the light-duty belt conveyor with an AGV. The discussion presented for precision can be made in this case as well. While the AGV may
not be the ideal choice of materials handling equipment, it is the best suited out of the set of materials handling equipment that we made available to our model.

Many other comparisons can be made from the data presented in Table 15. To avoid repetitiveness, these are not presented here. There were also more examples (runs) performed. These are included in the Appendix II. Overall, one will find that the model responds correctly and as expected to changes in the values of the key product variables for the parameter $\tilde{W}_t$.

5.1.1. Detailed Sensitivity Analysis of $\tilde{W}_t$

The comparisons derived from Table 15 were for changes in $\tilde{W}_t$ of 2 units (values of 2 and 4). In many situations, a difference of 2 units was enough to result in new assignments. However, this was not always the case. It is worthwhile to investigate this observation more carefully to better understand how sensitive the model is to the changes in the key product variables.

Looking back to Table 8, we can see that part type $i = 2$ from our case study example has the following ratings for $\tilde{W}_t$:

<table>
<thead>
<tr>
<th>Part Type</th>
<th>Complexity</th>
<th>Precision</th>
<th>Diversity</th>
<th>Batch Size</th>
<th>Mass/Linear Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 16. $\tilde{W}_t$ ratings for part type $i = 2$.

To test what magnitude of a change in the key product variables is required to result in a new set of assignments, some test runs were performed for different values of complexity and mass/linear dimension.
In the first case, we ran the model for all possible values of complexity while keeping the values of the remaining key product variables unchanged. The results indicated that a new set of assignments would only be generated when the value of complexity was equal to or less than a value of 2. This implies that a change of 3 units (5 − 2) was necessary for the model to choose a new set of assignments. In the new set of assignments, the AGV was replaced with a roller bed belt conveyor.

For the second case, we tested for all possible values of mass/linear dimension while keeping the values of all other key product variables unchanged. In this case, it was found that a change occurred when the value of mass/linear dimension was lowered to 2 or less from the original value of 4. This indicates that only a change of 2 units was required for new assignments to be generated.

Other similar tests were conducted. The testing indicates that usually a change of 2 or 3 units is required before a new set of assignments is generated by the model. This means that the model responds to changes in the \( \hat{W}_{it} \) 's, but it is not overly sensitive. In other words, small changes in the product design are possible without the need for a new set of materials handling equipment. This mimics what one typically expects to find in a real world situation.

5.2. Sensitivity Analysis of \( W_{et} \)

A sensitivity analysis for \( W_{et} \) was conducted, but not as extensively as for \( \hat{W}_{it} \). A total of 10 runs were conducted. For the first set, the light-load robot was considered while its complexity rating varied from 1 to 5. In the second set of runs, the AGV was
studied as its mass/linear dimension rating was set to values between 1 and 5. This small set of runs is far from being an exhaustive study of the effect of this parameter on the solution. However, by studying the effects of two key product variables on two different types of materials handling equipment, one can draw some preliminary conclusions of interest.

5.2.1. Effect of the Complexity Rating of the Light-load Robot

In this set of experiments, the complexity rating assigned to the light-load robot (i.e. $W_{cr} = W_{11}$) was varied between a very low value of 1 and a very high value of 5. The effects of this change on the overall solution were studied. Since the light-load robot was only used for load/unload and handle/rehandle operations, these were the only operations where different assignments occurred. The results from the various runs are summarized in Table 17.

Some empirical conclusions can be drawn from observing Table 17. The most important conclusion is that the model responds to changes in the parameter $W_{cr}$, but it is not overly sensitive. For example, if we look at part type $i = 1$, we see that for a value of complexity equal to 1, the human is the optimal materials handling equipment. When the value of complexity is changed to 2, the human remains optimal, but the light-load robot is also optimal. For values of complexity equal to or greater than 3, the human is the optimal assignment. This type of occurrence is common. In most instances, when the complexity change leads to a new assignment, we encounter a situation where at the transition point, we have two optimal choices of materials handling equipment.
Also interesting is that for certain part types, the chosen materials handling equipment type remains unchanged for any value of complexity. This means that the chosen materials handling equipment is very compatible with the part type. Although we vary the value of complexity, the previously chosen materials handling equipment type remains the best-suited equipment type for the part type.

5.2.2. Effect of the Mass/Linear Dimension Rating of the AGV

For the second set of runs in the sensitivity analysis of parameter $W_{et}$, the mass/linear dimension rating of the AGV was tested at all levels between 1 and 5 (i.e. $W_{et} = W_{85}$). In this case, it is only necessary to study the assignments for transportation and inspection operations since these are the only two major materials handling operations that the AGV performs. The assignments obtained from the various runs are summarized in Table 18.

Observation of Table 18 reveals that a change in value for the mass/linear dimension rating of the AGV has almost no effect in the assignments determined by the model. There are only two part types for which a change in the value of mass/linear dimension affects the output of the model. These are part types $i = 4$ and $i = 6$. As it was found in the previous set of runs for the light-duty robot, a change in the final assignments involves a transition point where two pieces of materials handling equipment are optimal.
<table>
<thead>
<tr>
<th>Part</th>
<th>Material Handling Op.</th>
<th>Complexity = 1</th>
<th>Complexity = 2</th>
<th>Complexity = 3</th>
<th>Complexity = 4</th>
<th>Complexity = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(un)Load/Orientation</td>
<td>Human</td>
<td>Light-load robot/Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td>2</td>
<td>(un)Load/None</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
</tr>
<tr>
<td>3</td>
<td>(un)Load/Orientation</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>Robots(*)</td>
<td>High-load robot</td>
<td>High-load robot</td>
</tr>
<tr>
<td>4</td>
<td>(un)Load/Position</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
</tr>
<tr>
<td>5</td>
<td>(un)Load/None</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td>6</td>
<td>(un)Load/None</td>
<td>Human</td>
<td>Light-load robot/Human</td>
<td>Light-load robot</td>
<td>Light-load robot/Human</td>
<td>Human</td>
</tr>
<tr>
<td>8</td>
<td>(un)Load/None</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Light-load robot/Human</td>
<td>Light-load robot</td>
</tr>
<tr>
<td>9</td>
<td>(un)Load/Orientation</td>
<td>Robots(*)</td>
<td>Light-load robot</td>
<td>Robots(*)</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
</tr>
</tbody>
</table>

(*) Indicates that any of the available materials handling equipment of that type was optimal for the assignment.

Table 17. Sensitivity analysis results for $W_a = W_{11}$. 

53
<table>
<thead>
<tr>
<th>Part</th>
<th>Material Handling Op.</th>
<th>Complexity = 1</th>
<th>Complexity = 2</th>
<th>Complexity = 3</th>
<th>Complexity = 4</th>
<th>Complexity = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>(un)Load/None</td>
<td>Robots (*)</td>
<td>Light-load robot</td>
<td>Robots(*)</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td>(un)Load/Orientation</td>
<td>Robots (*)</td>
<td>Light-load robot</td>
<td>Robots(*)</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td>(un)Load/None</td>
<td>Light-load robot</td>
<td>Light-load robot</td>
<td>Light-load robot</td>
<td>Light-load robot</td>
<td>Light-load robot</td>
</tr>
<tr>
<td>12</td>
<td>(un)Load/Orientation</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td>(un)Load/None</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td>13</td>
<td>(un)Load/None</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Robots(*)</td>
<td>Light-load robot</td>
<td>Robots(*)</td>
</tr>
<tr>
<td></td>
<td>(re)Handle/Position</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Robots(*)</td>
<td>Light-load robot</td>
<td>Robots(*)</td>
</tr>
<tr>
<td></td>
<td>(un)Load/Orientation</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Robots(*)</td>
<td>Light-load robot</td>
<td>Robots(*)</td>
</tr>
<tr>
<td></td>
<td>(un)Load/None</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Robots(*)</td>
<td>Light-load robot</td>
<td>Robots(*)</td>
</tr>
<tr>
<td>14</td>
<td>(un)Load/Orientation</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
</tr>
<tr>
<td></td>
<td>(un)Load/Position</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
<td>Heavy-load robot</td>
</tr>
</tbody>
</table>

Table 17. (cont’d 1)
At the first look, it is somewhat surprising that the change of value in mass/linear dimension for the AGV does not affect the assignments more strongly. However, one must remember that mass/linear dimension is only one of the five key product variables. If the match between equipment and part type for the other four key product variables is good, then a change in mass/linear dimension alone may not sufficiently affect the model to result in a new set of assignments.

One can note that the AGV is only one of five available types of materials handling equipment that can be used for transportation and inspection operations. As such, it is very likely that a part type can be closely matched with one of the available types of equipment. If the compatibility between the part type and a piece of materials handling equipment is great enough, then a change in one of the key product variables is unlikely to cause a change in assignments.

In general, small changes in the rating of a key product variable for a given part type have little effect on the final set of assignments. Large changes in the rating are much more likely to affect the assignments from the model. However, if the compatibility between a part type and a piece of equipment is high, then a change in the value of just one of the key product variables is much less likely to result in a change of assignments.

5.3. Sensitivity Analysis of $W_{hi\epsilon}$

The final portion of the sensitivity analysis was performed on parameter $W_{hi\epsilon}$. 
<table>
<thead>
<tr>
<th>Part</th>
<th>Material Handling Op.</th>
<th>Dim./Mass = 1</th>
<th>Dim./Mass = 2</th>
<th>Dim./Mass = 3</th>
<th>Dim./Mass = 4</th>
<th>Dim./Mass = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Transportation/None</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
</tr>
<tr>
<td></td>
<td>Inspection/Orientation</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td>Transportation/None</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
</tr>
<tr>
<td></td>
<td>Inspection/None</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
</tr>
<tr>
<td>3</td>
<td>Transportation/Quantity</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
</tr>
<tr>
<td></td>
<td>Inspection/None</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
</tr>
<tr>
<td></td>
<td>Inspection/None</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
</tr>
<tr>
<td>4</td>
<td>Transportation/Orientation</td>
<td>Forklift</td>
<td>Forklift</td>
<td>Forklift/AGV(*)</td>
<td>AGV</td>
<td>Forklift/AGV(*)</td>
</tr>
<tr>
<td>5</td>
<td>Transportation/None</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
</tr>
<tr>
<td></td>
<td>Inspection/None</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
</tr>
<tr>
<td></td>
<td>Inspection/None</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
<td>Conveyors(*)</td>
</tr>
<tr>
<td>6</td>
<td>Transportation/None</td>
<td>Light-duty conv./AGV(*)</td>
<td>AGV</td>
<td>Light-duty conv./AGV(*)</td>
<td>Light-duty conv.</td>
<td>Light-duty conv.</td>
</tr>
<tr>
<td></td>
<td>Inspection/Orientation</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td>Inspection/Quantity</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td>7</td>
<td>Transportation/Orientation</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
</tr>
<tr>
<td></td>
<td>Inspection/Orientation</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td>Inspection/Position</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
</tr>
<tr>
<td>8</td>
<td>Transportation/None</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
</tr>
<tr>
<td></td>
<td>Inspection/Orientation</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
</tr>
<tr>
<td></td>
<td>Inspection/Orientation</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td>Inspection/Position</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td>9</td>
<td>Transportation/None</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
</tr>
<tr>
<td></td>
<td>Inspection/Position</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
</tr>
<tr>
<td></td>
<td>Inspection/Position</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
</tbody>
</table>

Table 18. Sensitivity analysis results for $W_\alpha = W_\beta$.  

56
<table>
<thead>
<tr>
<th>Part</th>
<th>Material Handling Op.</th>
<th>Dim./Mass = 1</th>
<th>Dim./Mass = 2</th>
<th>Dim./Mass = 3</th>
<th>Dim./Mass = 4</th>
<th>Dim./Mass = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Transportation/Quantity</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
</tr>
<tr>
<td></td>
<td>Transportation/Quantity</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
</tr>
<tr>
<td></td>
<td>Inspection/Quantity</td>
<td>Conveyors (*)</td>
<td>Conveyors (*)</td>
<td>Conveyors (*)</td>
<td>Conveyors (*)</td>
<td>Conveyors (*)</td>
</tr>
<tr>
<td></td>
<td>Inspection/Position</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td>12</td>
<td>Transportation/None</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
</tr>
<tr>
<td></td>
<td>Transportation/Position</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
<td>AGV</td>
</tr>
<tr>
<td></td>
<td>Inspection/Orientation</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td>Inspection/Quantity</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td></td>
<td>Transportation/Position</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
<td>Power handtruck</td>
</tr>
</tbody>
</table>

(*) Indicates that any of the available materials handling equipment of that type was optimal for the assignment.

Table 18. (cont’d 1)
Again, a total of ten runs, divided into two sets, were conducted. The first set studied the effect of the (un)Load/Orientation rating of the light-load robot (i.e. $W_{hhe} = W_{1,1}$) on the assignments. The second set of runs studied the effect of the Transportation/Quantity rating of the roller bed belt conveyor (i.e. $W_{hhe} = W_{330}$) on the materials handling assignments.

5.3.1. Effect of the (un)Load/Orientation Rating of the Light-load Robot

As can be observed from Table 19, the effects of the (un)Load/Orientation rating of the light robot were as predicted. The results from the test runs show that the model is capable of responding to changes in the $W_{hhe}$ ratings, without exhibiting over-sensitivity. The results also highlight the fact that the model behaves as expected by choosing the optimal assignments.

For most of the part types, the assignments remained unchanged for all values of the (un)Load/Orientation rating of the light robot. This indicates that the original assignments were the best possible assignments, while the next best assignments were far behind it. This also indicates that in order to cause a change in assignments, it is necessary to change more than one rating.

For the few part types where the original assignments were replaced by new assignments, the change was progressive. At the interface between assignments, two optimal assignments were chosen for a rating. An example of this is part type $i = 7$. From Table 19, it can be seen that the human is the preferred choice of materials handling
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(un)Load/Orientation</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td>2</td>
<td>(un)Load/Orientation</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
</tr>
<tr>
<td>3</td>
<td>(un)Load/Orientation</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
</tr>
<tr>
<td>5</td>
<td>(un)Load/Orientation</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
</tr>
<tr>
<td>6</td>
<td>(un)Load/Orientation</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Light-load robot/Human (*)</td>
<td>Light-load robot</td>
</tr>
<tr>
<td>7</td>
<td>(un)Load/Orientation</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Light-load robot/Human(*)</td>
<td>Light-load robot</td>
</tr>
<tr>
<td>8</td>
<td>(un)Load/Orientation</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Human</td>
<td>Light-load robot/Human(*)</td>
</tr>
<tr>
<td>9</td>
<td>(un)Load/Orientation</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
</tr>
<tr>
<td>10</td>
<td>(un)Load/Orientation</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
</tr>
<tr>
<td>12</td>
<td>(un)Load/Orientation</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
<td>High-load robot</td>
</tr>
</tbody>
</table>

Table 19. Sensitivity analysis results for $W_{he} = W_{hl}$. 
equipment for un(Load)/Orientation values between 1 and 3. For a rating equal to 4, the model chose two optimal pieces of materials handling equipment, the human and the light-load robot. When the rating is further increased to a value of 5, the optimal assignment becomes the light-load robot. Please note that part types $i = 6$ and $i = 11$ are not present in Table 18 because these part types do not require any (un)Load/Orientation operations.

5.3.2. Effect of the Transportation/Quantity Rating of the Roller Bed Belt Conveyor

The results obtained for the runs conducted to study the effect of the Transportation/Quantity rating of the roller bed belt conveyor are summarized in Table 20. Only those part types requiring any Transportation/Quantity operations are listed in the table. It is for this reason that part types $i = \{2, 5, 7, 8, 9, 11, 12\}$ are not listed in Table 20.

There is one case, that of part type $i = 3$, where at the transition point, two different types of materials handling equipment are optimal. However, for all other part types, there is no smooth transition as has been observed quite often in other runs of the sensitivity analysis. This happens because when the Transportation/Quantity rating of the roller bed belt conveyor is set equal to 5, it assumes the highest rating for that operation/suboperation combination of all types of materials handling equipment. Essentially, it tells the model that the roller bed belt conveyor is the best piece of materials handling equipment to perform the required materials handling functions,
which in this case is the Transportation/Quantity function. As such, there is a strong bias for this type of materials handling equipment to be assigned to part types requiring those functions.

Again, we can observe that in general, the model responds to changes in the rating value, but is not overly sensitive.
<table>
<thead>
<tr>
<th>Part</th>
<th>Material Handling Op.</th>
<th>Quantity = 1</th>
<th>Quantity = 2</th>
<th>Quantity = 3</th>
<th>Quantity = 4</th>
<th>Quantity = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Transportation/Quantity</td>
<td>Forklift/Light conv. (*)</td>
<td>Forklift/Light conv. (*)</td>
<td>Forklift/Light conv. (*)</td>
<td>Roller bed belt conv.</td>
<td>Roller bed belt conv.</td>
</tr>
<tr>
<td>10</td>
<td>Transportation/Quantity</td>
<td>Powered hand truck</td>
<td>Powered hand truck</td>
<td>Powered hand truck</td>
<td>Powered hand truck</td>
<td>Roller bed belt conv.</td>
</tr>
</tbody>
</table>

Table 20. Sensitivity analysis results for $W_{kk} = W_{zz}$

62
CHAPTER 6. THE SOLVER PROGRAM

A program was created to simplify the use of the developed models and to illustrate the potential for this work as an industrial application. The program was coded with Visual Basic 5.0 and interfaces with Microsoft Access 8.0 and Lingo 5.0. The program is capable of handling any problem size. Any limitations to problem size are imposed by the number of records that Microsoft Access 8.0 can handle and the number of constraints that Lingo 5.0 can solve. The larger the problem, the more random access memory will be required for the successful operation of Microsoft Access and Lingo. The full code listing can be found in Appendix IV.

The Solver program fulfills three major goals. It allows access to the models data for creation or modification, it solves the models and it produces reports containing the solution to those models. The data creation or editing is allowed through Microsoft Access. One single database called OA.MDB holds all the data that the two models require. The models are solved when the Solver program calls a Lingo runtime library that provides access to Lingo’s solving capabilities. A Lingo script file was created for each model (see Appendix I). This script file is passed by the Solver program to the Lingo runtime library. Finally the output reports are created and can be viewed and printed from within the program.

The program graphical interface is very simple and clean. The main form that is visible after launching the program can be seen in Figure 1. This form consists of several key components. There is a menu bar and its accompanying menu items. There are also two buttons that are used to solve the two models. Each button is accompanied by a
display area where the value of the objective function is displayed once the model is successfully solved. Finally, there is also an exit button that terminates the application.

![Figure 1. Main form of the Solver program.](image)

The various components in the main window are presented in more detail below.

### 6.1. The Menu Bar

The menu bar consists of a number of menu entries. The headings in the menu bar drop down to reveal one or more menu entries. Each menu entry performs a specific function. This is similar to nearly every other application in the Microsoft Windows platform that contains a menu bar. Each heading and the menu entries it contains are now described.
6.1.1. The **File** Heading

The **File** heading in the menu bar contains two menu entries: **Launch Access** and **Exit**. These entries can be seen in Figure 2.

![Figure 2. The File menu entries.](image)

The **Launch Access** menu entry will start Microsoft Access with the data file, *OA.MDB*, open for editing. This is the recommended way of editing the data when extensive modifications are required.

The **Exit** menu entry simply stops the program. It performs the same function as the **Exit** button at the bottom of the form.

6.1.2. The **Data** Heading

The **Data** heading contains a number of menu entries (see Figure 3). When clicked, each menu entry will open a corresponding data table from the database *OA.MDB*. Not all the tables in the database are shown. Some of the tables do not require
modifications. These include those listing the major materials handling operations and suboperations and the key product variables. In addition, the table containing the parameter $A_{sij}$ is not shown. The data in this table is filled in automatically after the operation allocation model is solved.

![OA and MHS Solving Tool](image)

**Figure 3. The Data menu entries.**

The list of the tables in the database *OA.MDB*, their corresponding menu entries in the menu bar and purpose are summarized in Table 21.

As mentioned above, each menu entry under the **Data** heading will open a database table. Figure 4 shows such a data table. In this case, the data table shown was opened by clicking on the menu entry **Machines**. Various operations can be performed within that table view. It is possible to add, delete or edit a record. Other miscellaneous operations are available through the button bar at the top of the form.
<table>
<thead>
<tr>
<th>Data menu entry</th>
<th>Table in database</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts</td>
<td>Parts</td>
<td>Lists the part types and associated demands</td>
</tr>
<tr>
<td>Machines</td>
<td>Machines</td>
<td>Lists the machines and associated capacity and setup costs</td>
</tr>
<tr>
<td>MH Equip</td>
<td>MH_Equip</td>
<td>Lists the materials handling equipment and time available</td>
</tr>
<tr>
<td>Mfg Operations</td>
<td>MfgOps</td>
<td>Lists the manufacturing operations</td>
</tr>
<tr>
<td>Plans</td>
<td>Plans</td>
<td>Lists the process plans</td>
</tr>
<tr>
<td>Part-Plan</td>
<td>PartPlan</td>
<td>Lists the possible Part-Plan combinations</td>
</tr>
<tr>
<td>Part-Plan-Op</td>
<td>PartPlanOp</td>
<td>Lists the possible Part-Plan-Manufacturing Operations combinations</td>
</tr>
<tr>
<td>MH Costs</td>
<td>MHC</td>
<td>Lists the materials handling costs of moving a part type from machine $j$ to machine $j$</td>
</tr>
<tr>
<td>W_it</td>
<td>W_it</td>
<td>Lists the ratings for the parameter $W_{it}$</td>
</tr>
<tr>
<td>W_et</td>
<td>W_et</td>
<td>Lists the ratings for the parameter $W_{et}$</td>
</tr>
<tr>
<td>W_hhe</td>
<td>W_hhe-Time</td>
<td>Lists the ratings for the parameter $W_{hhe}$ as well as the time required to perform the materials handling function $hh$ with equipment $e$</td>
</tr>
<tr>
<td>MH Requirements</td>
<td>MH_Req</td>
<td>Lists the materials handling requirements $\alpha_{hhej}$</td>
</tr>
</tbody>
</table>

Table 21. Data menu entries and associated database tables.
To add a new record, the user simply clicks on an empty row and adds the information. To delete a record, the user must highlight the desired row (record) and press the DEL or DELETE key in the keyboard. Other operations are possible and the button bar at the top of the form explicitly lists these.

A complete description of all the tables in the database along with their fields and requirements can be found in Appendix III.

6.1.3. The Reports Heading

This heading contains two menu entries as can be seen in Figure 5. The Operation Allocation entry will open an output report for the operation allocation model. Similarly, the Materials Handling menu entry will open an output report for the materials handling system selection model.
The reports are opened with Notepad, which allows for output to a printer. Figure 6 shows a sample report.

Every time the report file is written, its date and time of creation are written in the top line. The output from the model is then listed in an easy to understand format.

6.1.4. The Help Heading

The Help heading contains two menu entries as can be seen in Figure 7. The first menu entry is entitled Documentation. This entry will open a help file in the Microsoft Word format. The file is essentially the same as the current chapter of the thesis report with the addition of the information pertaining to the database OA.MDB, its tables and fields. This help document provides a quick and convenient way for the user to become familiar with the program.
### Machine Selection and Operation Assignments

<table>
<thead>
<tr>
<th>Part</th>
<th>Process Plan</th>
<th>Operation</th>
<th>Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Figure 6. Operation Allocation report window.

### OA and MHSS Solving Tool

- **Total Cost**: 0
- **Total**: 0

Figure 7. The *Help* menu entries.
The *About* menu entry opens a new window with general information about the Solver program.

### 6.2. The Buttons

There are three buttons in the main window. The boxed buttons are used to solve the two models, while the button at the bottom of the window will exit the program.

#### 6.2.1. The Model Solver Buttons

The labels on the buttons clearly indicate their purpose. The top button is labeled *Solve Operation Allocation Problem*. When pressed, the program is instructed to pass on instructions to the Lingo runtime library to solve the operation allocation model. The commands passed to Lingo are presented in a dialog box that is shown immediately after the button is clicked. Figure 8 illustrates this dialog box.

![solver dialog box](image)

*Figure 8. Dialog box illustrating the commands passed to Lingo.*

71
Once Lingo has solved the model and returned command to the Solver program, a dialog box is activated to inform the user about the successfulness of the operation. If Lingo was unable to solve the model, a dialog box warns the user that an error was encountered. If Lingo was successful, a dialog box such as the one in Figure 9 is observed.

![Dialog box](image)

**Figure 9.** Dialog box informing the user about Lingo's success in solving the model.

The main window is then updated to show the value of the objective function. This is shown in the label beside the solve button. Figure 10 shows the main window after the two models were solved. Please note that the values of the objective functions are shown to the right of the solver buttons.

The process to solve the materials handling system selection problem is similar to that described above for the operation allocation problem. Again, clicking on the button entitled **Solve Materials Handling Problem** will start the solving procedure.

The operation allocation problem must be solved prior to solving the materials handling system selection problem. This is necessary because the output from the operation allocation model is used as input to the materials handling system selection model. As such, the program does not allow the **Solve Materials Handling Problem** button to perform its action until the operation allocation model has been solved. Figure 11 shows a dialog box that is displayed to the user if the second button is pressed prior to solving the operation allocation problem.
6.2.2. The Exit Button

The third and final button is a simple *Exit* button. When pressed, this button causes the program to finish execution.
CHAPTER 7. DISCUSSIONS AND CONCLUSIONS

The models presented here are planning models that generate some of the design details required to draw a conceptual design of a manufacturing system. As such, these are not operational models. Currently, the models are applicable to a single cell. The objectives are to select groups of machines, load those machines to manufacture a specific group of parts and then select a suitable materials handling system to handle those parts at or in between the machines. The models were developed in the context of a flexible manufacturing system or cell.

The integrated approach described by this work consists of major phases that are completed in a sequential manner. In the first phase, the operation allocation model determines the optimal group of machines to process the part types and then assigns the operations of the part types to the machines. The assignments are done so that the total cost of operations, machine set-up and materials handling (transportation) is minimized. These assignments are then passed on to the materials handling system selection problem where they are used as input values.

In the second phase, the materials handling system selection model is solved. The materials handling system selection model determines the optimal group of materials handling equipment to perform all the required materials handling operations. The equipment is chosen based on its ability to perform the required materials handling functions and its compatibility to handle part types with certain key product characteristics. The model considers several important constraints imposed on the system such as biases towards equipment combinations and equipment capacity. The result of
the second phase is the selection of the best materials handing system to handle a given product-process combination.

The sensitivity analysis performed for the materials handling system selection model shows that the model responds to changes in the rating values (i.e., the ratings for $W_{klie}, W_{el}$ and $\tilde{W}_{el}$) but is not overly sensitive. In most situations, a value of four in one of the ratings will produce the same assignments as a value of five. In general, a difference of two points or greater is required for a change in assignments to occur. This finding is extremely important since the values for the rating factors are subjective and will undoubtedly differ from user to user. The fact that the model is not overly sensitive to changes in the ratings indicates that two users are likely to reach the same assignments even though the values they assign to the various ratings may differ somewhat.

The conclusions made from the limited sensitivity analysis need to be further confirmed. These conclusions were based on the analysis of a single and perhaps somewhat simple example. It would be extremely interesting to study how the models react to extreme examples with various "tight" constraints. Nonetheless, the preliminary results from the sensitivity analysis are very positive and encouraging.

A Visual Basic program was developed to ease the use of the models and demonstrate the potential of the work as a practical application in industry. The Solver program allows for data editing, solution to the operation allocation and materials handling system selection problems and report generation of the models’ output. The program allows for quick evaluation of what-if scenarios while providing a simple interface with a small learning curve, even for personnel not familiar with linear integer programming.
The results of the case study example show that the model can be successfully used as a tool for solving the joint operation allocation and materials handling system selection problems with product design considerations. The computational time required to solve the models was under one minute on a personal computer (AMD K6-2 350Mhz). The time will increase for larger and more realistic problems, but it is expected to remain computationally feasible.

The methodology presented here provides the decision-maker with a rapid modeling tool to solve the joint operation allocation and materials handling system selection problems. The Solver program that was developed is a first implementation of a rapid modeling tool to solve both problems. The two problems are solved sequentially at this point. There is a flow of information from the operation allocation problem to the materials handling system problem, but there is no feedback. This means that it is not possible to provide feedback to the operation allocation model and then rerun the models to fine-tune the overall solution. This is a recognized shortcoming of the present implementation of the model to be addressed in future research.

The current implementation of the model offers some flexibility not available in other models. For a given operation allocation solution, it is possible to experiment with the product-process combination. For instance, it is possible to identify the materials handling equipment best suited for a given product design. This is what was done in the case study example. However, it is also possible to study what product characteristics are best suited for a specific set or group of materials handling equipment. This provides a welcomed link between product design and manufacturing logistics design.
7.1. Future Research

The current work may be further developed and extended in a variety of ways. Of course, future researchers' interests will dictate the path, but some of the possible extensions to this work are discussed below.

The first and most natural extension to the work is to develop a feedback link between the materials handling system selection problem and the operation allocation problem. The main idea is to provide a mechanism by which the goal becomes to optimize the combined solution to the two problems, rather than the optimization of each individual problem. The current implementation of the models already allows for one-way communication between the models. The output from the operation allocation problem is used as input to the materials handling system selection. What remains to be done is to develop a similar link from the materials handling systems selection model to the operation allocation model. One could then enter a cyclic iterative process until the 'best' solution to both problems is obtained. Two great challenges to this extension are developing the feedback link and determining the condition or process that informs the models to stop iterating.

Another possible extension to the model, and one that requires serious consideration, is the development of a rigorous method of assigning values to the currently subjective rating factors of the materials handling system selection model. This will be a difficult task since the relationships that the ratings attempt to measure do not lend themselves easily to measurement.

The materials handling system selection model may also be extended by considering cost. Presently, the model makes no consideration for the cost of materials
handling equipment or the functions they perform. As such, one can readily think of the dilemma that occurs when the model must choose between two equally able pieces of materials handling equipment. Since they are both equally ‘able’, the model makes a random decision as to which one to choose. In reality, one would choose the lowest cost option. This is an extreme case, but also one that highlights a limitation of the current model. In such cases, it may make sense to extend the model to include cost considerations.

The Visual Basic program could also be further developed to improve its functionality and appeal as an industrial application. This would be of even greater importance once some of the work outlined above has been completed. When the complete integration of the operation allocation and materials handling system selection models has been fully addressed, it might be interesting to improve the Solver program to make it a viable product for engineers and managers. There are a number of areas where the program can be improved. These include user interface design, better handling of errors, user defined files and paths, etc.

There are other options for further development of the work presented. The ideas outlined above are just some of the more obvious extensions. It is this author’s opinion that the completion of those would greatly improve upon the work that has been achieved.
CHAPTER 8. CONCLUSIONS

The work described in this report represents an important step in the development of an integrated approach to manufacturing systems design. An existing operation allocation model (Atmani et al., 1995) was modified appropriately. The materials handling system selection model by Atmani and Dutta (1996) was extensively overhauled and reformulated for the context of this work. The two models were linked and solved sequentially. A large problem was solved and sensitivity analysis was carried out for the rating factors in the materials handling system selection model. This analysis showed that the model tolerates small differences in the values of the subjective ratings, without generating a new solution. A Visual Basic application was developed to simplify the use of the models. This practical application can be used as a rapid modeling tool by the decision-maker.

The integration of the operation allocation and materials handling system selection models is a very interesting and challenging problem. The work presented in this thesis is a first step in developing a thorough solution to that problem. It provides a base from which future researchers may further develop and extend the current solution to this integration problem.
References


Appendices

Appendix I.  Lingo Program Files
Appendix II. Sensitivity Analysis Results
Appendix III. Database Tables
Appendix IV. Solver Program Source Code
Appendix I. Lingo Program Files

Operation Allocation Model

MODEL:

!------------------------------------------ ;
! LINGO PROGRAM FOR OPERATION ALLOCATION MODEL ;
! AUTHOR: JORGE PAULO ;
! DATE: 1999-2000 ;
!------------------------------------------ ;

!------- SETS DEFINITION ------ ;
SETS:
  Part: D;
  MC: M, b, SC;
  Plan;
  Operation;
  PPlan (Part, Plan): Z;
  PPO (Part, Plan, Operation);
  PPOM (Part, Plan, Operation, MC): OC, t;
  PPOM1 (Part, Plan, Operation, MC);
  MH_Cost (Part, MC, MC): MHC;
  Lin (Operation, MC, MC, Part, Plan): L;
  Decision (Part, Plan, Operation, MC): X;
ENDSETS

!------- OBJECTIVE FUNCTION ------ ;

!-- Operating Cost ;
C1 =
  @SUM (PPOM(i,p,s,j):
    D(i) * OC(i,p,s,j) * X(i, p, s, j)
  )
;

!-- Setup Cost ;
C2 =
  @SUM (MC(j):
    SC(j) * M(j)
  )
;
!-- Material Handling Cost ;
C3 = @SUM (PPOM(i,p,s,j):
    @SUM (PPOM1(i1,p1,s1,j1) | i #EQ# i1 #AND# p #EQ# p1
    #AND# s1 #EQ# s+1 :
        MHC(i, j, j1) * L(s, j, j1, i, p)
    ) = 1;
);

!-- Actual Objective Function ;
[OBJ] MIN = C1 + C2 + C3 ;

!------- CONSTRAINT SET #1 ------- ;
@FOR (Part(i):
    @SUM (PPlan(i,p):
        Z(i, p)
    ) = 1;
);

!------- CONSTRAINT SET #2 ------- ;
@FOR (PPO(i,p,s):
    @SUM (PPOM(i,p,s,j):
        X(i, p, s, j)
    ) = Z(i, p);
);

!------- CONSTRAINT SET #3 ------- ;
@FOR (MC(j):
    @SUM (PPOM(i,p,s,j):
        X(i, p, s, j)
    ) >= M(j);
);

!------- CONSTRAINT SET #4 ------- ;
@FOR (MC(j): [TIME_CONSTRAINT]
    @SUM (PPOM(i,p,s,j):
        D(i) * t(i,p,s,j) * X(i, p, s, j)
    ) <= b(j) * M(j);
);

!------- CONSTRAINT SET #5 ------- ;
@FOR (PPOM(i,p,s,j):
    @FOR (PPOM1(i1,p1,s1,j1) | i #EQ# i1 #AND# p #EQ# p1
    #AND# s1 #EQ# s+1 :
        X(i, p, s, j) + X(i1, p1, s1, j1)
        - 2 * L(s, j, j1, i, p) >= 0
    )
);

84
!-------- CONSTRAINT SET #6 -------- ;
@FOR (PPOM(i,p,s,j):
    @FOR (PPOM1(i1,p1,s1,j1) | i #EQ# i1 #AND# p #EQ# p1
    #AND# s1 #EQ# s+1 :
        X(i, p, s, j) + X(i1, p1, s1, j1)
        - L(s, j, j1, i, p) <= 1
    );
);

!-------- INTEGER CONSTRAINTS -------- ;

! L must be integer ;
@FOR (Part(i):
    @FOR (Plan(p):
        @FOR (Operation(s):
            @FOR (MC(j):
                @FOR (MC(j):
                    @BIN (L(s, j, j, i, p))
                );
            );
        );
    );
);

! X must be integer ;
@FOR (Part(i):
    @FOR (Plan(p):
        @FOR (Operation(s):
            @FOR (MC(j):
                @BIN (X(i, p, s, j))
            );
        );
    );
);

! Z must be integer ;
@FOR (PPlan(i,p):
    @BIN(Z(i, p))
);

! M must be integer ;
@FOR (MC(j):
    @BIN(M(j))
);
!------- DATA DEFINITION -------;
DATA:
    Part = @ODBC('OA', 'PARTS', 'PART');
    D = @ODBC('OA', 'PARTS', 'DEMAND');
    MC = @ODBC('OA', 'MACHINES', 'MACHINE');
    b = @ODBC('OA', 'MACHINES', 'CAPACITY');
    SC = @ODBC('OA', 'MACHINES', 'SETUP_COST');
    Plan = @ODBC('OA', 'PLANS', 'PLAN');
    Operation = @ODBC('OA', 'MFGOPS', 'MFGOP');
    PPlan = @ODBC('OA', 'PARTPLAN', 'PART', 'PLAN');
    PPO = @ODBC('OA', 'PARTPLANOP', 'PART', 'PLAN', 'MFGOP');
    PPOM = @ODBC('OA', 'PARTPLANOPMC', 'PART', 'PLAN', 'MFGOP', 'MACHINE');
    PPOM1 = @ODBC('OA', 'PARTPLANOPMC', 'PART', 'PLAN', 'MFGOP', 'MACHINE');
    OC = @ODBC('OA', 'PARTPLANOPMC', 'MFGOP_COST');
    t = @ODBC('OA', 'PARTPLANOPMC', 'MFGOP_TIME');
    MHC = @ODBC('OA', 'MHC', 'MHC');
@POINTER(1) = OBJ;
@POINTER(2) = @STATUS();
ENDDATA

!------- THAT IS ALL, FOLKS! -------;
END

Materials Handling System Selection Model

MODEL:

!----------------------------------------------------------------------------------;
! Lingo program for the Materials Handling System Selection;
! Revised to explicitly list sets and generalize  
! Author: Jorge Paulo
! Date: 1999-2000
!----------------------------------------------------------------------------------;
!----------------------------------------------------------------------------------
!--------------- DEFINE THE SETS ---------------;
SETS:
    Part: Demand;
    MH_Equip: Avail, Time;
    MC;
    Mfg_Op;
    Tech;
    MH_Op;
    MH_Subop;

86
A (Mfg_Op, Part, MC);
Alpha (Part, Mfg_Op, MC, MH_Op, MH_Subop);
Rate_hhe(MH_Equip, MH_Op, MH_Subop): W_ehh, t;
Rate_et(MH_Equip, Tech): W_et;
Rate_it(Part, Tech): W_it;
! HH_Time (MH_Equip, MH_Op, MH_Subop): t;
Decision (Alpha, MH_Equip): Y;
ENDSETS

!---------------------------OBJECTIVE FUNCTION---------------------------;
[OBJ] MAX =
@SUM( Decision(i,s,j,h,hbar,e):
    W_ehh(e,h,hbar) * Y(i,s,j,h,hbar,e) *
    (1 - @SUM (Tech(t):
          @ABS(W_et(e,t) - W_it(i,t))) / 20
    )
);

!-----------------------------CONSTRAINT SET #1---------------------------;
@FOR( Alpha(i,s,j,h,hbar):
    @SUM( MH_Equip(e):
        Y(i,s,j,h,hbar,e)
    ) = 1;
);

!-----------------------------CONSTRAINT SET #2---------------------------;
! Blank for now;

!-----------------------------CONSTRAINT SET #3---------------------------;
@FOR( MH_Equip(e):
    @SUM( Alpha(i,s,j,h,hbar):
        Y(i,s,j,h,hbar,e)
    ) >= Avail(e);
);

!-----------------------------CONSTRAINT SET #4---------------------------;
@FOR( MH_Equip(e): [TIME_CONS]
    @SUM( Alpha(i,s,j,h,hbar):
        Demand(i) * t(e, h,hbar) * Y(i,s,j,h,hbar,e)
    ) <= Time(e) * Avail(e);
);

!-----------------------------Y must be binary---------------------------;
@FOR( Decision(i,s,j,h,hbar,e):
    @BIN( Y(i,s,j,h,hbar,e)
    );
);
!-----------------------Avail must be binary-----------------------
@FOR( MH_Equip(e):
    @BIN( Avail(e)
    );
);

!-----------------------DEFINE THE DATA-----------------------
DATA:
Part = @ODBC('OA', 'PARTS', 'PART');
Demand = @ODBC('OA', 'PARTS', 'DEMAND');
MH_Equip = @ODBC('OA', 'MH_EQUIP', 'MH_EQUIP');
Time = @ODBC('OA', 'MH_EQUIP', 'TIME');
MC = @ODBC('OA', 'MACHINES', 'MACHINE');
Mfg_OP = @ODBC('OA', 'MFGOPS', 'MFGOP');
Tech = @ODBC('OA', 'TECH', 'TECH');
MH_Op = @ODBC('OA', 'MH_Op', 'MH_Op');
MH_Subop = @ODBC('OA', 'MH_SUBOP', 'MH_SUBOP');
A = @ODBC('OA', 'A', 'MFG_OP', 'PART', 'MACHINE');
Alpha = @ODBC('OA', 'MH_REQ',
    'PART', 'MFG_OP', 'MACHINE', 'MH_Op', 'MH_SUBOP');
Rate_hhe = @ODBC('OA', 'W_HHE-TIME', 'MH_EQUIP',
    'MHOP', 'MHSUBOP');
W_ehh = @ODBC('OA', 'W_HHE-TIME', 'W_HHE');
Rate_et = @ODBC('OA', 'W_ET', 'MH_EQUIP', 'TECH');
W_et = @ODBC('OA', 'W_ET', 'W_ET');
Rate_it = @ODBC('OA', 'W_IT', 'PART', 'TECH');
W_it = @ODBC('OA', 'W_IT', 'W_IT');
t = @ODBC('OA', 'W_HHE-TIME', 'MH_TIME');
@POINTER(1) = OBJ;
@POINTER(2) = @STATUS();
ENDDATA

!-------------------THAT IS ALL-----------------------------
END
Appendix II. Sensitivity Analysis Results

Sensitivity Analysis of $\hat{W}_r$

An extensive study was conducted for the sensitive analysis of $\hat{W}_r$. The results from the runs performed are presented in Table 23. Please note that due to space limitations, some abbreviations were used. The legend for the abbreviations is presented below.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Complexity</td>
</tr>
<tr>
<td>P</td>
<td>Precision</td>
</tr>
<tr>
<td>D</td>
<td>Diversity</td>
</tr>
<tr>
<td>B S</td>
<td>Batch Size</td>
</tr>
<tr>
<td>L/M D</td>
<td>Linear/Mass Dimension</td>
</tr>
</tbody>
</table>

Table 22. List of abbreviations used in sensitivity analysis of $\hat{W}_r$.  

89
<table>
<thead>
<tr>
<th>C</th>
<th>P</th>
<th>D</th>
<th>S</th>
<th>B</th>
<th>L</th>
<th>M</th>
<th>D</th>
<th>(un)Load/Unload</th>
<th>Orientation</th>
<th>None Inspection/</th>
<th>Orientation</th>
<th>S&amp;B</th>
<th>Position</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 23. Results of sensitivity analysis for \( H_\alpha \) ratings - part type 2.
<table>
<thead>
<tr>
<th>C</th>
<th>P</th>
<th>D</th>
<th>B</th>
<th>S</th>
<th>L/M</th>
<th>(un)Load/none</th>
<th>(un)Load/Orientation</th>
<th>Transportation/None</th>
<th>Inspection/Orientation</th>
<th>S&amp;R/Position</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>324.85</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Light robot</td>
<td>Human</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>324.85</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>Human</td>
<td>Human</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>322.85</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>322.85</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>322.35</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>320.35</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>Human</td>
<td>Human</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>322.35</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>319.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>321.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>321.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>Human</td>
<td>Human</td>
<td>Power hand truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>321.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>Human</td>
<td>Human</td>
<td>Power hand truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>322.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Power hand truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>321.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Power hand truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>319.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>319.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>319.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Forklift truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>319.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>Human</td>
<td>Human</td>
<td>Forklift truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>319.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Forklift truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>320.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Forklift truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>319.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>321.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>322.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>322.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>320.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>321.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>321.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>321.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>319.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Power hand truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>322.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>Human</td>
<td>Human</td>
<td>Power hand truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>322.35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Power hand truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>320.85</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Power hand truck</td>
<td>Human</td>
<td>AS/RS</td>
<td>320.35</td>
<td></td>
</tr>
</tbody>
</table>

Table 23. (cont'd 1)
<table>
<thead>
<tr>
<th>C</th>
<th>P</th>
<th>D</th>
<th>B</th>
<th>S</th>
<th>L/M</th>
<th>D</th>
<th>(un)Load/none</th>
<th>(un)Load/Orientation</th>
<th>Transportation/None</th>
<th>Inspection/Orientation</th>
<th>S&amp;R/Position</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>324.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>324.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>324.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>324.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>327.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>328.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>326.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>Human</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>326.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>326.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>325.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>328.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>326.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>Human</td>
<td>Light robot</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>326.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>326.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>Light robot</td>
<td>Human</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>328.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>327.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>Light robot</td>
<td>Light robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>328.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>327.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>AGV</td>
<td>Human</td>
<td>AS/RS</td>
<td>329.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>Human</td>
<td>Human</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>327.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>Human</td>
<td>Human</td>
<td>Light belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>327.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>Heavy robot</td>
<td>Heavy robot</td>
<td>Roller belt conveyor</td>
<td>Human</td>
<td>AS/RS</td>
<td>327.85</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 23. (cont'd 2)
Appendix III. Database Tables

All the data required by the operation allocation and the materials handling system selection models is contained in one Microsoft Access database file – OA.MDB.

This database contains a number of data tables as can be seen in Table 24.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Model *</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts</td>
<td>OA, MHSS</td>
<td>Lists part types and their demands</td>
</tr>
<tr>
<td>Machines</td>
<td>OA, MHSS</td>
<td>Lists machines, their capacities (time available) and setup costs</td>
</tr>
<tr>
<td>Plans</td>
<td>OA</td>
<td>Lists the process plans</td>
</tr>
<tr>
<td>MfgOps</td>
<td>OA, MHSS</td>
<td>Lists the manufacturing operations</td>
</tr>
<tr>
<td>PartPlan</td>
<td>OA</td>
<td>Lists the possible Part-Plan combinations</td>
</tr>
<tr>
<td>PartPlanOp</td>
<td>OA</td>
<td>Lists the possible Part-Plan-Manufacturing Operations combinations</td>
</tr>
<tr>
<td>PartPlanOpMC</td>
<td>OA</td>
<td>Lists the possible Part-Plan-Manufacturing Operations-Machine combinations, their cost and time required</td>
</tr>
<tr>
<td>MHC</td>
<td>OA</td>
<td>Lists the cost of transporting each part type between any two machines</td>
</tr>
<tr>
<td>A</td>
<td>OA, MHSS</td>
<td>Lists the output (solution) from the OA problem - generated during runtime</td>
</tr>
<tr>
<td>MH_Equip</td>
<td>MHSS</td>
<td>Lists the available types of materials handling equipment</td>
</tr>
<tr>
<td>MH_Op</td>
<td>MHSS</td>
<td>Lists the major materials handling operations</td>
</tr>
<tr>
<td>MH_Subop</td>
<td>MHSS</td>
<td>Lists the materials handling suboperations</td>
</tr>
<tr>
<td>Tech</td>
<td>MHSS</td>
<td>Lists the key product variables</td>
</tr>
<tr>
<td>MH_Req</td>
<td>MHSS</td>
<td>Lists the required materials handling operations and suboperations</td>
</tr>
<tr>
<td>W_it</td>
<td>MHSS</td>
<td>Lists the ratings for the parameter $W_{it}$</td>
</tr>
<tr>
<td>W_et</td>
<td>MHSS</td>
<td>Lists the ratings for the parameter $W_{et}$</td>
</tr>
<tr>
<td>W_hhe-time</td>
<td>MHSS</td>
<td>Lists the ratings for the parameter $W_{hhe}$ and the time required to perform each materials handling operation-suboperation combination on each type of materials handling equipment</td>
</tr>
</tbody>
</table>

* Each data table is used in at least one of the models. Please note that OA refers to the Operation Allocation model and MHSS refers to the Materials Handling System Selection model.

Table 24. Description of the tables in the database OA.MDB.
Each data table in the database is composed of a number of fields. All tables contain an *AutoNumber* index that is used as the primary key. This field is used to uniquely identify each record in the table. Although the use of this field is not required, its presence allows for the construction of queries, forms and reports.

Each data table is now described in a tabular format.

**Data Table: Parts**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>Part</td>
<td>Text</td>
<td>$i$</td>
<td>Part type identifier</td>
</tr>
<tr>
<td>Demand</td>
<td>Number</td>
<td>$d_i$</td>
<td>Demand of part type $i$</td>
</tr>
</tbody>
</table>

Table 25. *OA.MDB* Database - description of the table: Parts.

**Data Table: Machines**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>Machine</td>
<td>Text</td>
<td>$j$</td>
<td>Machine identifier</td>
</tr>
<tr>
<td>Capacity</td>
<td>Number</td>
<td>$b_j$</td>
<td>Time available on machine $j$</td>
</tr>
<tr>
<td>Setup_Cost</td>
<td>Currency</td>
<td>$SC_j$</td>
<td>Setup Cost for machine $j$</td>
</tr>
</tbody>
</table>

Table 26. *OA.MDB* Database - description of the table: Machines.

**Data Table: Plans**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>Plan</td>
<td>Text</td>
<td>$p$</td>
<td>Process plan identifier</td>
</tr>
</tbody>
</table>

Table 27. *OA.MDB* Database - description of the table: Plans.
Data Table: MfgOps

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MfgOp_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>MfgOp</td>
<td>Text</td>
<td>$s$</td>
<td>Manufacturing operation identifier</td>
</tr>
</tbody>
</table>

Table 28. *OA.MDB* Database - description of the table: MfgOps.

Data Table: PartPlan

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartPlan_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>Part</td>
<td>Text</td>
<td>$i$</td>
<td>Part type identifier</td>
</tr>
<tr>
<td>Plan</td>
<td>Text</td>
<td>$p$</td>
<td>Process plan identifier</td>
</tr>
</tbody>
</table>

Table 29. *OA.MDB* Database - description of the table: PartPlan.

Data Table: PartPlanOp

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartPlanOp_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>Part</td>
<td>Text</td>
<td>$i$</td>
<td>Part type identifier</td>
</tr>
<tr>
<td>Plan</td>
<td>Text</td>
<td>$p$</td>
<td>Process plan identifier</td>
</tr>
<tr>
<td>MfgOp</td>
<td>Text</td>
<td>$s$</td>
<td>Manufacturing operation identifier</td>
</tr>
</tbody>
</table>


Data Table: PartPlanOpMC

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartPlanOpMC_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>Part</td>
<td>Text</td>
<td>$i$</td>
<td>Part type identifier</td>
</tr>
<tr>
<td>Plan</td>
<td>Text</td>
<td>$p$</td>
<td>Process plan identifier</td>
</tr>
<tr>
<td>MfgOp</td>
<td>Text</td>
<td>$s$</td>
<td>Manufacturing operation identifier</td>
</tr>
<tr>
<td>Machine</td>
<td>Text</td>
<td>$j$</td>
<td>Machine identifier</td>
</tr>
<tr>
<td>MfgOp_Cost</td>
<td>Currency</td>
<td>$OC_{gs}(ip)$</td>
<td>Cost of performing operation $s$ on machine $j$ for the combination $(ip)$</td>
</tr>
<tr>
<td>MfgOp_Time</td>
<td>Number</td>
<td>$t_{gs}(ip)$</td>
<td>Time required to perform operation $s$ on machine $j$ for the combination $(ip)$</td>
</tr>
</tbody>
</table>

Table 31. *OA.MDB* Database - description of the table: PartPlanOpMC.
Data Table: MHC

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHC_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>Part</td>
<td>Text</td>
<td>$i$</td>
<td>Part type identifier</td>
</tr>
<tr>
<td>Machine1</td>
<td>Text</td>
<td>$j$</td>
<td>Machine identifier</td>
</tr>
<tr>
<td>Machine2</td>
<td>Text</td>
<td>$j$</td>
<td>Machine identifier</td>
</tr>
<tr>
<td>MHC</td>
<td>Currency</td>
<td>$T_{ij}$</td>
<td>Cost of moving a unit of part type $i$ from machine $j$ to machine $j$</td>
</tr>
</tbody>
</table>

Table 32. OA.MDB Database - description of the table: MHC.

Data Table: A

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>MfgOp</td>
<td>Text</td>
<td>$s$</td>
<td>Manufacturing operation identifier</td>
</tr>
<tr>
<td>Part</td>
<td>Text</td>
<td>$i$</td>
<td>Part type identifier</td>
</tr>
<tr>
<td>Machine</td>
<td>Text</td>
<td>$j$</td>
<td>Machine identifier</td>
</tr>
</tbody>
</table>

Table 33. OA.MDB Database - description of the table: A.

Data Table: MH_Equip

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHEquip_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>MH_Equip</td>
<td>Text</td>
<td>$e$</td>
<td>Materials handling equipment identifier</td>
</tr>
<tr>
<td>MH_Equip_d</td>
<td>Text</td>
<td>--</td>
<td>Description of materials handling equipment identifier</td>
</tr>
<tr>
<td>Time</td>
<td>Number</td>
<td>$T_e$</td>
<td>Time available on materials handling equipment identifier</td>
</tr>
</tbody>
</table>

Table 34. OA.MDB Database - description of the table: MH_Equip.

Data Table: MH_Op

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHOp_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>MH_Op</td>
<td>Text</td>
<td>$h$</td>
<td>Materials handling operation identifier</td>
</tr>
<tr>
<td>MH_Op_d</td>
<td>Text</td>
<td>--</td>
<td>Description of materials handling operation identifier</td>
</tr>
</tbody>
</table>

### Data Table: MH_Subop

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHSubop_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>MH_Subop</td>
<td>Text</td>
<td>$h$</td>
<td>Materials handling suboperation identifier</td>
</tr>
<tr>
<td>MH_Subop_d</td>
<td>Text</td>
<td>--</td>
<td>Description of materials handling suboperation identifier</td>
</tr>
</tbody>
</table>

Table 36. *OAMDB Database - description of the table: MH_Subop.*

### Data Table: Tech

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tech_index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>Tech</td>
<td>Text</td>
<td>$t$</td>
<td>Key product variable identifier</td>
</tr>
<tr>
<td>Tech_d</td>
<td>Text</td>
<td>--</td>
<td>Description of key product variable identifier</td>
</tr>
</tbody>
</table>

Table 37. *OAMDB Database - description of the table: Tech.*

### Data Table: MH_Req

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH_Req_Index</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>Part</td>
<td>Text</td>
<td>$i$</td>
<td>Part type identifier</td>
</tr>
<tr>
<td>MfgOp</td>
<td>Text</td>
<td>$s$</td>
<td>Manufacturing operation identifier</td>
</tr>
<tr>
<td>Machine</td>
<td>Text</td>
<td>$j$</td>
<td>Machine identifier</td>
</tr>
<tr>
<td>MH(Op)</td>
<td>Text</td>
<td>$h$</td>
<td>Materials handling operation identifier</td>
</tr>
<tr>
<td>MH_Subop</td>
<td>Text</td>
<td>$h$</td>
<td>Materials handling suboperation identifier</td>
</tr>
</tbody>
</table>

Table 38. *OAMDB Database - description of the table: MH_Req.*
### Data Table: \( W_{it} \)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_{it} _Index )</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>Part</td>
<td>Text</td>
<td>( i )</td>
<td>Part type identifier</td>
</tr>
<tr>
<td>Tech</td>
<td>Text</td>
<td>( t )</td>
<td>Key product variable identifier</td>
</tr>
<tr>
<td>( W_{it} )</td>
<td>Number</td>
<td>( W_{it} )</td>
<td>Relative weight of the product variable ( t ) on part type ( i )</td>
</tr>
</tbody>
</table>

Table 39. *OA.MDB* Database - description of the table: \( W_{it} \).

### Data Table: \( W_{et} \)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_{et} _Index )</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>MH_Equip</td>
<td>Text</td>
<td>( e )</td>
<td>Materials handling equipment identifier</td>
</tr>
<tr>
<td>Tech</td>
<td>Text</td>
<td>( t )</td>
<td>Key product variable identifier</td>
</tr>
<tr>
<td>( W_{et} )</td>
<td>Number</td>
<td>( W_{et} )</td>
<td>Relative weight of the product variable ( t ) on materials handling equipment ( e )</td>
</tr>
</tbody>
</table>

Table 40. *OA.MDB* Database - description of the table: \( W_{et} \).

### Data Table: \( W_{hhe-time} \)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Index in Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_{hhe} _Index )</td>
<td>AutoNumber</td>
<td>--</td>
<td>Record index</td>
</tr>
<tr>
<td>MH_Equip</td>
<td>Text</td>
<td>( e )</td>
<td>Materials handling equipment identifier</td>
</tr>
<tr>
<td>MH_Op</td>
<td>Text</td>
<td>( h )</td>
<td>Materials handling operation identifier</td>
</tr>
<tr>
<td>MH_Subop</td>
<td>Text</td>
<td>( h )</td>
<td>Materials handling suboperation identifier</td>
</tr>
<tr>
<td>( W_{hhe} )</td>
<td>Number</td>
<td>( W_{hhe} )</td>
<td>Relative degree of capability of materials handling equipment ( e ) to perform the operation/sub-operation combination ( hh )</td>
</tr>
<tr>
<td>MH_Time</td>
<td>Number</td>
<td>( t_{hhe} )</td>
<td>Time for equipment ( e ) to perform materials handling operation/sub-operation ( hh )</td>
</tr>
</tbody>
</table>

Table 41. *OA.MDB* Database - description of the table: \( W_{hhe-time} \).
Please note that Lingo requires that all set members be of type *Text*, while attributes of those sets should be of type *Number* (type *Currency* is also acceptable).

Also, Lingo requires that the database be registered as a ODBC Data Source before it can be accessed. The process to register the database is explained in the Lingo 5.0 User Guide (pp. 278-282).
Appendix IV. Solver Program Source Code

The Solver program is composed of a number of different files. Visual Basic saves each form (form attributes and code) in a single file. Similarly, it also saves each module in a different file. In our case, the source code to Solver consists of 17 files. There is one project file, one module file and 15 form files. Table 42 summarizes the information about these files.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>solver.vbp</td>
<td>Project file – contains information about the various form files, module files, objects and compilation options</td>
</tr>
<tr>
<td>datacon.bas</td>
<td>Module file – contains general procedures required by the program (ex. report creation)</td>
</tr>
<tr>
<td>solver.frm</td>
<td>Form file – contains information and code related to the main form for the program</td>
</tr>
<tr>
<td>frmAbout.frm</td>
<td>Form file – contains information and code related to the About dialog box</td>
</tr>
<tr>
<td>frmMdgMC.frm</td>
<td>Form file – contains information and code related to the form where the machine data is displayed</td>
</tr>
<tr>
<td>frmMdgMfgOp.frm</td>
<td>Form file – contains information and code related to the form where the manufacturing operation data is displayed</td>
</tr>
<tr>
<td>frmMdgMHC.frm</td>
<td>Form file – contains information and code related to the form where the materials handling cost data is displayed</td>
</tr>
<tr>
<td>frmMdgMHEquip.frm</td>
<td>Form file – contains information and code related to the form where the materials handling equipment data is displayed</td>
</tr>
<tr>
<td>frmMdgMHReq.frm</td>
<td>Form file – contains information and code related to the form where the materials handling requirements data is displayed</td>
</tr>
<tr>
<td>frmMdgPartPlan.frm</td>
<td>Form file – contains information and code related to the form where the Part-Plan data is displayed</td>
</tr>
<tr>
<td>frmMdgPartPlanOp.frm</td>
<td>Form file – contains information and code related to the form where the Part-Plan-Manufacturing Operation data is displayed</td>
</tr>
<tr>
<td>frmMdgPartPlanOpMC.frm</td>
<td>Form file – contains information and code related to the form where the Part-Plan-Manufacturing Operation-Machine data is displayed</td>
</tr>
<tr>
<td>frmMdgParts.frm</td>
<td>Form file – contains information and code related to the form where the part type data is displayed</td>
</tr>
<tr>
<td>frmMdgPlan.frm</td>
<td>Form file – contains information and code related to the form where the process plan data is displayed</td>
</tr>
<tr>
<td>frmdgWet.frm</td>
<td>Form file – contains information and code related to the form where the $W_{el}$ data is displayed</td>
</tr>
<tr>
<td>frmdgWit.frm</td>
<td>Form file – contains information and code related to the form where the $W_{hi}$ data is displayed</td>
</tr>
<tr>
<td>frmdgWhhe.frm</td>
<td>Form file – contains information and code related to the form where the $W_{he}$ data is displayed</td>
</tr>
</tbody>
</table>

Table 42. Source code files in the Solver program.

The source code listing for each file in the project follows. The source code includes comments that explain what each section of the programming accomplishes.

Source Code for solver.vbp

Type=Exe
Reference="C:\00020430-0000-0000-C000-00000000046\#2.0#0\..\\WINDOWS\SYSTEM\StdOle2.tlb#OLE Automation Reference="C:\0025E01-0000-0000-C000-00000000046\#4.0#0\..\\PROGRAM FILES\COMMON FILES\MICROSOFT SHARED\DC\PROGRAM Files\Microsoft DAO 3.5 Object Library Object=(5E9E78A0-531B-11CF-91F6-C2863C385E30)#1.0#0; MSFLXGRD.OCX Object=(FAEE763-117E-101B-8933-0B002B2F45F5A)#1.0#0; DBlIST32.OCX Object=(0008C01-0000-0000-0000-00000000046)#1.0#0; DBGRID32.OCX Form=solver.frm Module=Module1; datacon.bas Form=frmdgMC.frm Form=frmdgParts.frm Form=frmdgMHEquip.frm Form=frmdgMfgOp.frm Form=frmdgPlan.frm Form=frmdgPartPlan.frm Form=frmdgPartPlanOp.frm Form=frmdgPartPlanOpMC.frm Form=frmdgMHC.frm Form=frmdgWit.frm Form=frmdgWet.frm Form=frmdgWhhe.frm Form=frmdgMHRreq.frm Form=frmdgAbout.frm IconForm="frmMain" Startup="frmMain" HelpFile="" Title="solver" ExeName32="solver.exe" Command32="" Name="Project1"
Source Code for datacon.bas

Attribute VB_Name = "Module1"
' Define some arrays, no size given for now
Public MyArray() As String
Public Equip() As String

Public Sub Reportf(FileName As String, Model As String)
' The procedure takes two arguments. The first is the
' name of the model's solution file, including the
' path. The second is an indicator that tells this
' procedure what model is calling it.
'
' This procedure sets up the report headings, parses the
' LINGO output, isolates the sub-string that contains the
' assignments and calls 2 other functions to complete the
' work necessary to write the report.

Dim TextLine As String, Indicator As String
Dim ReportFile As String
Dim n As Integer
' Set correct indicator for model and
' Prepare report headings
If Model = "MH" Then
    ' Y is the decision variable for the MH model
    Indicator = "Y"
    ' Define the name of the report file
    ReportFile = "C:THESIS_WORK\SOLVER\MH_REPORT.TXT"
    ' Open the report file
    Open ReportFile For Output As #2
    ' Print some header information
    Print #2, "Creation Date and Time: " & Now
    Print #2, ""
    Print #2, Tab(20); "Materials Handling Assignments"
    Print #2, "--------------------------------------";
    "--------------------------------------"
    Print #2, "Part"; Tab(8); "Mfg. Op."; Tab(19);
    "Machine"; Tab(29); "MH Operation"; Tab(44);
    "MH Suboperation"; Tab(62); "MH Equipment"
    Print #2, "--------------------------------------";
    "--------------------------------------"
Else
    ' If we are writing the report for the MHSS problem
    ' then the indicator will be X instead
    Indicator = "X"
    ReportFile = "C:THESIS_WORK\SOLVER\AO_REPORT.TXT"
    Open ReportFile For Output As #2
    Print #2, "Creation Date and Time: " & Now
    Print #2, ""
    Print #2, ""
    Print #2, "Machine Selection and Operation Assignments"
    Print #2, "-----------------------------------------------"
    Print #2, "Part"; Tab(8); "Process Plan"; Tab(23);
    "Operation"; Tab(35); "Machine"
    Print #2, "-----------------------------------------------"
End If

' Open LINGO output file - model solution
Open FileName For Input As #1

Do While Not EOF(#1) ' Loop until end of file.
    Line Input #1, TextLine ' Read line into variable.
    TextLine = Trim(TextLine) ' Remove leading and trailing spaces
    If Mid(TextLine, 1, 1) = Indicator Then
        TextLine = Trim(Left(TextLine, 22)) 'Take first 28 characters
        TextLine = Trim(Mid(TextLine, 3)) 'Trim "Y( " or "X("
        TextLine = Mid(TextLine, 1, Len(TextLine) - 1) 'Trim trailing
        Debug.Print TextLine ' Print to Debug window to check string
        ' Call procedure StringToArray with appropriate attributes
        If Model = "MH" Then
            ' We send the string TextLine and the size of the necessary
            ' array to handle it
            StringToArray TextLine, 6
        Else

        End If

        TextLine = "" ' Reset TextLine for next line
    End If

Loop
' The decision variable from the OA model only has
' four indices
StringToArray TextLine, 4
End If
' Call procedure that writes the report
InterpretArray Indicator
End If
Loop
' Add divider to report for formatting
If Indicator = "Y" Then
    Print #2, "------------------------------------------";
    "------------------------------------------";
    "----------"
Else
    Print #2, "------------------------------------------"
End If
' Close files
Close #1
Close #2
End Sub
Public Sub StringToArray(txtstring As String, z As Integer)
' This procedure decomposes the string into a set of values
' that are stored in an array. The comma is the delimiter.
Dim i As Integer, n As Integer
Dim start_location, divider_location As Integer
' Resize the array to accept the required number of indices
Redim MyArray(z)
' Get numbers into an array
' Set starting point
start_location = 1
i = 0
' Step through the string and everytime it encounters a number
' with the comma as the delimiter, then save it to an array element
Do
    divider_location = InStr(start_location, txtstring, ",", )
    If divider_location = 0 Then Exit Do
    MyArray(i) = _
        Mid(txtstring, start_location, divider_location - start_location)
    start_location = divider_location + 1
    i = i + 1
Loop
' get the last part of the string - last array element
If start_location < Len(txtstring) Then MyArray(i) _
    = Mid$(txtstring, start_location, Len(txtstring)-start_location+1)
End Sub
Public Sub InterpretArray(DecisionV As String)
' This procedure interprets the members of the array and
' writes the rest of output report.

Dim MHop As String, MHsubop As String

If DecisionV = "Y" Then

    Select Case Val(MyArray(3)) 'MH Operation
        Case 1
            MHop = "(un)Load"
        Case 2
            MHop = "(re)Handle"
        Case 3
            MHop = "Transportation"
        Case 4
            MHop = "Inspection"
        Case 5
            MHop = "S & R"
        Case Else
            MHop = MyArray(3)
    End Select

    Select Case Val(MyArray(4)) 'MH Suboperation
        Case 1
            MHsubop = "Orientation"
        Case 2
            MHsubop = "Position"
        Case 3
            MHsubop = "Quantity"
        Case 4
            MHsubop = "Timing"
        Case 5
            MHsubop = "Sequence"
        Case 6
            MHsubop = "None"
        Case Else
            MHsubop = MyArray(4)
    End Select

    'Print the solution output into human readable format
    Print #2, Tab(2); MyArray(0); Tab(11); MyArray(1); Tab(22); _
        MyArray(2); Tab(29); MHop; Tab(44); MHsubop; _
        Tab(62); Equip(MyArray(5) - 1)

Else

    ' Same, but for the OA model
    Print #2, Tab(2); MyArray(0); Tab(13); MyArray(1); Tab(26); _
        MyArray(2); Tab(37); MyArray(3)

    ' Let's use the opportunity to write the Table for parameter Asij
    ' This is table A in the database OA.MDB
        With frmMain.dbA.Recordset
            .AddNew
            !Mfg_Op = MyArray(2)
            !Part = MyArray(0)
            !Machine = MyArray(3)
            .Update
        End With

End With
End If
End Sub

Source Code for solver.frm

VERSION 5.00
Begin VB.Form frmMain
    BorderStyle = 1 'Fixed Single
    Caption = "OA and MHSS Solving Tool"
    ClientHeight = 3795
    ClientLeft = 150
    ClientTop = 720
    ClientWidth = 5550
    FillColor = &H00C0C0C0&
    ForeColor = &H80000000&
    LinkTopic = "Form1"
    MaxButton = 0 'False
    ScaleHeight = 3795
    ScaleWidth = 5550
    StartUpPosition = 3 'Windows Default
Begin VB.Data dbA
    Caption = "Data1"
    Connect = "Access"
    DatabaseName = "C:\thesis_work\solver\oa.mdb"
    DefaultCursorType= 0 'DefaultCursor
    DefaultType = 2 'UseODBC
    EOFAction = 2 'Add New
    Exclusive = 0 'False
    Height = 345
    Left = 360
    Options = 0
    ReadOnly = 0 'False
    RecordsetType = 1 'Dynaset
    RecordSource = "A"
    Top = 3120
    Visible = 0 'False
    Width = 1215
End
Begin VB.CommandButton cmdExit
    Caption = "Exit"
    Height = 495
    Left = 3960
    TabIndex = 2
    Top = 3000
    Width = 1335
End
Begin VB.CommandButton cmdMH
    Caption = "Solve Materials Handling Problem"
End
Height = 615
Left = 360
TabIndex = 1
Top = 1800
Width = 3495
End

Begin VB.CommandButton cmdOA
Caption = "Solve Operation Allocation Problem"
Height = 615
Left = 360
TabIndex = 0
Top = 480
Width = 3495
End

Begin VB.Shape Shape1
Height = 2415
Left = 120
Top = 240
Width = 5295
End

Begin VB.Label lblMHValue
Alignment = 2 "Center"
Caption = "0"
Height = 255
Left = 4320
TabIndex = 6
Top = 2160
Width = 735
End

Begin VB.Label lblMHCost
Alignment = 2 "Center"
Caption = "Total"
BeginProperty Font
Name = "MS Sans Serif"
Size = 8.25
Charset = 0
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 375
Left = 4080
TabIndex = 5
Top = 1800
Width = 1215
End

Begin VB.Label lblOACost
Alignment = 2 "Center"
Caption = "Total Cost"
BeginProperty Font
Name = "MS Sans Serif"
Size = 8.25
Charset = 0
Weight = 700
Underline = 0 'False
Italic = 0 'False
EndProperty
Height = 375
Left = 4080
TabIndex = 5
Top = 1800
Width = 1215
End
Strikethrough = 0 'False
EndProperty
Height = 375
Left = 4080
TabIndex = 4
Top = 480
Width = 1215
End
Begin VB.Label lblOAValue
    Alignment = 2 'Center
    Caption = "0"
    Height = 255
    Left = 4200
    TabIndex = 3
    Top = 840
    Width = 855
End
Begin VB.Menu mnuFile
    Caption = "&File"
    Begin VB.Menu mnuLaunchAccess
        Caption = "&Launch Access"
    End
End
Begin VB.Menu mnuExit
    Caption = "E&xit"
End
End
Begin VB.Menu mnuData
    Caption = "&Data"
    Begin VB.Menu mnuParts
        Caption = "Parts"
    End
End
Begin VB.Menu mnuMachines
    Caption = "Machines"
End
Begin VB.Menu mnuMH_Equip
    Caption = "MH Equip"
End
Begin VB.Menu mnuMfgOp
    Caption = "Mfg Operations"
End
Begin VB.Menu mnuPlans
    Caption = "Plans"
End
Begin VB.Menu mnuPartPlan
    Caption = "Part-Plan"
End
Begin VB.Menu mnuPartPlanOp
    Caption = "Part-Plan-Op"
End
Begin VB.Menu mnuPartPlanOpMC
    Caption = "Part-Plan-Op-Machine"
End
Begin VB.Menu mnuMHCosts
    Caption = "MH Costs"
End
Begin VB.Menu mnuW_it
    Caption = "W_it"
End
End
Begin VB.Menu mnuW_et
  Caption = "W_et"
End
Begin VB.Menu mnuW_hhe
  Caption = "W_hhe"
End
Begin VB.Menu mnuMHReq
  Caption = "MH Requirements"
End

End
Begin VB.Menu mnuReports
  Caption = "Reports"
Begin VB.Menu mnuOA
  Caption = "Operation Allocation"
End
Begin VB.Menu mnuMH
  Caption = "Materials Handling"
End

End
Begin VB.Menu mnuHelp
  Caption = "Help"
Begin VB.Menu mnuDoc
  Caption = "Documentation"
End
Begin VB.Menu mnuAbout
  Caption = "About"
End

End

Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
  Dim oaflag As Integer

Private Declare Sub LGVBSCRIPT _
  Lib "lingodll.dll" _
  (ByVal cScript As String, _
  ByVal cLogFile As String, _
  nNumberOfTransferAreas As Long, _
  dTransferAreas As Double, _
  nTransferPtrs As Long, _
  nErrorCode As Long)

Private Sub cmdExit_Click()
  End
End Sub

Private Sub cmdMH_Click()

  ' Check if OA problem has been solved
  If oaflag = 1 Then

  ' Array to transfer data to and from LINGO
  Dim dTransferAreaa(2) As Double
' Point to where the transfer areas begin in the transfer array
Dim nTransferPointerss(2) As Long
nTransferPointerss(1) = 1    ' Objective function value
nTransferPointerss(2) = 2    ' Status of solution

' Start building LINGO's script (commands must be terminated
' with an ASCII 10)
Dim cScripttt As String

' Suppress solution reports
cScripttt = "SET TERSEO 1" & Chr(10)

' Set workspace memory limit to 16 Megs
cScripttt = cScripttt & "SET MXMEMB 16" & Chr(10)

' Save settings into the configuration file
cScripttt = cScripttt & "FREEZE" & Chr(10)

' Read in the model file
cScripttt = cScripttt & 
    & "TAKE C:\THESIS_WORK\SOLVER\MHSS_G2D.LNG" _
    & Chr(10)

' Solve the model
cScripttt = cScripttt & "GO" & Chr(10)

' Write a solution report
' cScripttt = cScripttt & 
    & "DIVERT C:\THESIS_WORK\SOLVER\MH_SOLUTION.TXT" & Chr(10)
cScripttt = cScripttt & "NONZ Y" & Chr(10)
cScripttt = cScripttt & "RVRT" & Chr(10)

' Quit LINGO DLL
' cScripttt = cScripttt & "QUIT" & Chr(10)

' Mark the end of the script with a null byte
' cScripttt = cScripttt & Chr(0)

' Route standard output to a file
Dim cLogFilee As String
cLogFilee = "C:\THESIS_WORK\SOLVER\MH_OUT.LOG"

' Show the commands sent to Lingo
Dim jj As Integer
jj = MsgBox(cScripttt)

' Change mouse pointer to busy
Screen.MousePointer = vbHourglass

' Run the LINGO script
Call LGVBSCRIPT(cScripttt, cLogFilee, 2, dTransferAreaa(1), _
    nTransferPointerss(1), nErrorCodee)

' Check for problems
Dim ii As Integer
If nErrorCode > 0 Or dTransferAreaa(2) > 0 Then
    ' We have problems, change mouse pointer to normal
    ' and show error box
    Screen.MousePointer = vbDefault
    ii = MsgBox("Unable to solve!")
Else
    ' No problems! Change mouse pointer back and
    ' print success message
    Screen.MousePointer = vbDefault
    ii = MsgBox("Success! All done...")
End If

' Put objective function value in dialog box
lblMHValue.Caption = dTransferAreaa(1)

' Write the report
reportf "C:\THESIS_WORK\SOLVER\MH_SOLUTION.TXT", "MH"

' If OA problem has not been solved
Dim l As Integer
Else
    l = MsgBox(_
        "Please solve the Operation Allocation Problem first!", , _
        "Error")
End If

End Sub

Private Sub cmdOA_Click()

' Array to transfer data to and from LINGO
Dim dTransferArea(2) As Double

' Point to where the transfer areas begin in the transfer array
Dim nTransferPointers(2) As Long
nTransferPointers(1) = 1   ' Objective function value
nTransferPointers(2) = 2   ' Status of solution

' Start building LINGO's script (commands must be terminated
' with an ASCII 10)
Dim cScript As String

' Supress solution reports
cScript = "SET TERSEO 1" & Chr(10)

' Set workspace memory limit to 16 Megs
cScript = cScript & "SET MXMEME 16" & Chr(10)

' Save settings into the configuration file
cScript = cScript & "FREEZE" & Chr(10)

' Read in the model file
cScript = cScript_
    & "TAKE C:\THESIS_WORK\SOLVER\OA_GD.LNG" _
    & Chr(10)
' Solve the model
cScript = cScript & "GO" & Chr(10)

' Write a solution report
cScript = cScript & _
"DIVERT C:\THESIS_WORK\SOLVER\OA SOLUTION.TXT" & Chr(10)
cScript = cScript & "NONZ X" & Chr(10)
cScript = cScript & "RVRT" & Chr(10)

' Quit LINGO DLL
cScript = cScript & "QUIT" & Chr(10)

' Mark the end of the script with a null byte
cScript = cScript & Chr(0)

' Route standard output to a file
Dim cLogFile As String
cLogFile = "C:\THESIS_WORK\SOLVER\OA_OUT.LOG"

' Show the commands sent to Lingo
Dim j As Integer
j = MsgBox(cScript)

' Change mouse pointer to busy
Screen.MousePointer = vbHourglass

' Run the LINGO script
Call LGVBSCRIPT(cScript, cLogFile, 2, dTransferArea(1), _
   nTransferPointers(1), nErrorCode)

' Check for problems
Dim k As Integer
If nErrorCode > 0 Or dTransferArea(2) > 0 Then
   ' We have problems, change mouse pointer to normal
   ' and show error box
   Screen.MousePointer = vbDefault
   k = MsgBox("Unable to solve!")
Else
   ' No problems! Change mouse pointer back and
   ' print success message
   Screen.MousePointer = vbDefault
   k = MsgBox("Success! All done...")
   ' Set oaflag to 1 so that we can now solve MHSS model
   oaflag = 1
End If

' Put objective function value in dialog box
lb10AValue.Caption = dTransferArea(1)

' Write the report
reportf "C:\THESIS_WORK\SOLVER\OA SOLUTION.TXT", "OA"

End Sub
Private Sub Label2_Click()
End Sub

Private Sub Form_Load()
    Dim EquipFile, nogood As String
    Dim i As Integer
    i = 0
    oaflag = 0
    EquipFile = "C:\THEESIS_WORK\SOLVER\EQUIP.TXT"
    Open EquipFile For Input As #1
    Do While Not EOF(1)
        Input #1, nogood
        i = i + 1
    Loop
    Close #1
    Debug.Print i
    ReDim Equip(i)
    Open EquipFile For Input As #1
    i = 0
    Do While Not EOF(1)
        Input #1, Equip(i)
        Debug.Print Equip(i)
        i = i + 1
    Loop
    Close #1
    ' On Error GoTo databaseProblem
    dbA.Refresh
    Dim currindex As Long
    With dbA.Recordset
        Do While Not .EOF
            .Delete
            .MoveNext
        Loop
    End With

' databaseProblem:
' MsgBox "Problem opening or using the " & _
' dbA.DatabaseName & " database."
' "OA and MHSS Solving Tool"
'End
End Sub

Private Sub Form_Unload(Cancel As Integer)
    dbA.Recordset.Close
End Sub

Private Sub mnuAbout_Click()
    frmAbout.Show vbModal, Me
End Sub
Private Sub mnuDoc_Click()
    DimRetVal
   RetVal = Shell("C:\PROGRAM FILES\MICROSOFT
OFFICE\OFFICE\WINWORD.EXE C:\THEESIS_WORK\SOLVER\SOLVER_HELP.DOC", 1)
    If RetVal = 0 Then
        MsgBox "Problem opening the report. " & _
        "Does the file: C:\THEESIS_WORK\SOLVER\SOLVER_HELP.DOC exist?", , _
        "Error Opening File"
    End If
End Sub

Private Sub mnuExit_Click()
    End
End Sub

Private Sub mnuLaunchAccess_Click()
    DimRetVal
   RetVal = Shell("C:\PROGRAM FILES\MICROSOFT
OFFICE\OFFICE\MSACCESS.EXE C:\THEESIS_WORK\SOLVER\OA.MDB", 1)
    If RetVal = 0 Then
        MsgBox "Problem opening the report. " & _
        "Does the file: C:\THEESIS_WORK\SOLVER\OA.MDB exist?", , _
        "Error Opening File"
    End If
End Sub

Private Sub mnuMachines_Click()
    frmMachines.Show vbModal, Me
End Sub

Private Sub mnuMfgOp_Click()
    frmMfgOp.Show vbModal, Me
End Sub

Private Sub mnuMH_Click()
    DimRetVal
   RetVal = Shell("C:\WINDOWS\NOTEPAD.EXE
C:\THEESIS_WORK\SOLVER\MH_REPORT.TXT", 1)
    If RetVal = 0 Then
        MsgBox "Problem opening the report. " & _
        "Does the file: C:\THEESIS_WORK\SOLVER\MH_REPORT.TXT exist?", , _
        "Error Opening File"
    End If
End Sub

Private Sub mnuMH_Equip_Click()
    frmMHEquip.Show vbModal, Me
End Sub

Private Sub mnuMHCosts_Click()
    frmMHCosts.Show vbModal, Me
End Sub

Private Sub mnuMHReq_Click()
    frmMHCReq.Show vbModal, Me
End Sub
Private Sub mnuOA_Click()
    Dim RetVal
    RetVal = Shell("C:\WINDOWS\NOTEPAD.EXE C:\THEESIS\WORK\SOLVER\OA_REPORT.TXT", 1)
    If RetVal = 0 Then
        MsgBox "Problem opening the report. " & 
        "Does the file: C:\THEESIS\WORK\SOLVER\OA_REPORT.TXT exist?", , 
        "Error Opening File"
    End If
End Sub

Private Sub mnuPartPlan_Click()
    frmmdgPartPlan.Show vbModal, Me
End Sub

Private Sub mnuPartPlanOp_Click()
    frmmdgPartPlanOp.Show vbModal, Me
End Sub

Private Sub mnuPartPlanOpMC_Click()
    frmmdgPartPlanOpMC.Show vbModal, Me
End Sub

Private Sub mnuParts_Click()
    frmmdgParts.Show vbModal, Me
End Sub

Private Sub mnuPlans_Click()
    frmmdgPlan.Show vbModal, Me
End Sub

Private Sub mnuWet_Click()
    frmmdgWet.Show vbModal, Me
End Sub

Private Sub mnuWhe_Click()
    frmmdgWhhe.Show vbModal, Me
End Sub

Private Sub mnuWit_Click()
    frmmdgWit.Show vbModal, Me
End Sub

Source Code for frmAbout.frm

VERSION 5.00
Begin VB.Form frmAbout
    BorderStyle = 3 'Fixed Dialog
Caption = "About MyApp"
ClientHeight = 3555
ClientLeft = 2340
ClientTop = 1935
ClientWidth = 5730
ClipControls = 0 'False
LinkTopic = "Form2"
MaxButton = 0 'False
MinButton = 0 'False
ScaleHeight = 2453.724
ScaleMode = 0 'User
ScaleWidth = 5380.766
ShowInTaskbar = 0 'False
Begin VB.PictureBox picIcon
    AutoSize = -1 'True
    ClipControls = 0 'False
    Height = 540
    Left = 240
    Picture = "frmAbout.frx":0000
    ScaleHeight = 337.12
    ScaleMode = 0 'User
    ScaleWidth = 337.12
    TabIndex = 1
    Top = 240
    Width = 540
End
Begin VB.CommandButton cmdOK
    Cancel = -1 'True
    Caption = "OK"
    Default = -1 'True
    Height = 345
    Left = 4245
    TabIndex = 0
    Top = 2625
    Width = 1260
End
Begin VB.CommandButton cmdSysInfo
    Caption = "&System Info..."
    Height = 345
    Left = 4260
    TabIndex = 2
    Top = 3075
    Width = 1245
End
Begin VB.Label Label1
    Caption = "by Jorge Paulo (1999-2000)"
    Height = 255
    Left = 1080
    TabIndex = 7
    Top = 960
    Width = 3975
End
Begin VB.Line Line1
    BorderColor = &H00808080&
    BorderStyle = 6 'Inside Solid
    Index = 1
    X1 = 84.515
End
X2 = 5309.398
Y1 = 1687.583
Y2 = 1687.583
End

Begin VB.Label lblDescription
Caption = "Tool to solve the Operation Allocation and Materials Handling System Selection Problems. Requires LINGO and Microsoft Access."
ForeColor = &H00000000&
Height = 810
Left = 1050
TabIndex = 3
Top = 1485
Width = 3885
End

Begin VB.Label lblTitle
Caption = "OA and MHSS Solver"
BeginProperty Font
Name = "MS Sans Serif"
Size = 8.25
Charset = 0
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
ForeColor = &H00000000&
Height = 240
Left = 1050
TabIndex = 5
Top = 240
Width = 3885
WordWrap = -1 'True
End

Begin VB.Line Line1
BorderColor = &H00FFFFFF&
BorderWidth = 2
Index = 0
X1 = 96.6
X2 = 5309.398
Y1 = 1697.936
Y2 = 1697.936
End

Begin VB.Label lblVersion
Caption = "Version 0.1"
Height = 225
Left = 1080
TabIndex = 6
Top = 480
Width = 3885
End

Begin VB.Label lblDisclaimer
Caption = "Warning: Please read documentation and source code carefully to ensure proper operation."
ForeColor = &H00000000&
Height = 825
Left = 255
End
TabIndex = 4
Top = 2625
Width = 3750
End
End
Attribute VB_Name = "frmAbout"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

' Reg Key Security Options...
Const READ_CONTROL = &H200000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + 
    KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + 
    KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1 ' Unicode null terminated string
Const REG_DWORD = 4 ' 32-bit number

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey As Long, _
    ByVal lpSubKey As String, ByVal ulOptions As Long, _
    ByVal samDesired As Long, ByRef phResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByVal hKey As Long, _
    ByVal lpValueName As String, ByVal lpReserved As Long, _
    ByVal lpType As Long, ByVal lpData As String, _
    ByVal pcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub Form_Load()
Me.Caption = "About " & App.Title
    & App.Minor & "." & App.Revision
    '   lblTitle.Caption = App.Title
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr
    
    Dim rc As Long
    Dim SysInfoPath As String
    
    ' Try To Get System Info Program Path\Name From Registry...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO, _
    gREGVALSYSINFO, SysInfoPath) Then
        ' Try To Get System Info Program Path Only From Registry...
        ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, _
        gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC, SysInfoPath) Then
            ' Validate Existence Of Known 32 Bit File Version
            If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
                SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
            
            ' Error - File Can Not Be Found...
        Else
            GoTo SysInfoErr
        End If
        ' Error - Registry Entry Can Not Be Found...
    Else
        GoTo SysInfoErr
    End If

    Call Shell(SysInfoPath, vbNormalFocus)

    Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, _
    SubKeyRef As String, ByRef KeyVal As String) As Boolean
    Dim i As Long           ' Loop Counter
    Dim rc As Long          ' Return Code
    Dim hKey As Long        ' Handle To An Open Registry Key
    Dim hDepth As Long      ' 
    Dim KeyValType As Long  ' Data Type Of A Registry Key
    Dim tmpVal As String    ' Tempory Storage For A Registry Key Value
    Dim KeyValSize As Long  ' Size Of Registry Key Variable
    '-----------------------------------------------
    ' Open RegKey Under KeyRoot (HKEY_LOCAL_MACHINE...)
    '-----------------------------------------------
    ' Open Registry Key
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey)
    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyErr  ' Handle Error...
    tmpVal = String$(1024, 0)                 ' Allocate Variable Space
    KeyValSize = 1024                         ' Mark Variable Size
' Retrieve Registry Key Value...
rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
    KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value

If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError ' Handle Errors

' Win95 Adds Null Terminated String...
  If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then
' Null Found, Extract From String
    tmpVal = Left(tmpVal, KeyValSize - 1)
Else ' WinNT Does NOT Null Terminate String...
' Null Not Found, Extract String Only
    tmpVal = Left(tmpVal, KeyValSize)
End If

' Determine Key Value Type For Conversion...
Select Case KeyValType
    ' Search Data Types...
    Case REG_SZ
        ' Copy String Value
        KeyVal = tmpVal
    Case REG_DWORD
        ' Double Word Registry Key Data Type
        For i = Len(tmpVal) To 1 Step -1 ' Convert Each Bit
            KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))
        Next
        KeyVal = Format$"&h" + KeyVal ' Convert Double Word To String
End Select

GetKeyValue = True ' Return Success
rc = RegCloseKey(hKey) ' Close Registry Key
Exit Function

GetKeyError: ' Cleanup After An Error Has Occurred...
    KeyVal = "" ' Set Return Val To Empty String
    GetKeyValue = False ' Return Failure
    rc = RegCloseKey(hKey) ' Close Registry Key
End Function

Source Code for frmdgMC.frm

VERSION 5.00
Object = "{00028C01-0000-0000-0000-000000000046}"#1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmdgMC
    ClientHeight = 4590
    ClientLeft = 1650
    ClientTop = 1545
    ClientWidth = 6150

120
LinkTopic = "Form1"
ScaleHeight = 4590
ScaleMode = 0 'User
ScaleWidth = 6150

Begin VB.PictureBox picButtons
    Align = 1 'Align Top
    Appearance = 0 'Flat
    BorderStyle = 0 'None
    ForeColor = &H80000008&
    Height = 330
    Left = 0
    ScaleHeight = 330
    ScaleWidth = 6150
    TabIndex = 1
    TabStop = 0 'False
    Top = 0
    Width = 6150

Begin VB.CommandButton cmdClose
    Caption = "&Close"
    Height = 330
    Left = 4398
    TabIndex = 5
    Tag = "&Close"
    Top = 0
    Width = 1437
End

Begin VB.CommandButton cmdFilter
    Caption = "&Filter"
    Height = 330
    Left = 2924
    TabIndex = 4
    Tag = "&Filter"
    Top = 0
    Width = 1462
End

Begin VB.CommandButton cmdSort
    Caption = "&Sort"
    Height = 330
    Left = 1462
    TabIndex = 3
    Tag = "&Sort"
    Top = 0
    Width = 1462
End

Begin VB.CommandButton cmdRefresh
    Caption = "&Refresh"
    Height = 330
    Left = 0
    TabIndex = 2
    Tag = "&Refresh"
    Top = 0
    Width = 1462
End

End

Begin VB.Data Datal
    Caption = "Datal"
    Connect = "Access"
End
DatabaseName = ""  
DefaultCursorType= 0 'DefaultCursor  
DefaultType = 2 'UseODBC  
Exclusive = 0 'False  
Height = 300  
Left = 2505  
Options = 0  
ReadOnly = 0 'False  
RecordsetType = 1 'Dynaset  
RecordSource = ""  
Top = 2175  
Visible = 0 'False  
Width = 1140
End

Begin MSDBGrid.DBGrid grdDataGrid  
Align = 1 'Align Top  
Bindings = "frmdgMC.frx":0000  
Height = 3645  
Left = 0  
OleObjectBlob = "frmdgMC.frx":00CE  
TabIndex = 0  
Top = 330  
Width = 6150
End

End

Attribute VB_Name = "frmdgMC"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gsdatabase As String
Dim gsRecordsource As String

Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdFilter_Click()
    On Error GoTo FilterErr

    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim sFilterStr As String

    If Datal.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Filter a Table Recordset!", 48
        Exit Sub
    End If

    Set recRecordset1 = Datal.Recordset                'copy the recordset
sFilterStr = InputBox("Enter Filter Expression:")
If Len(sFilterStr) = 0 Then Exit Sub

Screen.MousePointer = vbHourglass
recRecordset1.Filter = sFilterStr
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
'establish the filter
Set Data1.Recordset = recRecordset2
'assign back to original recordset object

Screen.MousePointer = vbDefault
Exit Sub

FilterErr:
Screen.MousePointer = vbDefault
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
On Error GoTo RefErr

Data1.Recordset.Requery

Exit Sub

RefErr:
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdSort_Click()
On Error GoTo SortErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim SortStr As String

If Data1.RecordsetType = vbRSTypeTable Then
    Beep
    MsgBox "You Cannot Sort a Table Recordset!", 48
    Exit Sub
End If

Set recRecordset1 = Data1.Recordset
'copy the recordset

If Len(msSortCol) = 0 Then
    SortStr = InputBox("Enter Sort Column:")
    If Len(SortStr) = 0 Then Exit Sub
Else
    SortStr = msSortCol
End If

Screen.MousePointer = vbHourglass
recRecordset1.Sort = SortStr

'establish the Sort
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
Set Datal.Recordset = recRecordset2

Screen.MousePointer = vbDefault
Exit Sub

SortErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
    Dim bParmQry As Integer
    Dim qdfTmp As QueryDef
    gsdatabase = "C:\THEESIS_WORK\SOLVER\OA.MDB"
    gsRecordsource = "Machines"

    On Error GoTo LoadErr

    ' To Do
    ' gsDatabase is a global string that needs
    ' to be set by the startup sub for the app
    Datal.DatabaseName = gsdatabase
    ' gsRecordSource is a global string that needs
    ' to be set by the sub routine that loads this form
    Datal.RecordSource = gsRecordsource
    Datal.RecordsetType = 1     ' dynaset
    Datal.Options = 0
    Datal.Refresh

    If Len(Datal.RecordSource) > 50 Then
        Me.Caption = "SQL Statement"
    Else
        Me.Caption = Datal.RecordSource
    End If

    Exit Sub

LoadErr:
    MsgBox "Error:" & Err & " " & Err.Description
    Unload Me
End Sub

Private Sub Form_Resize()
    On Error Resume Next
    If Me.WindowState <> 1 Then
        grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
    End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
    If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub
Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
    If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
    'let's sort on this_column
    If Datal.RecordsetType = vbRSTypeTable Then Exit Sub

    'check for the use of the ctrl key for descending sort
    If mbCtrlKey Then
        msSortCol = "[" & Datal.Recordset(ColIndex).Name & "] desc"
        mbCtrlKey = 0 'reset it
    Else
        msSortCol = "[" & Datal.Recordset(ColIndex).Name & "]"
    End If
    cmdSort_Click
    msSortCol = vbNullString 'reset it
End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer,_
    x As Single, Y As Single)
    mbCtrlKey = Shift
End Sub

Source Code for frmMfgOp.frm

VERSION 5.00
Object = "(00028C01-0000-0000-0000-000000000046)#1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmMfgOp
    ClientHeight = 4590
    ClientLeft = 1650
    ClientTop = 1545
    ClientWidth = 6150
    LinkTopic = "Form1"
    ScaleHeight = 4590
    ScaleMode = 0 'User
    ScaleWidth = 6150
Begin VB.PictureBox picButtons
    Align = 1 'Align Top
    Appearance = 0 'Flat
    BorderStyle = 0 'None
    ForeColor = &H80000008&
    Height = 330
    Left = 0
    ScaleHeight = 330
End VB.PictureBox
ScaleWidth    =  6150
TabIndex      =  1
TabStop       =  0   'False
Top           =  0
Width         =  6150

Begin VB.CommandButton cmdClose
   Caption      =  "&Close"
   Height       =  330
   Left         =  4398
   TabIndex     =  5
   Tag          =  "&Close"
   Top          =  0
   Width        =  1437
End

Begin VB.CommandButton cmdFilter
   Caption      =  "&Filter"
   Height       =  330
   Left         =  2924
   TabIndex     =  4
   Tag          =  "&Filter"
   Top          =  0
   Width        =  1462
End

Begin VB.CommandButton cmdSort
   Caption      =  "&Sort"
   Height       =  330
   Left         =  1462
   TabIndex     =  3
   Tag          =  "&Sort"
   Top          =  0
   Width        =  1462
End

Begin VB.CommandButton cmdRefresh
   Caption      =  "&Refresh"
   Height       =  330
   Left         =  0
   TabIndex     =  2
   Tag          =  "&Refresh"
   Top          =  0
   Width        =  1462
End

End

Begin VB.Data Data1
   Caption        =  "Data1"
   Connect        =  "Access"
   DatabaseName   =  ""
   DefaultCursorType=  0   'DefaultCursor
   DefaultType    =  2   'UseODBC
   Exclusive      =  0   'False
   Height         =  300
   Left           =  2505
   Options        =  0
   ReadOnly       =  0   'False
   RecordsetType  =  1   'Dynaset
   RecordSource   =  ""
   Top            =  2175
   Visible        =  0   'False
Dim gsdatabase As String
Dim gsRecordsource As String
Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdFilter_Click()
On Error GoTo FilterErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim sFilterStr As String

If Datal.RecordsetType = vbRSTypeTable Then
    Beep
    MsgBox "You Cannot Filter a Table Recordset!", 48
    Exit Sub
End If

Set recRecordset1 = Datal.Recordset
'copy the recordset

sFilterStr = InputBox("Enter Filter Expression:")
If Len(sFilterStr) = 0 Then Exit Sub

Screen.MousePointer = vbHourglass
recRecordset1.Filter = sFilterStr
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
'establish the filter
Set Datal.Recordset = recRecordset2
'assign back to original recordset object

Screen.MousePointer = vbDefault
Exit Sub

FilterErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & ", " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
    On Error GoTo RefErr

    Data1.Recordset.Requery

    Exit Sub

RefErr:
    MsgBox "Error:" & Err & ", " & Err.Description
End Sub

Private Sub cmdSort_Click()
    On Error GoTo SortErr

    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim SortStr As String

    If Data1.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Sort a Table Recordset!", 48
        Exit Sub
    End If

    Set recRecordset1 = Data1.Recordset

    'copy the recordset

    If Len(msSortCol) = 0 Then
        SortStr = InputBox("Enter Sort Column:")
        If Len(SortStr) = 0 Then Exit Sub
    Else
        SortStr = msSortCol
    End If

    Screen.MousePointer = vbHourglass
    recRecordset1.Sort = SortStr

    'establish the Sort
    Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
    Set Data1.Recordset = recRecordset2

    Screen.MousePointer = vbDefault
    Exit Sub

SortErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & ", " & Err.Description
End Sub

128
Private Sub Form_Load()
    Dim bParmQry As Integer
    Dim qdfTmp As QueryDef

    gsdatabase = "C:\THESIS_WORK\SOLVER\OA.MDB"
    gsRecordSource = "MfgOp5"

    On Error GoTo LoadErr

    'To Do
    'gsDatabase is a global string that needs
    'to be set by the startup sub for the app
    Datal.DatabaseName = gsdatabase
    'gsRecordSource is a global string that needs
    'to be set by the sub routine that loads this form
    Datal.RecordSource = gsRecordSource
    Datal.RecordsetType = 1     'dynaset
    Datal.Options = 0
    Datal.Refresh

    If Len(Datal.RecordSource) > 50 Then
        Me.Caption = "SQL Statement"
    Else
        Me.Caption = Datal.RecordSource
    End If

    Exit Sub

LoadErr:
    MsgBox "Error:" & Err & " " & Err.Description
    Unload Me
End Sub

Private Sub Form_Resize()
    On Error Resume Next
    If Me.WindowState <> 1 Then
        grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
    End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
    If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
    If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
    'let's sort on this column
    If Datal.RecordsetType = vbRSTypeTable Then Exit Sub

    Exit Sub
End Sub
'check for the use of the ctrl key for descending sort
If mbCtrlKey Then
    msSortCol = "[" & Data1.Recordset(ColIndex).Name & "] desc"
    mbCtrlKey = 0 'reset it
Else
    msSortCol = "[" & Data1.Recordset(ColIndex).Name & "]"
End If
End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer,_
    x As Single, y As Single)
    mbCtrlKey = Shift
End Sub

Source Code for frmMHC.frm

VERSION 5.00
Object = "{00028C01-0000-0000-0000-00000000046}"#1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmMHC
    ClientHeight = 4590
    ClientLeft = 1650
    ClientTop = 1545
    ClientWidth = 6150
    LinkTopic = "Form1"
    ScaleHeight = 4590
    ScaleMode = 0 'User
    ScaleWidth = 6150
End VB.Form
Begin VB.PictureBox picButtons
    Align = 1 'Align Top
    Appearance = 0 'Flat
    BorderStyle = 0 'None
    ForeColor = &H80000008&
    Height = 330
    Left = 0
    ScaleHeight = 330
    ScaleWidth = 6150
    TabIndex = 1
    TabStop = 0 'False
    Top = 0
    Width = 6150
End VB.PictureBox
Begin VB.CommandButton cmdClose
    Caption = "&Close"
    Height = 330
    Left = 4398
    TabIndex = 5
    Tag = "&Close"
End VB.CommandButton
Top      =  0
Width    =  1437
End

Begin VB.CommandButton cmdFilter
  Caption    =  "&Filter"
  Height     =  330
  Left       =  2924
  TabIndex   =  4
  Tag        =  "&Filter"
  Top        =  0
  Width      =  1462
End

Begin VB.CommandButton cmdSort
  Caption    =  "&Sort"
  Height     =  330
  Left       =  1462
  TabIndex   =  3
  Tag        =  "&Sort"
  Top        =  0
  Width      =  1462
End

Begin VB.CommandButton cmdRefresh
  Caption    =  "&Refresh"
  Height     =  330
  Left       =  0
  TabIndex   =  2
  Tag        =  "&Refresh"
  Top        =  0
  Width      =  1462
End

End

Begin VB.Data Data1
  Caption =  "Data1"
  Connect =  "Access"
  DatabaseName =  ""
  DefaultCursorType=  0 'DefaultCursor
  DefaultType =  2 'UseODBC
  Exclusive =  0 'False
  Height    =  300
  Left      =  2505
  Options   =  0
  ReadOnly  =  0 'False
  RecordsetType =  1 'Dynaset
  RecordSource =  ""
  Top       =  2175
  Visible   =  0 'False
  Width     =  1140
End

Begin MSDBGrid.DBGrid grdDataGrid
  Align      =  1 'Align Top
  Bindings   =  "frmdgMHC.frx":0000
  Height     =  3645
  Left       =  0
  OleObjectBlob =  "frmdgMHC.frx":00CE
  TabIndex   =  0
  Top        =  330
  Width      =  6150
End
End
End
Attribute VB_Name = "frmdgMHC"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gdatabase As String
Dim gRecordsource As String

Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdFilter_Click()
On Error GoTo FilterErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim sFilterStr As String

If Datal.RecordsetType = vbRSTypeTable Then
    Beep
    MsgBox "You Cannot Filter a Table Recordset!", 48
    Exit Sub
End If

Set recRecordset1 = Datal.Recordset
'copy the recordset
sFilterStr = InputBox("Enter Filter Expression:")
If Len(sFilterStr) = 0 Then Exit Sub
Screen.MousePointer = vbHourglass
recRecordset1.Filter = sFilterStr
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
'establish the filter
Set Datal.Recordset = recRecordset2
'assign back to original recordset object
Screen.MousePointer = vbDefault
Exit Sub

FilterErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & ", " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
On Error GoTo RefErr

Datal.Recordset.Requery

132
Exit Sub

RefErr:
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdSort_Click()
On Error GoTo SortErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim SortStr As String

If Data1.RecordsetType = vbRSTypeTable Then
    Beep
    MsgBox "You Cannot Sort a Table Recordset!", 48
Exit Sub
End If

Set recRecordset1 = Data1.Recordset
'copy the recordset

If Len(msSortCol) = 0 Then
    SortStr = InputBox("Enter Sort Column:")
    If Len(SortStr) = 0 Then Exit Sub
Else
    SortStr = msSortCol
End If

Screen.MousePointer = vbHourglass
recRecordset1.Sort = SortStr

'establish the Sort
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
Set Data1.Recordset = recRecordset2

Screen.MousePointer = vbDefault
Exit Sub

SortErr:
Screen.MousePointer = vbDefault
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
Dim bParmQry As Integer
Dim qdfTmp As QueryDef

gsdatabase = "C:\THESIS_WORK\SOLVER\QA.MDB"
gsRecordsource = "MHC"

On Error GoTo LoadErr

'To Do
'gsDatabase is a global string that needs
'to be set by the startup sub for the app
Data1.DatabaseName = gsdatabase
'gsRecordSource is a global string that needs
'to be set by the sub routine that loads this form
Data1.RecordSource = gsRecordSource
Data1.RecordsetType = 1   'dynaset
Data1.Options = 0
Data1.Refresh

If Len(Data1.RecordSource) > 50 Then
    Me.Caption = "SQL Statement"
Else
    Me.Caption = Data1.RecordSource
End If

Exit Sub

LoadErr:
    MsgBox "Error:" & Err & "  " & Err.Description
    Unload Me
End Sub

Private Sub Form_Resize()
    On Error Resume Next
    If Me.WindowState <> 1 Then
        grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
    End If
    Exit Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
    If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
    If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
    'let's sort on this column
    If Data1.RecordsetType = vbRSTypeTable Then Exit Sub

    'check for the use of the ctrl key for descending sort
    If mbCtrlKey Then
        msSortCol = "[" & Data1.Recordset(ColIndex).Name & "] desc"
    mbCtrlKey = 0 'reset it
    Else
    msSortCol = "[" & Data1.Recordset(ColIndex).Name & "]"
    End If
    cmdSort_Click
    msSortCol = vbNullString 'reset it

End Sub
Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer, _
    x As Single, y As Single)
    mbCtrlKey = Shift
End Sub

Source Code for frmdgMHEquip.frm

VERSION 5.00
Object = "{00028C01-0000-0000-0000-000000000046}"1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmdgMHEquip
    ClientHeight = 4590
    ClientLeft = 1650
    ClientTop = 1545
    ClientWidth = 6150
    LinkTopic = "Form1"
    ScaleHeight = 4590
    ScaleMode = 0 'User
    ScaleWidth = 6150
Begin VB.PictureBox picButtons
    Align = 1 'Align Top
    Appearance = 0 'Flat
    BorderStyle = 0 'None
    ForeColor = &H80000008&
    Height = 330
    Left = 0
    ScaleHeight = 330
    ScaleWidth = 6150
    TabIndex = 1
    TabStop = 0 'False
    Top = 0
    Width = 6150
Begin VB.CommandButton cmdClose
    Caption = "&Close"
    Height = 330
    Left = 4398
    TabIndex = 5
    Tag = "&Close"
    Top = 0
    Width = 1437
End
Begin VB.CommandButton cmdFilter
    Caption = "&Filter"
    Height = 330
    Left = 2924
    TabIndex = 4
    Tag = "&Filter"
    Top = 0
    Width = 1462
End
End
Begin VB.CommandButton cmdSort
  Caption = "&Sort"
  Height = 330
  Left = 1462
  TabIndex = 3
  Tag = "&Sort"
  Top = 0
  Width = 1462
End
Begin VB.CommandButton cmdRefresh
  Caption = "&Refresh"
  Height = 330
  Left = 0
  TabIndex = 2
  Tag = "&Refresh"
  Top = 0
  Width = 1462
End

End
Begin VB.Data Datal
  Caption = "Datal"
  Connect = "Access"
  DatabaseName = ""
  DefaultCursorType = 0 'DefaultCursor
  DefaultType = 2 'UseODBC
  Exclusive = 0 'False
  Height = 300
  Left = 2505
  Options = 0
  ReadOnly = 0 'False
  RecordsetType = 1 'Dynaset
  RecordSource = ""
  Top = 2175
  Visible = 0 'False
  Width = 1140
End
Begin MSDBGrid.DBGrid grdDataGrid
  Align = 1 'Align Top
  Bindings = "frmdgMHEquip.frx":0000
  Height = 3645
  Left = 0
  OleObjectBlob = "frmdgMHEquip.frx":00CE
  TabIndex = 0
  Top = 330
  Width = 6150
End

End
Attribute VB_Name = "frmdgMHEquip"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gsdatabase As String
Dim gsRecordsource As String
Dim mSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdFilter_Click()
    On Error GoTo FilterErr
    Dim recRecordset As Recordset, recRecordset2 As Recordset
    Dim sFilterStr As String
    If Datal.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Filter a Table Recordset!", 48
        Exit Sub
    End If
    Set recRecordset1 = Datal.Recordset
    'copy the recordset
    sFilterStr = InputBox("Enter Filter Expression:")
    If Len(sFilterStr) = 0 Then Exit Sub
    Screen.MousePointer = vbHourglass
    recRecordset1.Filter = sFilterStr
    Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
    'establish the filter
    Set Datal.Recordset = recRecordset2
    'assign back to original recordset object
    Screen.MousePointer = vbDefault
End Sub

FilterErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
    On Error GoTo RefErr
    Datal.Recordset.Requery
End Sub

RefErr:
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdSort_Click()
    On Error GoTo SortErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim SortStr As String

If Data1.RecordsetType = vbRSTypeTable Then
    Beep
    MsgBox "You Cannot Sort a Table Recordset!", 48
    Exit Sub
End If

Set recRecordset1 = Data1.Recordset
' copy the recordset

If Len(msSortCol) = 0 Then
    SortStr = InputBox("Enter Sort Column:")
    If Len(SortStr) = 0 Then Exit Sub
Else
    SortStr = msSortCol
End If

Screen.MousePointer = vbHourglass
recRecordset1.Sort = SortStr

' establish the sort
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
Set Data1.Recordset = recRecordset2

Screen.MousePointer = vbDefault
Exit Sub

SortErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
    Dim bParmQry As Integer
    Dim qdfTmp As QueryDef

    gsdatabase = "C:\THESIS_WORK\SOLVER\OA.MDB"
    gsRecordsource = "MH_Equip"

    On Error GoTo LoadErr

' To Do
' gsDatabase is a global string that needs
' to be set by the startup sub for the app
Data1.DatabaseName = gsdatabase
' gsRecordSource is a global string that needs
' to be set by the sub routine that loads this form
Data1.RecordSource = gsRecordsource
Data1.RecordsetType = 1 'dynaset
Data1.Options = 0
Data1.Refresh

If Len(Data1.RecordSource) > 50 Then
    Me.Caption = "SQL Statement"

End Sub
Else
    Me.Caption = Datal.RecordSource
End If

Exit Sub

LoadErr:
    MsgBox "Error:" & Err & " " & Err.Description
    Unload Me
End Sub

Private Sub Form_Resize()
    On Error Resume Next
    If MeWindowState <> 1 Then
        grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
    End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
    If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
    If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
    'let's sort on this column
    If Datal.RecordsetType = vbRSTypeTable Then Exit Sub

    'check for the use of the ctrl key for descending sort
    If mbCtrlKey Then
        msSortCol = "[" & Datal.Recordset(ColIndex).Name & "] desc"
        mbCtrlKey = 0 'reset it
    Else
        msSortCol = "[" & Datal.Recordset(ColIndex).Name & "]"
    End If
    cmdSort_Click
    msSortCol = vbNullString 'reset it
End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer, _
    x As Single, y As Single)
    mbCtrlKey = Shift
End Sub
Source Code for frmMHRReq.frm

VERSION 5.00
Object = "\00028C01-0000-0000-0000-000000000046\#1.0\#0"; "DBGRID32.OCX"
Begin VB.Form frmMHRReq
ClientHeight = 4590
ClientLeft = 1650
ClientTop = 1545
ClientWidth = 6150
LinkTopic = "Form1"
ScaleHeight = 4590
ScaleMode = 0 'User
ScaleWidth = 6150
Begin VB.PictureBox picButtons
    Align = 1 'Align Top
    Appearance = 0 'Flat
    BorderStyle = 0 'None
    ForeColor = &H80000008&
    Height = 330
    Left = 0
    ScaleHeight = 330
    ScaleWidth = 6150
    TabIndex = 1
    TabStop = 0 'False
    Top = 0
    Width = 6150
End
Begin VB.CommandButton cmdClose
    Caption = "&Close"
    Height = 330
    Left = 4398
    TabIndex = 5
    Tag = "&Close"
    Top = 0
    Width = 1437
End
Begin VB.CommandButton cmdFilter
    Caption = "&Filter"
    Height = 330
    Left = 2924
    TabIndex = 4
    Tag = "&Filter"
    Top = 0
    Width = 1462
End
Begin VB.CommandButton cmdSort
    Caption = "&Sort"
    Height = 330
    Left = 1462
    TabIndex = 3
    Tag = "&Sort"
    Top = 0
    Width = 1462
End
Begin VB.CommandButton cmdRefresh
    Caption = "&Refresh"
End
Height = 330
Left = 0
TabIndex = 2
Tag = "&Refresh"
Top = 0
Width = 1462
End
End

Begin Vb.Data Data1
Caption = "Data1"
Connect = "Access"
DatabaseName = ""
DefaultCursorType = 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 300
Left = 2505
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = ""
Top = 2175
Visible = 0 'False
Width = 1140
End

Begin MSDBGrid DBGrid grdDataGrid
Align = 1 'Align Top
Bindings = "frmdgMHRreq.frx";0000
Height = 3645
Left = 0
OleObjectBlob = "frmdgMHRreq.frx";00CE
TabIndex = 0
Top = 330
Width = 6150
End

Attribute VB_Name = "frmdgMHRreq"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gsdatabase As String
Dim gsRecordsource As String

Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdFilter_Click()
On Error GoTo FilterErr
Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim sFilterStr As String

If Datal.RecordsetType = vbRSTypeTable Then
  Beep
  MsgBox "You Cannot Filter a Table Recordset!", 48
  Exit Sub
End If

Set recRecordset1 = Datal.Recordset
'copy the recordset

sFilterStr = InputBox("Enter Filter Expression:")
If Len(sFilterStr) = 0 Then Exit Sub

Screen.MousePointer = vbHourglass
recRecordset1.Filter = sFilterStr
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
'establish the filter
Set Datal.Recordset = recRecordset2
'assign back to original recordset object

Screen.MousePointer = vbDefault
Exit Sub

FilterErr:
  Screen.MousePointer = vbDefault
  MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
  On Error GoTo RefErr
  Datal.Recordset.Requery
  Exit Sub

RefErr:
  MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdSort_Click()
  On Error GoTo SortErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim SortStr As String

If Datal.RecordsetType = vbRSTypeTable Then
  Beep
  MsgBox "You Cannot Sort a Table Recordset!", 48
  Exit Sub
End If

Set recRecordset1 = Datal.Recordset
'copy the recordset
If Len(msSortCol) = 0 Then
    SortStr = InputBox("Enter Sort Column:")
    If Len(SortStr) = 0 Then Exit Sub
Else
    SortStr = msSortCol
End If

Screen.MousePointer = vbHourglass
recRecordset1.Sort = SortStr

' establish the Sort
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
Set Data1.Recordset = recRecordset2

Screen.MousePointer = vbDefault
Exit Sub

SortErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
    Dim bParmQry As Integer
    Dim qdfTmp As QueryDef

    gsdatabase = "C:\THESIS_WORK\SOLVER\OA.MDB"
    gsRecordsource = "MH_Req"

    On Error GoTo LoadErr

    ' gsDatabase is a global string that needs
    ' to be set by the startup sub for the app
    Data1.DatabaseName = gsdatabase
    ' gsRecordsource is a global string that needs
    ' to be set by the sub routine that loads this form
    Data1.RecordSource = gsRecordsource
    Data1.RecordsetType = 1
    ' dynaset
    Data1.Options = 0
    Data1.Refresh

    If Len(Data1.RecordSource) > 50 Then
        Me.Caption = "SQL Statement"
    Else
        Me.Caption = Data1.RecordSource
    End If

    Exit Sub

LoadErr:
    MsgBox "Error:" & Err & " " & Err.Description
    Unload Me
End Sub

Private Sub Form_Resize()
On Error Resume Next
If Me.WindowState <> 1 Then
    grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
    If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
    If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
'let's sort on this column
If Data1.RecordsetType = vbRSTypeTable Then Exit Sub

'check for the use of the ctrl key for descending sort
If mbCtrlKey Then
    msSortCol = "[" & Data1.Recordset(ColIndex).Name & "] desc"
    mbCtrlKey = 0 'reset it
Else
    msSortCol = "[" & Data1.Recordset(ColIndex).Name & "]"
End If

cmdSort_Click
msSortCol = vbNullString 'reset it
End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer, _
    x As Single, y As Single)
    mbCtrlKey = Shift
End Sub

Source Code for frmdgPartPlan.frm

VERSION 5.00
Object = "\00028C01-0000-0000-0000-000000000046#1.0#0": "DBGRID32.OCX"
Begin VB.Form frmdgPartPlan
    ClientHeight = 4590
    ClientLeft = 1650
    ClientTop = 1545
    ClientWidth = 6150
End VB.Form
LinkTopic = "$Form1"
ScaleHeight = 4590
ScaleMode = 0 'User
ScaleWidth = 6150

Begin VB.PictureBox picButtons
    Align = 1 'Align Top
    Appearance = 0 'Flat
    BorderStyle = 0 'None
    ForeColor = &H80000008&
    Height = 330
    Left = 0
    ScaleHeight = 330
    ScaleWidth = 6150
    TabIndex = 1
    TabStop = 0 'False
    Top = 0
    Width = 6150
End

Begin VB.CommandButton cmdClose
    Caption = "$Close"
    Height = 330
    Left = 4398
    TabIndex = 5
    Tag = "$Close"
    Top = 0
    Width = 1437
End

Begin VB.CommandButton cmdFilter
    Caption = "$Filter"
    Height = 330
    Left = 2924
    TabIndex = 4
    Tag = "$Filter"
    Top = 0
    Width = 1462
End

Begin VB.CommandButton cmdSort
    Caption = "$Sort"
    Height = 330
    Left = 1462
    TabIndex = 3
    Tag = "$Sort"
    Top = 0
    Width = 1462
End

Begin VB.CommandButton cmdRefresh
    Caption = "$Refresh"
    Height = 330
    Left = 0
    TabIndex = 2
    Tag = "$Refresh"
    Top = 0
    Width = 1462
End
End

Begin VB.Data Data1
    Caption = "Data1"
    Connect = "Access"
DatabaseName = ""
DefaultCursorType = 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 300
Left = 2505
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = ""
Top = 2175
Visible = 0 'False
Width = 1140

End
Begin MSDBGrid=DBGrid grdDataGrid
  Align = 1 'Align Top
  Bindings = "frmdgPartPlan.frx":0000
  Height = 3645
  Left = 0
  OleObjectBlob = "frmdgPartPlan.frx":00CE
  TabIndex = 0
  Top = 330
  Width = 6150
End
End
Attribute VB_Name = "frmdgPartPlan"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gsdatabase As String
Dim gsRecordsource As String

Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
  Unload Me
End Sub

Private Sub cmdFilter_Click()
  On Error GoTo FilterErr
  Dim recRecordset1 As Recordset, recRecordset2 As Recordset
  Dim sFilterStr As String
  If Data1.RecordsetType = vbRSTypeTable Then
    Beep
    MsgBox "You Cannot Filter a Table Recordset!", 48
    Exit Sub
  End If
  Set recRecordset1 = Data1.Recordset
  'copy the recordset
sFilterStr = InputBox("Enter Filter Expression:")
If Len(sFilterStr) = 0 Then Exit Sub

Screen.MousePointer = vbHourglass
recRecordset1.Filter = sFilterStr
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
'establish the filter
Set Datal.Recordset = recRecordset2
'assign back to original recordset object

Screen.MousePointer = vbDefault
Exit Sub

FilterErr:
Screen.MousePointer = vbDefault
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
On Error GoTo RefErr

Datal.Recordset.Requery

Exit Sub

RefErr:
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdSort_Click()
On Error GoTo SortErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim SortStr As String

If Datal.RecordsetType = vbRSTypeTable Then
    Beep
    MsgBox "You Cannot Sort a Table Recordset!", 48
    Exit Sub
End If

Set recRecordset1 = Datal.Recordset
'copy the recordset

If Len(msSortCol) = 0 Then
    SortStr = InputBox("Enter Sort Column:")
    If Len(SortStr) = 0 Then Exit Sub
Else
    SortStr = msSortCol
End If

Screen.MousePointer = vbHourglass
recRecordset1.Sort = SortStr

'establish the Sort
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
Set Data1.Recordset = recRecordset2

Screen.MousePointer = vbDefault
Exit Sub

SortErr:
Screen.MousePointer = vbDefault
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
Dim bParmQry As Integer
Dim qdfTmp As QueryDef

gsdatabase = "C:\THEESIS_WORK\SOLVER\OA.MDB"
gsRecordsource = "PartPlan"

On Error GoTo LoadErr

' To Do
' gsDatabase is a global string that needs
' to be set by the startup sub for the app
Data1.DatabaseName = gsdatabase
'
' gsRecordSource is a global string that needs
' to be set by the sub routine that loads this form
Data1.RecordSource = gsRecordsource
Data1.RecordsetType = 1 ' dynaset
Data1.Options = 0
Data1.Refresh

If Len(Data1.RecordSource) > 50 Then
   Me.Caption = "SQL Statement"
Else
   Me.Caption = Data1.RecordSource
End If

Exit Sub

LoadErr:
MsgBox "Error:" & Err & " " & Err.Description
Unload Me
End Sub

Private Sub Form_Resize()
On Error Resume Next
If MeWindowState <> 1 Then
   grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
   If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
      Cancel = True
   End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
    If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
    'let's sort on this column
    If Data1.RecordsetType = vbRSTypeTable Then Exit Sub

    'check for the use of the Ctrl key for descending sort
    If mbCtrlKey Then
        msSortCol = "(" & Data1.Recordset(ColIndex).Name & ") desc"
        mbCtrlKey = 0 'reset it
    Else
        msSortCol = "(" & Data1.Recordset(ColIndex).Name & ")"'
    End If
End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer, _
    x As Single, Y As Single)
    mbCtrlKey = Shift
End Sub

Source Code for frmdgPartPlanOp.frm

VERSION 5.00
Object = "{(00028C01-0000-0000-0000-000000000046)#1.0#0}"; "DBGRID32.OCX"
Begin VB.Form frmdgPartPlanOp
    ClientHeight = 4590
    ClientLeft = 1650
    ClientTop = 1545
    ClientWidth = 6150
    LinkTopic = "Form1"
    ScaleHeight = 4590
    ScaleMode = 0 'User
    ScaleWidth = 6150
Begin VB.PictureBox picButtons
    Align = 1 'Align Top
    Appearance = 0 'Flat
    BorderStyle = 0 'None
    ForeColor = &H80000008&
    Height = 330
    Left = 0
End VB.PictureBox
ScaleHeight = 330
ScaleWidth = 6150
TabIndex = 1
TabStop = 0 'False
Top = 0
Width = 6150
Begin VB.CommandButton cmdClose
  Caption = "&Close"
  Height = 330
  Left = 4398
  TabIndex = 5
  Tag = "&Close"
  Top = 0
  Width = 1437
End

Begin VB.CommandButton cmdFilter
  Caption = "&Filter"
  Height = 330
  Left = 2924
  TabIndex = 4
  Tag = "&Filter"
  Top = 0
  Width = 1462
End

Begin VB.CommandButton cmdSort
  Caption = "&Sort"
  Height = 330
  Left = 1462
  TabIndex = 3
  Tag = "&Sort"
  Top = 0
  Width = 1462
End

Begin VB.CommandButton cmdRefresh
  Caption = "&Refresh"
  Height = 330
  Left = 0
  TabIndex = 2
  Tag = "&Refresh"
  Top = 0
  Width = 1462
End
End

Begin VB.Data Datal
  Caption = "Datal"
  Connect = "Access"
  DatabaseName = ""
  DefaultCursorType= 0 'DefaultCursor
  DefaultType = 2 'UseODBC
  Exclusive = 0 'False
  Height = 300
  Left = 2505
  Options = 0
  ReadOnly = 0 'False
  RecordsetType = 1 'Dynaset
  RecordSource = ""
  Top = 2175
Visible = 0 'False
Width = 1140
End
Begin MSDBGrid.DGGrid grdDataGrid
Align = 1 'Align Top
Bindings = "frmdgPartPlanOp.frx":0000
Height = 3645
Left = 0
OleObjectBlob = "frmdgPartPlanOp.frx":00CE
TabIndex = 0
Top = 330
Width = 6150
End
End
Attribute VB_Name = "frmdgPartPlanOp"
Attribute VB_GlobaNameSpace = False
Attribute VB_Createable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gsdatabase As String
Dim gsRecordsource As String

Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdFilter_Click()
On Error GoTo FilterErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim sFilterStr As String

If Datal.RecordsetType = vbRSTypeTable Then
Beep
MsgBox "You Cannot Filter a Table Recordset!", 48
Exit Sub
End If

Set recRecordset1 = Datal.Recordset
'copy the recordset

sFilterStr = InputBox("Enter Filter Expression:")
If Len(sFilterStr) = 0 Then Exit Sub

Screen.MousePointer = vbHourglass
recRecordset1.Filter = sFilterStr
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
'establish the filter
Set Datal.Recordset = recRecordset2
'assign back to original recordset object
Screen.MousePointer = vbDefault
Exit Sub

FilterErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
    On Error GoTo RefErr

        Data1.Recordset.Requery

    Exit Sub

RefErr:
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdSort_Click()
    On Error GoTo SortErr

    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim SortStr As String

    If Data1.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Sort a Table Recordset!", 48
        Exit Sub
    End If

    Set recRecordset1 = Data1.Recordset
    'copy the recordset

    If Len(msSortCol) = 0 Then
        SortStr = InputBox("Enter Sort Column:")
        If Len(SortStr) = 0 Then Exit Sub
    Else
        SortStr = msSortCol
    End If

    Screen.MousePointer = vbHourglass
    recRecordset1.Sort = SortStr

    'establish the Sort
    Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
    Set Data1.Recordset = recRecordset2

    Screen.MousePointer = vbDefault
    Exit Sub

SortErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub
Private Sub Form_Load()
    Dim bParmQry As Integer
    Dim qdfTmp As QueryDef
    gsdatabase = "C:\THEESIS WORK\SOLVER\OA.MDB"
    gsRecordSource = "PartPlanOp"

    On Error GoTo LoadErr

    'To Do
    'gsDatabase is a global string that needs
    'to be set by the startup sub for the app
    Datal.DatabaseName = gsdatabase
    'gsRecordSource is a global string that needs
    'to be set by the sub routine that loads this form
    Datal.RecordSource = gsRecordSource
    Datal.RecordsetType = 1 'dynaset
    Datal.Options = 0
    Datal.Refresh

    If Len(Datal.RecordSource) > 50 Then
        Me.Caption = "SQL Statement"
    Else
        Me.Caption = Datal.RecordSource
    End If

    Exit Sub

LoadErr:
    MsgBox "Error:" & Err & "" & Err.Description
    Unload Me
End Sub

Private Sub Form_Resize()    
    On Error Resume Next
    If Me.WindowState <> 1 Then
        grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
    End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
    If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
    If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
    'let's sort on this column
    If Datal.RecordsetType = vbRSTypeTable Then Exit Sub
Tag = "&Close"
Top = 0
Width = 1437
End
Begin VB.CommandButton cmdFilter
    Caption = "&Filter"
    Height = 330
    Left = 2924
    TabIndex = 4
    Tag = "&Filter"
    Top = 0
    Width = 1462
End
Begin VB.CommandButton cmdSort
    Caption = "&Sort"
    Height = 330
    Left = 1462
    TabIndex = 3
    Tag = "&Sort"
    Top = 0
    Width = 1462
End
Begin VB.CommandButton cmdRefresh
    Caption = "&Refresh"
    Height = 330
    Left = 0
    TabIndex = 2
    Tag = "&Refresh"
    Top = 0
    Width = 1462
End
End
Begin VB.Data Data1
    Caption = "Data1"
    Connect = "Access"
    DatabaseName = ""
    DefaultCursorType = 0 'DefaultCursor
    DefaultType = 2 'UseODBC
    Exclusive = 0 'False
    Height = 300
    Left = 2505
    Options = 0
    ReadOnly = 0 'False
    RecordsetType = 1 'Dynaset
    RecordSource = ""
    Top = 2175
    Visible = 0 'False
    Width = 1140
End
Begin MSDBGrid.DBGrid grdDataGrid
    Align = 1 'Align Top
    Bindings = "frmdgPartPlanOpMC.frx":0000
    Height = 3645
    Left = 0
    OleObjectBlob = "frmdgPartPlanOpMC.frx":00CE
    TabIndex = 0
    Top = 330
End
Width = 6150

End

Attribute VB_Name = "frmdgPartPlanOpMC"
Attribute VB_GlobalNamingSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gsdatabase As String
Dim gsRecordSource As String

Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdFilter_Click()
    On Error GoTo FilterErr

    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim sFilterStr As String

    If Data1.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Filter a Table Recordset!", 48
        Exit Sub
    End If

    Set recRecordset1 = Data1.Recordset
    'copy the recordset

    sFilterStr = InputBox("Enter Filter Expression:")
    If Len(sFilterStr) = 0 Then Exit Sub

    Screen.MousePointer = vbHourglass
    recRecordset1.Filter = sFilterStr
    Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
    'establish the filter
    Set Data1.Recordset = recRecordset2
    'assign back to original recordset object

    Screen.MousePointer = vbDefault
    Exit Sub

FilterErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
    On Error GoTo RefErr

    RefErr:
Datal.Recordset.Requery

Exit Sub

RefErr:
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdSort_Click()
    On Error GoTo SortErr

    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim SortStr As String

    If Datal.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Sort a Table Recordset!", 48
        Exit Sub
    End If

    Set recRecordset1 = Datal.Recordset
    'copy the recordset

    If Len(msSortCol) = 0 Then
        SortStr = InputBox("Enter Sort Column:")
        If Len(SortStr) = 0 Then Exit Sub
    Else
        SortStr = msSortCol
    End If

    Screen.MousePointer = vbHourglass
    recRecordset1.Sort = SortStr

    'establish the Sort
    Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
    Set Datal.Recordset = recRecordset2

    Screen.MousePointer = vbDefault
    Exit Sub

SortErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
    Dim bParmQry As Integer
    Dim qdfTmp As QueryDef

    gsdatabase = "C:\THESIS_WORK\SOLVER\OA.MDB"
gsRecordsource = "PartPlanOpMC"

    On Error GoTo LoadErr

    'To Do
'gsDatabase is a global string that needs to be set by the startup sub for the app
Datal.DatabaseName = gsdatabase
'gsRecordSource is a global string that needs to be set by the sub routine that loads this form
Datal.RecordSource = gsRecordsource
Datal.RecordsetType = 1 'dynaset
Datal.Options = 0
Datal.Refresh

If Len(Datal.RecordSource) > 50 Then
    Me.Caption = "SQL Statement"
Else
    Me.Caption = Datal.RecordSource
End If

Exit Sub

LoadErr:
    MsgBox "Error:" & Err & " " & Err.Description
    Unload Me
End Sub

Private Sub Form_Resize()
    On Error Resume Next
    If MeWindowState <> 1 Then
        grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
    End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
    If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
    If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
    'let's sort on this column
    If Datal.RecordsetType = vbRSTypeTable Then Exit Sub

    'check for the use of the ctrl key for descending sort
    If mbCtrlKey Then
        msSortCol = "[" & Datal.Recordset(ColIndex).Name & "] desc"
        mbCtrlKey = 0 'reset it
    Else
        msSortCol = "[" & Datal.Recordset(ColIndex).Name & "]"
    End If
End If

cmdSort_Click
msSortCol = vbNullString 'reset it

158
End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer, _
    X As Single, Y As Single)
    mbCtrlKey = Shift
End Sub

Source Code for frmdgParts.frm

VERSION 5.00
Object = "{00028C01-0000-0000-0000-000000000046}" #1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmdgParts
    ClientHeight = 6810
    ClientLeft = 1650
    ClientTop = 1545
    ClientWidth = 7380
    LinkTopic = "Form1"
    ScaleHeight = 6810
    ScaleMode = 0 'User
    ScaleWidth = 7380
Begin VB.PictureBox picButtons
    Align = 1 'Align Top
    Appearance = 0 'Flat
    BorderStyle = 0 'None
    ForeColor = &H80000008&
    Height = 330
    Left = 0
    ScaleHeight = 330
    ScaleWidth = 7380
    TabIndex = 1
    TabStop = 0 'False
    Top = 0
    Width = 7380
Begin VB.CommandButton cmdClose
    Caption = "&Close"
    Height = 330
    Left = 4398
    TabIndex = 5
    Tag = "&Close"
    Top = 0
    Width = 1437
End
Begin VB.CommandButton cmdFilter
    Caption = "&Filter"
    Height = 330
    Left = 2924
    TabIndex = 4
    Tag = "&Filter"
    Top = 0
End
Width            = 1462
End
Begin VB.CommandButton cmdSort
    Caption            = "&Sort"
    Height             = 330
    Left               = 1462
    TabIndex           = 3
    Tag                = "&Sort"
    Top                = 0
    Width              = 1462
End

Begin VB.CommandButton cmdRefresh
    Caption            = "&Refresh"
    Height             = 330
    Left               = 0
    TabIndex           = 2
    Tag                = "&Refresh"
    Top                = 0
    Width              = 1462
End

Begin VB.Data Data1
    Caption            = "Data1"
    Connect            = "Access"
    DatabaseName       = ""
    DefaultCursorType  = 0 'DefaultCursor
    DefaultType        = 2 'UseODBC
    Exclusive          = 0 'False
    Height             = 345
    Left               = 2505
    Options            = 0
    ReadOnly           = 0 'False
    RecordsetType      = 1 'Dynaset
    RecordSource       = ""
    Top                = 2175
    Visible            = 0 'False
    Width              = 1260
End

Begin MSDBGrid.DBGrid grdDataGrid
    Align              = 1 'Align Top
    Bindings           = "frmdgParts.frx":0000
    Height             = 3645
    Left               = 0
    OleObjectBlob      = "frmdgParts.frx":00CE
    TabIndex           = 0
    Top                = 330
    Width              = 7380
End

Attribute VB_Name = "frmdgParts"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim gsdatabase As String
Dim gsRecordsource As String
Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdDelete_Click()

End Sub

Private Sub cmdFilter_Click()
    On Error GoTo FilterErr
    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim sFilterStr As String
    If Data1.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Filter a Table Recordset!", 48
        Exit Sub
    End If
    Set recRecordset1 = Data1.Recordset
    'copy the recordset
    sFilterStr = InputBox("Enter Filter Expression:")
    If Len(sFilterStr) = 0 Then Exit Sub
    Screen.MousePointer = vbHourglass
    recRecordset1.Filter = sFilterStr
    Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
    'establish the filter
    Set Data1.Recordset = recRecordset2
    'assign back to original recordset object
    Screen.MousePointer = vbDefault
    Exit Sub
    FilterErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
    On Error GoTo RefErr
    Data1.Recordset.Requery
    Exit Sub
    RefErr:
    MsgBox "Error:" & Err & " " & Err.Description
End Sub
Private Sub cmdSort_Click()
    On Error GoTo SortErr

    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim SortStr As String

    If Datal.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Sort a Table Recordset!", 48
        Exit Sub
    End If

    Set recRecordset1 = Datal.Recordset
    'copy the recordset

    If Len(msSortCol) = 0 Then
        SortStr = InputBox("Enter Sort Column:")
        If Len(SortStr) = 0 Then Exit Sub
    Else
        SortStr = msSortCol
    End If

    Screen.MousePointer = vbHourglass
    recRecordset1.Sort = SortStr

    'establish the Sort
    Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
    Set Datal.Recordset = recRecordset2

    Screen.MousePointer = vbDefault
    Exit Sub

SortErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
    Dim bParmQry As Integer
    Dim qdfTmp As QueryDef

    gsdatabase = "C:\THESIS_WORK\SOLVER\OA.MDB"
    gsRecordsource = "Parts"

    On Error GoTo LoadErr

    'To Do
    'gsDatabase is a global string that needs 
    'to be set by the startup sub for the app
    Datal.DatabaseName = gsdatabase
    'gsRecordSource is a global string that needs 
    'to be set by the sub routine that loads this form
    Datal.RecordSource = gsRecordsource
    Datal.RecordsetType = 1        'dynaset
Datal.Options = 0
Datal.Refresh

If Len(Datal.RecordSource) > 50 Then
    Me.Caption = "SQL Statement"
Else
    Me.Caption = Datal.RecordSource
End If

Exit Sub

LoadErr:
    MsgBox "Error:" & Err & " " & Err.Description
    Unload Me
End Sub

Private Sub Form_Resize()
    On Error Resume Next
    If Me.WindowState <> 1 Then
        grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
    End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
    If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
    If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
    'let's sort on this column
    If Datal.RecordsetType = vbRSTypeTable Then Exit Sub

    'check for the use of the ctrl key for descending sort
    If mbCtrlKey Then
        msSortCol = "[" & Datal.Recordset(ColIndex).Name & "] desc"
        mbCtrlKey = 0 'reset it
    Else
        msSortCol = "[" & Datal.Recordset(ColIndex).Name & "]"
    End If
    cmdSort_Click
    msSortCol = vbNullString 'reset it
End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer, _
    x As Single, y As Single)
    mbCtrlKey = Shift
End Sub
Source Code for frmdgPlan.frm

VERSION 5.00
Object = "{00028C01-0000-0000-0000-000000000046}"; "DBGRID32.OCX"
Begin VB.Form frmdgPlan
  ClientHeight = 4590
  ClientLeft  = 1650
  ClientTop   = 1545
  ClientWidth = 6150
  LinkTopic   = "Form1"
  ScaleHeight = 4590
  ScaleMode   = 0 'User
  ScaleWidth  = 6150
Begin VB.PictureBox picButtons
  Align      = 1 'Align Top
  Appearance = 0 'Flat
  BorderStyle = 0 'None
  ForeColor  = &H80000008&
  Height     = 330
  Left       = 0
  ScaleHeight = 330
  ScaleWidth = 6150
  TabIndex   = 1
  TabStop    = 0 'False
  Top        = 0
  Width      = 6150
Begin VB.CommandButton cmdClose
  Caption = "&Close"
  Height  = 330
  Left    = 4398
  TabIndex = 5
  Tag     = "&Close"
  Top     = 0
  Width   = 1437
End
Begin VB.CommandButton cmdFilter
  Caption = "&Filter"
  Height  = 330
  Left    = 2924
  TabIndex = 4
  Tag     = "&Filter"
  Top     = 0
  Width   = 1462
End
Begin VB.CommandButton cmdSort
  Caption = "&Sort"
  Height  = 330
  Left    = 1462
  TabIndex = 3
End

Tag = "&Sort"
Top = 0
Width = 1462
End
Begin VB.CommandButton cmdRefresh
Caption = "&Refresh"
Height = 330
Left = 0
TabIndex = 2
Tag = "&Refresh"
Top = 0
Width = 1462
End
End
Begin VB.Data Data1
Caption = "Data1"
Connect = "Access"
DatabaseName = ""
DefaultCursorType = 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 300
Left = 2505
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = ""
Top = 2175
Visible = 0 'False
Width = 1140
End
Begin MSDBGrid.DBGrid grdDataGrid
Align = 1 'Align Top
Bindings = "frmdgPlan.frx":0000
Height = 3645
Left = 0
OleDbObjectBlob = "frmdgPlan.frx":00CE
TabIndex = 0
Top = 330
Width = 6150
End
End
Attribute VB_Name = "frmdgPlan"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gsdatabase As String
Dim gsRecordsource As String

Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
Unload Me

End Sub

Private Sub cmdFilter_Click()
    On Error GoTo FilterErr

    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim sFilterStr As String

    If Datal.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Filter a Table Recordset!", 48
        Exit Sub
    End If

    Set recRecordset1 = Datal.Recordset
    'copy the recordset

    sFilterStr = InputBox("Enter Filter Expression:")
    If Len(sFilterStr) = 0 Then Exit Sub

    Screen.MousePointer = vbHourglass
    recRecordset1.Filter = sFilterStr
    Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
    'establish the filter
    Set Datal.Recordset = recRecordset2
    'assign back to original recordset object

    Screen.MousePointer = vbDefault
    Exit Sub

FilterErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
    On Error GoTo RefErr

    Datal.Recordset.Refresh
    Exit Sub

RefErr:
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdSort_Click()
    On Error GoTo SortErr

    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim SortStr As String

    If Datal.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Sort a Table Recordset!", 48

166
Exit Sub
End If

Set recRecordset1 = Data1.Recordset
'copy the recordset

If Len(msSortCol) = 0 Then
    SortStr = InputBox("Enter Sort Column:")
    If Len(SortStr) = 0 Then Exit Sub
Else
    SortStr = msSortCol
End If

Screen.MousePointer = vbHourglass
recRecordset1.Sort = SortStr

'establish the Sort
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
Set Data1.Recordset = recRecordset2

Screen.MousePointer = vbDefault
Exit Sub

SortErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
    Dim bParmQry As Integer
    Dim qdfTmp As QueryDef

    gsdatabase = "C:\THESIS_WORK\SOLVER\OA.MDB"
    gsRecordsource = "Plans"

    On Error GoTo LoadErr

    'To Do
    'gsDatabase is a global string that needs
    'to be set by the startup sub for the app
    Data1.DatabaseName = gsdatabase
    'gsRecordSource is a global string that needs
    'to be set by the sub routine that loads this form
    Data1.RecordSource = gsRecordsource
    Data1.RecordsetType = 1  'dynaset
    Data1.Options = 0
    Data1.Refresh

    If Len(Data1.RecordSource) > 50 Then
        Me.Caption = "SQL Statement"
    Else
        Me.Caption = Data1.RecordSource
    End If

    Exit Sub
LoadErr:
    MsgBox "Error:" & Err & " " & Err.Description
Unload Me
End Sub

Private Sub Form_Resize()
    On Error Resume Next
    If Me.WindowState <> 1 Then
        grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
    End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
    If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
    If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
        Cancel = True
    End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
    'let's sort on this column
    If Datal.RecordsetType = vbRSTypeTable Then Exit Sub

    'check for the use of the ctrl key for descending sort
    If mbCtrlKey Then
        msSortCol = "[" & Datal.Recordset(ColIndex).Name & "] desc"
        mbCtrlKey = 0 'reset it
    Else
        msSortCol = "[" & Datal.Recordset(ColIndex).Name & "]"
    End If
    cmdSort_Click
    msSortCol = vbNullString 'reset it
End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer, _
    x As Single, y As Single)
    mbCtrlKey = Shift
End Sub

Source Code for frmDgWet.frm
VERSION 5.00
Object = "\{00028C01-0000-0000-0000-000000000046\}#1.0#0"; "DBGRID32.CCX"
Begin VB.Form frmdgWet
  ClientHeight = 4590
  ClientLeft  = 1650
  ClientTop   = 1545
  ClientWidth = 6150
  LinkTopic   = "Form1"
  ScaleHeight = 4590
  ScaleMode   = 0 'User
  ScaleWidth  = 6150
Begin VB.PictureBox picButtons
  Align      = 1 'Align Top
  Appearance = 0 'Flat
  BorderStyle = 0 'None
  ForeColor  = &H80000008&
  Height     = 330
  Left       = 0
  ScaleHeight = 330
  ScaleWidth = 6150
  TabIndex   = 1
  TabStop    = 0 'False
  Top        = 0
  Width      = 6150
Begin VB.CommandButton cmdClose
  Caption   = "&Close"
  Height    = 330
  Left      = 4398
  TabIndex  = 5
  Tag       = "&Close"
  Top       = 0
  Width     = 1437
End
Begin VB.CommandButton cmdFilter
  Caption   = "&Filter"
  Height    = 330
  Left      = 2924
  TabIndex  = 4
  Tag       = "&Filter"
  Top       = 0
  Width     = 1462
End
Begin VB.CommandButton cmdSort
  Caption   = "&Sort"
  Height    = 330
  Left      = 1462
  TabIndex  = 3
  Tag       = "&Sort"
  Top       = 0
  Width     = 1462
End
Begin VB.CommandButton cmdRefresh
  Caption   = "&Refresh"
  Height    = 330
  Left      = 0
  TabIndex  = 2
  Tag       = "&Refresh"
End
Top = 0
Width = 1462
End

Begin VB.Data Data1
Caption = "Data1"
Connect = "Access"
DatabaseName = ""
DefaultCursorType = 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 300
Left = 2505
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = ""
Top = 2175
Visible = 0 'False
Width = 1140
End

Begin MSDBGrid.DBGrid grdDataGrid
Align = 1 'Align Top
Bindings = "frmdgWet.frx":0000
Height = 3645
Left = 0
OleObjectBlob = "frmdgWet.frx":000CE
TabIndex = 0
Top = 330
Width = 6150
End

Attribute VB_Name = "frmdgWet"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gsdatabase As String
Dim gsRecordsource As String

Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdFilter_Click()
On Error GoTo FilterErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim sFilterStr As String

If Data1.RecordsetType = vbRSTypeTable Then
Beep
MsgBox "You Cannot Filter a Table Recordset!", 48
Exit Sub
End If

Set recRecordset1 = Data1.Recordset
'copy the recordset

sFilterStr = InputBox("Enter Filter Expression:"
If Len(sFilterStr) = 0 Then Exit Sub

Screen.MousePointer = vbHourglass
recRecordset1.Filter = sFilterStr
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
'establish the filter
Set Data1.Recordset = recRecordset2
'assign back to original recordset object

Screen.MousePointer = vbDefault
Exit Sub

ErrorHandler:
Screen.MousePointer = vbDefault
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
On Error GoTo RefErr

Data1.Recordset.Requery

Exit Sub

RefErr:
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdSort_Click()
On Error GoTo SortErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim SortStr As String

If Data1.RecordsetType = vbRSTypeTable Then
    Beep
    MsgBox "You Cannot Sort a Table Recordset!", 48
    Exit Sub
Else

    Set recRecordset1 = Data1.Recordset
    'copy the recordset

    If Len(msSortCol) = 0 Then
        SortStr = InputBox("Enter Sort Column:""
        If Len(SortStr) = 0 Then Exit Sub
    Else

    End If
SortStr = msSortCol
End If

Screen.MousePointer = vbHourglass
recRecordset1.Sort = SortStr

' establish the Sort
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
Set Datal.Recordset = recRecordset2

Screen.MousePointer = vbDefault
Exit Sub

SortErr:
Screen.MousePointer = vbDefault
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
Dim bParmQry As Integer
Dim qdfTmp As QueryDef

gsdatabase = "C:\THESIS_WORK\SOLVER\OA.MDB"
gsRecordsource = "W_et"

On Error GoTo LoadErr

'To Do
'gsDatabase is a global string that needs
'to be set by the startup sub for the app
Datal.DatabaseName = gsdatabase
'gsRecordSource is a global string that needs
'to be set by the sub routine that loads this form
Datal.RecordSource = gsRecordsource
Datal.RecordsetType = 1 ' dynaset
Datal.Options = 0
Datal.Refresh

If Len(Datal.RecordSource) > 50 Then
    Me.Caption = "SQL Statement"
Else
    Me.Caption = Datal.RecordSource
End If

Exit Sub

LoadErr:
MsgBox "Error:" & Err & " " & Err.Description
Unload Me
End Sub

Private Sub Form_Resize()
On Error Resume Next
If Me.WindowsState <> 1 Then
    grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
  Cancel = True
End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
  Cancel = True
End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
' let's sort on this column
If Data1.RecordsetType = vbRSTypeTable Then Exit Sub

' check for the use of the ctrl key for descending sort
If mbCtrlKey Then
  msSortCol = "([ & Data1.Recordset(ColIndex).Name & "] desc"
  mbCtrlKey = 0 ' reset it
Else
  msSortCol = "([ & Data1.Recordset(ColIndex).Name & "]"
End If
cmdSort_Click
msSortCol = vbNullString ' reset it
End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer,
  x As Single, y As Single)
  mbCtrlKey = Shift
End Sub

Source Code for frmDGWhhe.frm

' VERSION 5.00
Object = "\00028C01-0000-0000-0000-000000000046\#1.0\#0"; "DBGRID32.OCX"
Begin VB.Form frmDGWhhe
  ClientHeight = 4590
  ClientLeft = 1650
  ClientTop = 1545
  ClientWidth = 6150
  LinkTopic = "Form1"
  ScaleHeight = 4590
  ScaleMode = 0 'User
  ScaleWidth = 6150
Begin VB.PictureBox picButtons
    Align = 1 'Align Top
    Appearance = 0 'Flat
    BorderStyle = 0 'None
    ForeColor = &H80000008&
    Height = 330
    Left = 0
    ScaleHeight = 330
    ScaleWidth = 6150
    TabIndex = 1
    TabStop = 0 'False
    Top = 0
    Width = 6150
End

Begin VB.CommandButton cmdClose
    Caption = "&Close"
    Height = 330
    Left = 4398
    TabIndex = 5
    Tag = "&Close"
    Top = 0
    Width = 1437
End

Begin VB.CommandButton cmdFilter
    Caption = "&Filter"
    Height = 330
    Left = 2924
    TabIndex = 4
    Tag = "&Filter"
    Top = 0
    Width = 1462
End

Begin VB.CommandButton cmdSort
    Caption = "&Sort"
    Height = 330
    Left = 1462
    TabIndex = 3
    Tag = "&Sort"
    Top = 0
    Width = 1462
End

Begin VB.CommandButton cmdRefresh
    Caption = "&Refresh"
    Height = 330
    Left = 0
    TabIndex = 2
    Tag = "&Refresh"
    Top = 0
    Width = 1462
End

End

Begin VB.Data Data1
    Caption = "Data1"
    Connect = "Access"
    DatabaseName = ""
    DefaultCursorType = 0 'DefaultCursor
    DefaultType = 2 'UseODBC
    Exclusive = 0 'False
End
Height = 300
Left = 2505
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = ""
Top = 2175
Visible = 0 'False
Width = 1140

End Begin MSDBGrid.DGGrid grdDataGrid
Align = 1 'Align Top
Bindings = "frmdgWhhe.frx":0000
Height = 3645
Left = 0
OleObjectBlob = "frmdgWhhe.frx":00CE
TabIndex = 0
Top = 330
Width = 6150

End

Attribute VB_Name = "frmdgWhhe"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gsdatabase As String
Dim gsRecordsource As String

Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdFilter_Click()
On Error GoTo FilterErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim sFilterStr As String

If Data1.RecordsetType = vbRSTypeTable Then
Beep
MsgBox "You Cannot Filter a Table Recordset!", 48
Exit Sub
End If

Set recRecordset1 = Data1.Recordset
'copy the recordset
sFilterStr = InputBox("Enter Filter Expression:")
If Len(sFilterStr) = 0 Then Exit Sub
Screen.MousePointer = vbHourglass
recRecordset1.Filter = sFilterStr
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
' establish the filter
Set Data1.Recordset = recRecordset2
' assign back to original recordset object
Screen.MousePointer = vbDefault
Exit Sub
FilterErr:
    Screen.MousePointer = vbDefault
    MsgBox "Error:" & Err & "," & Err.Description
End Sub

Private Sub cmdRefresh_Click()
    On Error GoTo RefErr
    Data1.Recordset.Requery
    Exit Sub
RefErr:
    MsgBox "Error:" & Err & "," & Err.Description
End Sub

Private Sub cmdSort_Click()
    On Error GoTo SortErr
    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim SortStr As String
    If Data1.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Sort a Table Recordset!", 48
        Exit Sub
    End If
    Set recRecordset1 = Data1.Recordset
    ' copy the recordset
    If Len(msSortCol) = 0 Then
        SortStr = InputBox("Enter Sort Column:""
        If Len(SortStr) = 0 Then Exit Sub
    Else
        SortStr = msSortCol
    End If
    Screen.MousePointer = vbHourglass
    recRecordset1.Sort = SortStr
    ' establish the Sort
    Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
    Set Data1.Recordset = recRecordset2
    Screen.MousePointer = vbDefault
Exit Sub
SortErr:
  Screen.MousePointer = vbDefault
  MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
  Dim bParmQry As Integer
  Dim qdfTmp As QueryDef

  gsdatabase = "C:\THESIS_WORK\SOLVER\OA.MDB"
  gsRecordsource = "W_hhe-Time"

  On Error GoTo LoadErr
    'To Do
    'gsDatabase is a global string that needs
    'to be set by the startup sub for the app
    Datal.DatabaseName = gsdatabase
    'gsRecordSource is a global string that needs
    'to be set by the sub routine that loads this form
    Datal.RecordSource = gsRecordsource
    Datal.RecordsetType = 1   'dynaset
    Datal.Options = 0
    Datal.Refresh

    If Len(Datal.RecordSource) > 50 Then
      Me.Caption = "SQL Statement"
    Else
      Me.Caption = Datal.RecordSource
    End If
  Exit Sub

LoadErr:
  MsgBox "Error:" & Err & " " & Err.Description
  Unload Me
End Sub

Private Sub Form_Resize()
  On Error Resume Next
  If Me.WindowState <> 1 Then
    grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
  End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
  If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
    Cancel = True
  End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
  If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
    ' let's sort this column
    If Data1.RecordsetType = vbRSTypeTable Then Exit Sub

    ' check for the use of the ctrl key for descending sort
    If mbCtrlKey Then
        msSortCol = "[" & Data1.Recordset(ColIndex).Name & "] desc"
        mbCtrlKey = 0 ' reset it
    Else
        msSortCol = "[" & Data1.Recordset(ColIndex).Name & "]"
    End If
    cmdSort_Click
    msSortCol = vbNullString ' reset it
End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer, _
    x As Single, y As Single)
    mbCtrlKey = Shift
End Sub

Source Code for frmdgWit.frm

VERSION 5.00
Object = "{00028C01-0000-0000-0000-000000000046}#1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmdgWit
    ClientHeight = 4590
    ClientLeft = 1650
    ClientTop = 1545
    ClientWidth = 6150
    LinkTopic = "Form1"
    ScaleHeight = 4590
    ScaleMode = 0 ' User
    ScaleWidth = 6150
Begin VB.PictureBox picButtons
    Align = 1 ' Align Top
    Appearance = 0 ' Flat
    BorderStyle = 0 ' None
    ForeColor = &H80000008&
    Height = 330
    Left = 0
    ScaleHeight = 330
    ScaleWidth = 6150
    TabIndex = 1
    TabStop = 0 ' False
Top = 0
Width = 6150

Begin VB.CommandButton cmdClose
    Caption = "&Close"
    Height = 330
    Left = 4398
    TabIndex = 5
    Tag = "&Close"
    Top = 0
    Width = 1437
End

Begin VB.CommandButton cmdFilter
    Caption = "&Filter"
    Height = 330
    Left = 2924
    TabIndex = 4
    Tag = "&Filter"
    Top = 0
    Width = 1462
End

Begin VB.CommandButton cmdSort
    Caption = "&Sort"
    Height = 330
    Left = 1462
    TabIndex = 3
    Tag = "&Sort"
    Top = 0
    Width = 1462
End

Begin VB.CommandButton cmdRefresh
    Caption = "&Refresh"
    Height = 330
    Left = 0
    TabIndex = 2
    Tag = "&Refresh"
    Top = 0
    Width = 1462
End

End

Begin VB.Data Data1
    Caption = "Data1"
    Connect = "Access"
    DatabaseName = ""
    DefaultCursorType = 0 'DefaultCursor
    DefaultType = 2 'UseODBC
    Exclusive = 0 'False
    Height = 300
    Left = 2505
    Options = 0
    ReadOnly = 0 'False
    RecordsetType = 1 'Dynaset
    RecordSource = ""
    Top = 2175
    Visible = 0 'False
    Width = 1140
End

Begin MSDBGrid.DBGrid grdDataGrid
Align = 1 'Align Top
Bindings = "frmdgWit.frx":0000
Height = 3645
Left = 0
OleObjectBlob = "frmdgWit.frx":00CE
TabIndex = 0
Top = 330
Width = 6150

End
End
Attribute VB_Name = "frmdgWit"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gsdatabase As String
Dim gsRecordsresource As String

Dim msSortCol As String
Dim mbCtrlKey As Integer

Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdFilter_Click()
    On Error GoTo FilterErr
    Dim recRecordset1 As Recordset, recRecordset2 As Recordset
    Dim sFilterStr As String
    If Data1.RecordsetType = vbRSTypeTable Then
        Beep
        MsgBox "You Cannot Filter a Table Recordset!", 48
        Exit Sub
    End If
    Set recRecordset1 = Data1.Recordset
    'copy the recordset
    sFilterStr = InputBox("Enter Filter Expression:")
    If Len(sFilterStr) = 0 Then Exit Sub
    Screen.MousePointer = vbHourglass
    recRecordset1.Filter = sFilterStr
    Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
    'establish the filter
    Set Data1.Recordset = recRecordset2
    'assign back to original recordset object
    Screen.MousePointer = vbDefault
    Exit Sub

FilterErr:
Screen.MousePointer = vbDefault
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdRefresh_Click()
On Error GoTo RefErr
Data1.Recordset.Requery
Exit Sub

RefErr:
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub cmdSort_Click()
On Error GoTo SortErr

Dim recRecordset1 As Recordset, recRecordset2 As Recordset
Dim SortStr As String

If Data1.RecordsetType = vbRSTypeTable Then
    Beep
    MsgBox "You Cannot Sort a Table Recordset!", 48
    Exit Sub
End If

Set recRecordset1 = Data1.Recordset
'copy the recordset

If Len(msSortCol) = 0 Then
    SortStr = InputBox("Enter Sort Column:")
    If Len(SortStr) = 0 Then Exit Sub
Else
    SortStr = msSortCol
End If

Screen.MousePointer = vbHourglass
recRecordset1.Sort = SortStr

'establish the Sort
Set recRecordset2 = recRecordset1.OpenRecordset(recRecordset1.Type)
Set Data1.Recordset = recRecordset2

Screen.MousePointer = vbDefault
Exit Sub

SortErr:
Screen.MousePointer = vbDefault
MsgBox "Error:" & Err & " " & Err.Description
End Sub

Private Sub Form_Load()
Dim bParmQry As Integer
Dim qdfTmp As QueryDef
gsdatabase = "C:\THEESIS_WORK\SOLVER\OA.MDB"
gsRecordSource = "W_it"

On Error GoTo LoadErr

'To Do
'gsDatabase is a global string that needs
'to be set by the startup sub for the app
Datal.DatabaseName = gsdatabase
'gsRecordSource is a global string that needs
'to be set by the sub routine that loads this form
Datal.RecordSource = gsRecordSource
Datal.RecordsetType = 1 'diraset
Datal.Options = 0
Datal.Refresh

If Len(Datal.RecordSource) > 50 Then
  Me.Caption = "SQL Statement"
Else
  Me.Caption = Datal.RecordSource
End If

Exit Sub

LoadErr:
  MsgBox "Error:" & Err & ";" & Err.Description
  Unload Me
End Sub

Private Sub Form_Resize()
  On Error Resume Next
  If Me.WindowsState <> 1 Then
    grdDataGrid.Height = Me.Height - (425 + picButtons.Height)
  End If
End Sub

Private Sub grdDataGrid_BeforeDelete(Cancel As Integer)
  If MsgBox("Delete Current Row?", vbYesNo + vbQuestion) <> vbYes Then
    Cancel = True
  End If
End Sub

Private Sub grdDataGrid_BeforeUpdate(Cancel As Integer)
  If MsgBox("Commit changes?", vbYesNo + vbQuestion) <> vbYes Then
    Cancel = True
  End If
End Sub

Private Sub grdDataGrid_HeadClick(ByVal ColIndex As Integer)
  'let's sort on this column
  If Datal.RecordsetType = vbRSTypeTable Then Exit Sub

  'check for the use of the ctrl key for descending sort
  If mbCtrlKey Then
    msSortCol = "[" & Datal.Recordset(ColIndex).Name & "] desc"
  Else
    msSortCol = "[" & Datal.Recordset(ColIndex).Name & "] asc"
  End If

  Datal.Recordset.Close
  Datal.Refresh

  Exit Sub
End Sub
mbCtrl1Key = 0 'reset it
Else
    msSortCol = "[" & Data1.Recordset(ColIndex).Name & "]"
End If
cmdSort_Click
msSortCol = vbCrLf 'reset it

End Sub

Private Sub grdDataGrid_MouseUp(Button As Integer, Shift As Integer, _
    x As Single, y As Single)
    mbCtrl1Key = Shift
End Sub
VITA AUCTORIS

NAME: Jorge Paulo

PLACE OF BIRTH: Vila do Conde, Portugal

YEAR OF BIRTH: 1974

EDUCATION:
Kingsville District High School, Kingsville, Ontario 1990-1994
