Fuzzy Multi-Objective Linear Programming and Simulation Approach to the Development of Valid and Realistic Master Production Schedule

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Maschinenbau und Verfahrenstechnik
der
Universität Duisburg-Essen
zur Erlangung des akademischen Grades

eines

Doktors der Ingenieurwissenschaften

Dr. –Ing.

genehmigte Dissertation

von

Supriyanto Supriyanto
aus
Pati, Indonesien

Gutachter:

Univ.-Prof. Dr.-Ing. Bernd Noche
Univ.-Prof. Dr.-Ing. Uwe Bracht

Tag der mündlichen Prüfung: 21.10.2011

Abstract

Master production schedule (MPS) plays an important role in an integrated production planning system. It converts the strategic planning defined in a production plan into the tactical operation execution. The MPS is also known as a tool for top management to control over manufacture resources and becomes input of the downstream planning levels such as material requirement planning (MRP) and capacity requirement planning (CRP). Hence, inappropriate decision on the MPS development may lead to infeasible execution, which ultimately causes poor delivery performance. One must ensure that the proposed MPS is valid and realistic for implementation before it is released to real manufacturing system. In practice, where production environment is stochastic in nature, the development of MPS is no longer simple task. The varying processing time, random event such as machine failure is just some of the underlying causes of uncertainty that may be hardly addressed at planning stage so that in the end the valid and realistic MPS is tough to be realized. The MPS creation problem becomes even more sophisticated as decision makers try to consider multi-objectives; minimizing inventory, maximizing customer satisfaction, and maximizing resource utilization. This study attempts to propose a methodology for MPS creation which is able to deal with those obstacles. This approach takes into account uncertainty and makes trade off among conflicting multi-objectives at the same time. It incorporates fuzzy multi-objective linear programming (FMOLP) and discrete event simulation (DES) for MPS development.

Acknowledgment

In the name of Allah, the Most Gracious, the Most Merciful.

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Dr.-Ing. Bernd Noche, for the inspirational discussions and guidance throughout this work.

Special thanks to all academic and technical staff of Transport System and Logistic Institute for their many helpful suggestions and administrative support.

Another special gratitude owes to my parents, who pray for me anytime and from whom I always get psychological strength during my study. Their lovely care and endless patience enables me to finish this dissertation. Also I wish to acknowledge my gratitude to my family who always support me.

At last but not least, I would like to thank all friends with whom I spent precious time during in Germany. Their kind friendship gives me confidence for bright future and makes my life meaningful.

Contents

List of Tables

List of Figures

Acronyms

API	Application Programming Interface
APICS	American Production and Inventory Control Society
APP	Aggregate Production Planning
ATO	Assembly To Order
ATP	Available To Promise
BOM	Bill Of Material
BPMI	Business Process Management Initiative
BPMN	Business Process Modeling Notation
BSS	Below Safety Stock
CRP	Capacity Requirement Planning
DE	Differential Evolution
DES	Discrete Event Simulation
DM	Decision Maker
DIN	Deutsches Institut für Normung e.V.
EC	Evolutionary Computation
EI	Ending Inventory
ERP	Enterprise Resource Planning
ETPSP	Earliness/Tardiness Production Scheduling and Planning
FMOLP	Fuzzy Multi-Objective Linear Programming
FMS	Flexible Manufacturing System
GA	Genetic Algorithm
LP	Linear Programming
MADM	Multi-Attribute Decision Making
MM	Mixed Mode
MPC	Manufacturing Planning and Control
MPS	Master Production Scheduling
MRP	Material Requirement Planning
MTO	Make To Order
MTS	Make To Stock
OC	Over Capacity (Overtime)
OECD	Organization for Economic Co-operation and Development

PO	Production Order
POR	Planned Order Release
PPC	Production Planning and Control
PPL	Production Planning Logic
PP-SAP	Production Planning – SAP
RCCP	Rough-Cut Capacity Planning
RNM	Requirement Not Met
SKU	Stock Keeping Unit
SOP	Sales and Operation Planning
SR	Schedule Receipt
WC	Work Center
WIP	Work In Process

List of Variables

X	Variable of universal set
x	Independent variable of a arbitrary function
$\mu_z(x), \mu_D(x), \dots$	Membership grade of variable x at fuzzy set *Z, D, …*
α	Level set of membership degree
$\tilde{Z}, \widetilde{D}, \dots$	Fuzzy set of variable *Z, D, …*
Z_α	Membership grade of fuzzy set Z which exceed level set α
$f(x)$	Arbitrary mathematical function
λ	Auxiliary variable for fuzzy objectives
$Z_i(x)$	Objective function i^{th} with independent variable x
$\tilde{Z}_i(x)$	Fuzzy objective Z_i with independent variable x
$\tilde{g}_r(x)$	Fuzzy constraint g_r with independent variable x
w_i	Weighting coefficient of fuzzy objectives i^{th}
β_r	Weighting coefficient of fuzzy objectives r^{th}
γ_r	Auxiliary variable for fuzzy constraint r^{th}
P_m	Mutation Probability Parameter
P_c	Crossover Probability Parameter
K	Total quantity of different products (SKU)
R	Total quantity of different productive resources
p	Total number of planning periods
TH	Total planning horizon
OH_k	Initial available inventory (on-hand), at the first scheduling period
GR_{kp}	Gross requirement for product k at period p
BS_{kp}	Standard lot size for product k at period p
NR_{kp}	Net requirement for product k at period p, considering infinity capacity
SS_{kp}	Safety inventory level for product k at period p
UR_{kr}	Production rate for product k at resource r (units per hour)
AC_{rp}	Available capacity, in hours, at resource r at period p
BN_{kpr}	Quantity of standard lot sizes needed for the production of the product k at resource r, at period p (number of lots)
MPS_{kpr}	Total quantity to be manufactured of the product k at resource r, at period p
$MPST_{kp}$	Total quantity to be manufactured of the product k at resource r, at period p

	(considering all available resources)
BI_{kp}	Initial inventory level of the product k at period p
CUH_{rp}	Capacity used from the resource r at period p
CUP_{rp}	Percent rate obtained from the relation of the number of hours consumed from the resource r at period p, and the available number of hours to the same resource and period
GR_{kp}	Gross requirement for product k at period p
RM_{kp}	Total requirements met for product k at period p
RM_{kpr}	Total requirements met for product k at period p at resource r
RNM_{kp}	Requirements not met for product k at period p
SL_{kp}	Service level, relation of the requirement met, RM_{kp} and the gross requirements for product k at period p
$Z_1\ (EI)$	Ending inventory at period p
$Z_2\ (RNM)$	Average requirement not met
$Z_3\ (BSS)$	Average quantity below safety stock level
$Z_4\ (OC)$	Average over capacity needed (overtime)
OL_{max}	Maximum overtime allowed in one period
Z_i^0	Lower bound of aspiration level of objective Z_i
Z_i^1	Upper bound of aspiration level of objective Z_i
A	Lower tolerance violation for overtime
B	Upper tolerance violation for overtime
R_o	Over-loaded resource (overtime)
R_u	Under-loaded resource
RT	Remaining load capacity at resource R_o after leveling
DI	Quantity of potential defect product manufactured at resource R_u
y_k	Quantity of item k (multiple of lot size) to be shifted from R_o to R_u
x_{ko}	Planned quantity of item k to be manufactured at resource R_o
AC_o	Actual load capacity at resource R_o (hrs)
RC_u	Available capacity at resource R_u (hrs)
U_{ko}	Average of production rate at resource R_o to produce item k (SKU/hrs)
U_{ku}	Average of production rate at resource R_u to produce item k (SKU/hrs)
s_{ku}	Defect rate of product k at resource R_u
g	Constraint to ensures that the R_u does not turn into overloaded after leveling

Chapter 1

Preface and Overview

1.1 Motivation

The key success of a manufacturing company to be competitive in recent globalization is to satisfy customer's requirement on time with good quality of products and services. Obviously, this will be a challenging mission for decision makers particularly in today's highly dynamic market environment which is characterized by short product life-cycle, high product variety (customization), unpredictable demand pattern and short customer lead times. The gaining competitive advantage is being more difficult due to the uncertainty of the market environment. It's required smooth communication and effective coordination between marketing/sales and manufacturing department to ensure that the company resources can cope the varying customer demand. Master Production Schedule (MPS) is a function in Production Management System which provides mechanism for the dynamic interaction between marketing and manufacturing (Higgins & Browne, 1992).

According to American Production and Inventory Control Society (APICS), an MPS is declaration of what the company expects to manufacture that drives the material requirement planning (MRP). It represents what the company intends to produce expressed in term of configuration, quantities, and specific dates.

Since the master production schedule is used to drive all the subsequent activities of a manufacturing company, it's necessary to ensure that the MPS is valid. That is one which the material due dates equal to the material need dates and the planned capacity equal to the required capacity. Failure to create valid MPS results in the company's manufacturing and supplier resources being poorly deployed (Proud, 1999). This means that the company may be unresponsive to customer needs or wasteful in its use of resource and ultimately the company risks losing its competitive position.

The development of MPS is a very complex problem, especially in manufacturing environment with limited capacity, multiple products, and multiple productive resources (*flexible routing*). Within a restriction scenario, based on demand forecast, the company should be accurately defined what is going to be made. Thus, when more than one productive resource can be picked, the company should determine what is the most adequate one to use? Which products should be assigned to the resource? And what is the best assignment of product quantities to the resources? Moreover, the company must ensure that conflicting objectives such as minimizing inventory level, maximizing customer service level and the use of production resources can be compromised. *Garey & Johnson* (1979) proved mathematically that the productive planning problem is NP-hard, particularly, as capacity and setup restrictions are considered.

Proud (1999) emphasized that the real challenge of MPS creation is to balance supply of resources (available capacity) with the demand for those resources (requirement capacity). In this context, the viable solution may be easily found for given deterministic capacity, however, it turns into intricate problem if both available and requirement capacity are stochastic or fuzzy. It is a fact that the available capacity of productive resource cannot be determined precisely due to unpredictable event such as machine failure, labor shortage, material shortage, or unplanned breakdown. The requirement capacity for producing demand is also fuzzy because of varying processing time, setup time, or queue time.

In the real practice, it seems that this phenomenon is not taken into account by the modern ERP system. Rough-cut capacity planning (RCCP) function, which provides the productive resource profile for MPS, is not designed to deal with uncertainty. Rough-cut capacity (RCCP) even considers only key or critical resources. Actual building of a product, however, requires the resources in all work centers including their stochastic situation if necessary. It is in this sense that rough cut capacity planning is limited to answering the question: Do we have a chance to meet the production plan? And do we have a chance to meet the master production schedule? (Proud, 1999).

Some literatures argue that the detail overview of resource profile is not necessary in this phase because the lower level of capacity planning will be carried out by capacity requirement planning (CRP) function after MRP processing. Indeed, CRP provides a more detailed capacity check on MRP-generated production plans than RCCP. It considers all planned order release, existing work-in-process (WIP) positions, routing data, as well as capacity and lead times for all process centers. However, CRP persist the underlying assumption which is neglecting uncertainty. In addition, CRP performs only what is called *infinite forward loading* instead of finite capacity analysis. It means that CRP is incapable to make correction if the overloaded on a process center occurs (Hopp & Spearman, 2008).

Likewise, the predominant optimization techniques such as genetic algorithm (Soares & Vieira, 2008), heuristic (Vieira G. E., 2004), and simulated annealing (Vieira & Ribas, 2004), which are often employed for solving the MPS problem in research fields, also fail to capture the uncertainty phenomenon. As a result, the appropriate balancing of available capacity and requirement capacity cannot be ensured.

This fact underlines presumption that the present MPS system has obstacle to estimate precisely both of the requirement capacity and the available capacity, particularly, in fuzzy environment. In such situation, a procedure or mechanism to verify the feasibility of proposed master schedule must be developed. *Heizer & Render* (2006) proposed a concept of close-loop planning process that allow feedback to the next higher level planning (either MRP or MPS) and make necessary adjustment when the plan is infeasible to implement. However, they did not go detail how to evaluate the feasibility. *Kochhar et. al* (1998) have developed more concrete approach with a knowledge-based system, which represents the behavior and rules of the manufacturing system, to test the feasibility of tentative master production schedule.

Differs from the existing researches, this study attempts to develop valid and realistic MPS through two approaches; (i) to employ fuzzy multi-objective linear programming. Using the fuzzy model a master scheduler can express the fuzzy objective and constraint into mathematical model. (ii) to develop new closed-loop MPS that allows a master scheduler to verify the feasibility of master schedules. The discrete event simulation (DES) environment will be used to help with testing and finalization of master schedules.

1.2 Literature Review

This section presents the review of existing work in respect of the master production schedule. To provide thorough outlook, the literature review is organized into two sub-sections. The first sub-section (1.2.1) focuses on the works studying the optimization techniques or method employed in MPS creation. And, second sub-section (1.2.2) presents all works relating to the system and procedures applied to develop a valid and realistic MPS. The enterprise resource planning (ERP) is named as well in this sub-section since nowadays master production schedule constitutes one of modules in ERP's system. In addition, it is meant to show the limitation of ERP which may inherits to MPS.

1.2.1 Overview of MPS Development Techniques

Master production scheduling (MPS) is a one of function in the integrated production planning and control, which translate strategic goals from business planning into an anticipated statement of production, from which all other schedules at lower level are derived (Higgins & Browne, 1992). The MPS is also a key interface between marketing and manufacturing since it links customer service directly to efficient use of production resource (*Vollmann et al., 1997*). Mismanagement of the MPS is considered as one of fundamental problem in operation and it can potentially lead to poor customer satisfaction (Plossl & Lankford, 1984). Therefore, Slack *et al.* (2001) regarded MPS as the most important planning and control schedule in a business.

The development of MPS is managed differently depending on the manufacturing environment. In a make-to-stock (MTS) environment, the MPS parts can be leveled following the demand. However, in an assembly-to-order (ATO) and make-to-order (MTO), since the configuration of the final product is not known until the time a customer places the order, leveling the MPS is not viable option. To minimize the impact of demand variations in MTO environment, the inventory of raw material can be build. *McClelland* (1987) provides guidelines for the selection of appropriate master production schedules method for MTO environment. In contrary, in ATO firm, the demand variation is tackled by storing assemblies and subsets instead. *Vollmann* (1997) considers the creation of master plans in an ATO environment particularly challenging and thus, he proposes a method called two-level master production schedules. *King & Benton* (1988) was comparing two techniques, that are superbills and covering sets, in ATO environment with goal minimizing delivery time pressure, while *Tallon* (1988) made a study comparing four techniques in the same environment.

Up till now researchers have improved sustainably both mathematical and programming approaches for solving MPS problem in the various production systems as well as distinctive objectives and constraints. *Hill et al.* (2000) implemented a method called two-level master production schedule in process industry and by incorporating a scheduling heuristic which considers sequence-dependent changeovers and capacity constraints. *Chung & Lee* (1989) published a paper which investigated the MPS problems for Flexible Manufacturing System (FMS). This paper proposed a heuristic algorithm to develop an MPS for FMS used in make-to-stock factories. *Arosio & Sianesi* (1993) also proposed a complete heuristic algorithm, which solves the problem lot-sizing and sequencing in a single logic stage, to optimize master production schedules. Moreover, a practical heuristic algorithm, which considers the most common parameters and decision variables included in MPS, was proposed by *Vieira* (2004). This study also provides a detail example illustrating the complex task of MPS creation. *Freedendall & Lea* (1997) improved the product mix heuristic to create the optimum MPS which is capable of maximizing system throughput.

Chu (1995) employed optimization model to obtain "optimal" or "near-optimal" MPS. This study developed a linear programming model which maximizes profit subject to supply, demand and labor resource. *Lin & Krajewski* (1992) developed mathematical model for MPS in uncertain environment without capacity constraints, where the MPS is developed using a rolling schedule. The focus of this study is to determine appropriate re-planning interval (R), frozen interval (F), and forecast window (T) and the effect of those parameters on system cost. Thus, *Gonzalez & Reeves* (1983) have described and discussed a mathematical model for developing master production schedules which considers multiple objectives: minimize the total cost, minimize inventory level, minimize production time and minimize the underutilization of line.

In general, master scheduling problems are NP-hard problems, that is, unlikely there is no an algorithm that can find an optimal MPS solution in polynomial time. In practical, this means that the processing time required to solve such problem will quickly and enormously grow as the problem size grows. For this reason, truly optimal solution is quite difficult to be found. Therefore, meta-heuristic or artificial intelligence techniques such as genetic algorithm, simulated annealing, tabu search, etc. are also employed to obtain optimum solution.

Wu et al. (2002) built a mathematical model of MPS and developed a genetic algorithm incorporating several techniques to satisfy constraints for making an optimized MPS of production lines with both assembly and processing. *Keung et al.* (2001) adopted an earliness/tardiness production scheduling and planning (ETPSP) approach to generate accurate and realistic MPS in flexible manufacturing system. The ETPSP problem considers machine capacity, incorporating batch-size consideration for multiple part type production and sequential machine processing for FMS using the genetic algorithm (GA) approach. *Vieira & Ribas* (2004) applied simulated annealing to solve production planning problem, more specially, master production schedule. This study reveals some drawbacks of simulated annealing such as overcoming local optimum. *Soares & Vieira* (2008) introduced new genetic algorithm structure for solving MPS problem. This study formulates the fitness function which aims to minimize inventory level, maximize service level, minimize overtime and minimize inventory level below safety stock. At the end, *Vieira et. al* (2003) has compared genetic algorithms and simulated annealing for master production scheduling problems.

In general, the reviewed approaches have given worthy contribution to the development of MPS, specially, to balance requirement capacity, which is derived from customer demand, with available resource in effective manner while maintaining trade-off among the conflicting objectives. Only one thing needs to be criticized that is the underlying assumption of those approaches which ignore the capacity uncertainty. To bridge the drawback, this research proposes a method called *Fuzzy multi-objective linear programming* (FMOLP) that allows a master scheduler to consider the fuzzy objective and resource.

1.2.2 Overview of MPS System

MRP has been described as a system that consists of a set of logically related procedures, decision rules, and records designed to translate a master production schedules (MPS) and the planned coverage of each requirement, for each component inventory item needed to implement this schedule (Orlicky & Plossl, 1994).

In the 1975, MRP was expanded to MRP II, which refers to manufacturing resource planning (Gray & Landvater, 1989). MRP II integrates the information and manufacturing technology, plans and resources to improve efficiency of a manufacturing enterprise. Thus, in the 1990s, MRP II has evolved further to Enterprise Resource Planning (ERP) with enhanced and added functionality (Gumaer, 1996), (Yusuf & Little, 1998). The key difference between MRP II and ERP is that while MRP II has traditionally focused on the planning and scheduling of internal resources, ERP strives to plan and schedule supplier resources as well, based on customer demand and schedules (Chen I. J., 2001).

In spite of its evolution success, the ERP is not capable of eliminating the underlying assumption of its predecessor (MRP and MRP II) that is unable to handle stochastic situation or uncertainty (Moon & Phatak, 2005), which is defined as unpredictable events occurring during production that cannot be planned such as material shortage, faulty components and machine breakdowns. By mean of the questionnaire survey and telephone interviews to ERP’s users, *Koh & Simpson* (2005) has also proved that the ERP system has incapability to cope uncertainty due to its stochastic and unpredictable nature. They found that the underlying causes of uncertainty significantly affects the product late delivery performance in make-to-stock (MTS), make-to-order (MTO), and mixed mode (MM) manufacturing environment in small medium enterprise using ERP. In addition, *Koh et al.* (2006) have also reported that the uncertainty has caused tardy delivery performance of company in most ERP-controlled manufacturing environment in UK and China.

A lot of studies have examined the various buffering and dampening techniques to tackle the effect of uncertainties (*Koh et al.,* 2006). However, almost all of the techniques behave reactive instead of preventive. Those techniques can only be applied after the underlying cause of uncertainty takes places. Only few studies which attempt to diminish the effect of uncertainty before it happens, for example, at the production planning development phase. Moon & Phatak (2005) was one of few researchers who developed a production planning by considering uncertainty. They model the predicted uncertainty and then based on it, the realistic planned lead time is determined. *Koh et al.* (2005) have also published the similar framework. A simulation is carried out to examine the effects of underlying causes of uncertainty on tardy delivery in an ERP-controlled manufacturing environment that was modelled without the application of buffering or dampening techniques. Thus, a knowledge management approach was developed to recommend explicit and tacit knowledge in tackling uncertainty in such environment. Both studies used ARENA as simulation environment.

These review has shown that the assumption of ERP's system in ignoring stochastic nature has caused the production planning inaccuray and ultimately lead to poor performance of the ERP-controlled manufacturing system. Therefore, it is necessary to make verification on the production planning generated by ERP including master production schedule (MPS).

Concerning to the verification of master production schedule, *Higgins & Browne* (1992) have proposed a simulation of "What-if" analysis to be executed on the tentative MPS in order to find an optimum and realistic MPS. *Heizer & Render* (2006) shared an idea of what called „close-loop planning process" which allows a planner to check the validity of each planning process. *Kochhar, Ma, & Khan* (1998) have developed a knowledge-based system approach, which is combining human expertise with computer computation, to achieve an accurate and realistic master production schedule. This paper claimed that the resulting system can determine the load imposed by tentative MPS on the available resources, identity the bottlenecks, and suggest solution for resolving the problem as well as perform what-if analysis to verify the effectiveness of solution.

To this end, this research proposes "Closed Loop MPS" which utilizes discrete event simulation (DES) to verify the validity and realism of the tentative MPS. Principally, the verification is carried out by simulating the execution of MPS in shop-floor level. The MPS is considered as feasible if the planned orders, which are generated based on the MPS, can be completed on time without excessive unplanned overtime.

1.3 Goal of the Thesis

The foremost goal of this research is to develop the ***valid*** and ***realistic*** MPS subject to fuzzy objective and resources. The term "valid" refers to a MPS which the material due dates equal to the material need dates and the planned capacity equal to the required capacity (Proud, 1999). The term "realistic" means that the performance measures, which are inventory level (EI), requirement not met (RNM), inventory below safety stock (BSS) and overtime (OC), is equal or close to the decision maker's target. In this research, fuzzy multi-objective linear programming and simulation approach are employed to create a valid and realistic MPS.

In order to achieve the primary goal, the following milestones must be completed sequentially.

- To develop the mathematical model of MPS problem[1]
- To formulize the fuzzy multi-objective linear programming for MPS problem and subsequently de-fuzzifies the model into crisp solution
- To solve the crisp formulation using MATLAB's genetic algorithms
- To develop the closed-loop MPS process, that is a conceptual diagram to describe how the generation of valid and realistic MPS
- To design and develop an information system that enables the proposed closed-loop MPS to be implemented. This milestone can be divided into three phases; system design, system integration and coding development
- To generate the production planning in SAP for a manufacturing system
- To develop the simulation model of manufacturing system using DOSIMIS-3. This milestone consists of four steps; problem formulation, development, validation, and testing
- To develop the standard procedure (SOP) how to create MPS using the closed-loop MPS and eventually perform numerical example

1.4 Research Methodology

This thesis employs the following research methods: *literature review*, *mathematical modeling*, *stochastic programming*, *system design & development*, and *simulation*.

Literature Review

The literature review (Chapter 1) reveals that existing works does not yet take into account the capacity uncertainty. In addition, the review has also given an idea about the MPS validation.

[1] Refers to the mathematical model used by *Soares et al.* (2008)

Another extensive literature review (Chapter 2) provides thorough overview of theoretical framework encompassing the fundamental theory of MPS, the role of MPS in manufacturing resource planning (MRP II), fuzzy modeling and genetic algorithm.

Mathematical Modeling

To obtain a solution, one has to express the MPS problem into fuzzy multi-objective linear programming (FMOLP) and subsequently de-fuzzifies the FMOLP model into a crisp linear programming (Chapter 3).

Stochastic Programming

In chapter 4, the crisp single objective linear programming equivalent to FMOLP is solved by using genetic algorithm (GA), which is essentially stochastic programming. New genetic structure of MPS problem proposed by *Soares et al* (2008) is adopted in this study. In addition to GA, the differential evolution algorithm is also developed in order to provide a benchmarking.

System Design and Development

A platform or information system, which enables the proposed closed-loop MPS to be implemented, is designed and developed. The information system is made of three software packages which are integrated; (1) MATLAB for optimization (2) DOSIMIS-3 for simulation (3) SAP for executing production planning logic.

Simulation

The planning execution in shop-floor is simulated using DOSIMIS-3. The stochastic situation (disturbance) such as machine failure and varying processing time is also modeled. The planning discrepancy due to the disturbance is quantified by the performance measures. The MPS is considered valid and realistic if the performance measures close or equal to the DM’s target and the material due date is equal to the material need date.

1.5 Organization of the Thesis

In principle, to develop a valid and realistic master production schedules, two approaches have been proposed: (i) to improve optimization technique (ii) and to verify the tentative MPS, which is realized by the designing of new MPS framework. Step-by-step implementation of those approaches will be discussed in this thesis. Theoretical framework providing prerequisite knowledge of both main topics (optimization technique and MPS framework) is described in chapter 2. The implementation of optimization technique is explained more detail in chapter 3 & 4, while the development of the MPS framework is discussed comprehensively in chapter 5 & 6.

Chapter 7 combined both approaches to create a valid and realistic MPS. Figure 1 illustrates the structure of this thesis. The detail content of each chapter is described as follows.

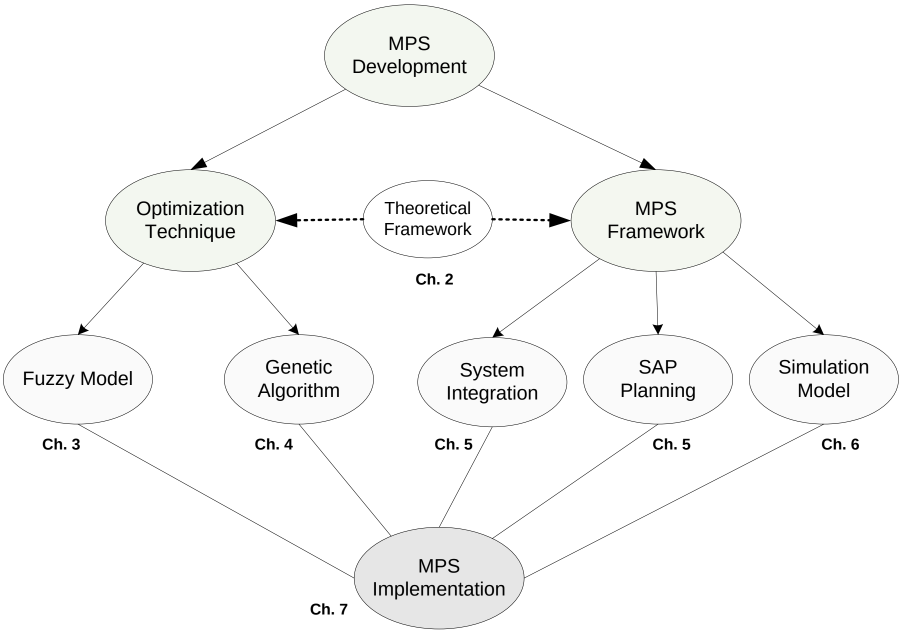


Figure 1 The structure of this thesis

Chapter 1 starts with describing briefly the motivation of thesis and the unaddressed problem in master production schedule. Then, the extensive literature review is presented to survey the existing works that are similar or related to this topic. In addition, this chapter states clearly both the primary goal of thesis and the employed research methodology. In the end, the organization of thesis, which resumes the contents of each chapter, is documented.

Chapter 2 summarizes the cross disciplinary overview of definition, theory and algorithm which is being the prerequisite knowledge of this thesis. This chapter begins by explaining the fundamental theory of master production schedules and the scheme of MRP II. The scheme shows the general framework of MPS development in an integrated production planning system. This chapter is also devoted to the basic terminology and concept of fuzzy theory, especially, on decision making. It reviews the formulation of FMOLP, and how to transform it into crisp solution. It’s very worthwhile for one of whom no knowledge of fuzzy theory. Thus, the fundamental theory of GA is presented, however, it covers only the core operation of genetic algorithms which are selection, crossover, mutation, and search termination.

Chapter 3 discusses firstly about the mathematical model of MPS problem. Thus, it explains procedures how to build the fuzzy multi-objective linear programming (FMOLP) for MPS problem. A study case is also given to provide numerical example. Using the same study case, some evaluation such as preference's analysis (weighting factor), α-Level analysis, and sensitivity analysis are performed. Eventually, the MPS-GA, that is the MPS solution proposed by *Soares et al.* (2008), is being benchmarking of the fuzzy model.

Chapter 4 represents the GA for solving the crisp linear programming equivalent to the FMOLP. The implementation of genetic algorithm in MATLAB will be discussed briefly, particularly, about the customization of GA toolbox's function. The new genetic structure of MPS problem is also reviewed in this chapter. The numerical example is carried out to give clear picture of the GA use. At the end, differential evolution algorithm will be presented as benchmarking.

Chapter 5 gives an overview of the design and development of MPS information system. The information system is built of three system-agents; DOSIMIS.3, MATLAB and SAP. The integration among these agents will be described intensely in this chapter. The first topic to be discussed is the system design which covers the designing of closed-loop MPS system, the development of integration model and business process. Thus, all application programming interfaces (API) of each agent are reviewed and one, that is most feasible to be implemented in this case, is chosen. This chapter also explains the data exchange among system-agent and context diagram that provides the global sight of the selected information platform. At last discussion, this chapter describes step-by-step how to maintain the sales operation and planning (SOP), the demand management, and the production planning master data as well as how to run MRP in PP-SAP module.

Chapter 6 presents the development of simulation model. The development is divided into five phases; problem formulation, model development, model validation, experimental testing and output analysis. The simulation aims to mimic the real execution of planned orders in the manufacturing system.

Chapter 7 explains the implementation of closed-loop MPS to develop a valid and realistic master production schedule. The performance measures are defined to quantify its feasibility. The adjustment is required if the tentative MPS is considered as infeasible. The MPS adjustment is carried out by leveling load capacity from over-utilized resources to under-utilized resources. For this purpose, the fuzzy multi-objective is developed to determine the optimum quantity to be leveled such that the total overtime and defect products can be minimized. The numerical example of MPS creation and evaluation is also presented in this chapter.

Chapter 8 provides the conclusion of thesis and further research to improve the methodology proposed in this research.

Chapter 2

MPS and Optimization Technique

The comprehensive approach is a necessity in order to generate valid and realistic master production schedule (MPS), particularly, when the number of products, number of periods, and number of resources are large and involving fuzzy objectives and resources. For providing broad overview of MPS, this chapter presents the MPS fundamental theory, the manufacturing resource planning (MRP II) framework, and optimization technique that can be potentially employed to improve MPS solution.

2.1 Fundamental of MPS

The master production schedule (MPS) is an operational plan, a subset of the larger production plan. It is a declaration of what the company expects to manufacture. The master plan is not a sales forecast, which represents a demand declaration. It should take into consideration the demand, pending orders, material availability, projected ending inventory levels, capacity availability, managerial policies and goals. American Production and Inventory Control Society (APICS)[2] define master production schedule as

> *(1) The anticipated build schedule of those items assigned to the master scheduler. The master scheduler maintains this schedule and in turn, it becomes a set of planning numbers that drives material requirement planning. It represents what the company plans to produce expressed in specific configurations, quantities, and dates. The master production schedule is not a sales forecast that represents a statement of demand. The master production schedule must take into account the forecast, the production plan, and other important considerations such as backlog, availability of material, availability of capacity, management policy and goals, etc.*
>
> *(2) It is the result of the master scheduling process. The master schedule is a presentation of demand, forecast, backlog, the MPS projected on-hand inventory, and the available-to-promise quantity.*

[2] APICS Dictionary, 9th Edition (1998),

According to *Proud* (1999), the real challenge of MPS development is to effectively balance product supply with product demand (Figure 2). The demand can consist of a forecast, customer orders (which may or may not be part of that forecast), contracts or long-term agreements, engineering prototypes, branch warehouse requirements (i.e. replenishing a distribution center), or orders from another division within the company if the product in question is, in turn, a component of that division's product. Demand can also originate in the need for specials (industry shows, sample), service parts or spares, increase in safety stock requirements, or lot sizes. All those demands commonly refer to "Gross Requirement".

To satisfy these demands, the master scheduler needs to consider the availability of materials and capacity resources. These materials include those being produced internally as well as those being procured from outside sources. Besides the item itself, the other parameters such as quantities, dates, and lead time must be also taken into account. The capacity involves people and equipment – both of one's own company and of its suppliers. Time, space, and money are also important parameters need to be considered.

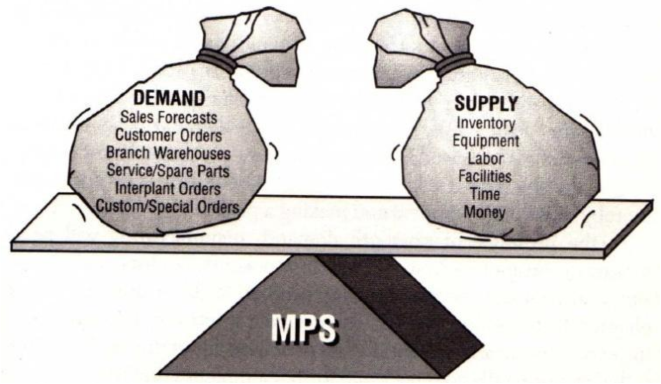


Figure 2 Balancing Supply and Demand[3]

The following are the detail parameters involving in a MPS processes. *Vieira G. E.* (2004) categorized these parameters into two groups, that are *input parameters* (updatable) and *output parameters* (non-updatable). Most output parameters result from the calculation of input parameters, and therefore, are non-updatable.

Main Input Parameters

- *Planning horizon*: Usually week to a couple of months
- *Resources and products* (SKUs)
- *Gross requirement*: mainly demand or production forecasts and customers orders. Usually, initial periods (time buckets) rely more on orders while further periods more on forecasts.

[3] Source: Proud (1999), *Master Scheduling*.

- *Subcontracted*: Quantity to be manufactured by third-party companies at a certain period of time.
- *Standard lot size*: The quantity to be manufactured should be a multiple of the standard lot size. It can be estimated based on costs, pallet sizes, minimum raw material (or components) purchase order size, number of parts per box.
- *Minimum lot size*: the minimum quantity to be scheduled
- *On-hand inventory*: Represents the SKU inventory at the beginning of the planning horizon. Sometimes it is confused with beginning inventory (output parameter explained later)
- *Safety (or minimum) inventory*: Quantity of inventory kept to deal with uncertainties, usually when demand surpasses forecast.
- *Maximum inventory*: Maximum quantity the company can carry in a time period. This is particularly important for perishable products, where maximum shelf-life is an important issue. It can also be given in term of maximum inventory coverage, meaning that inventory should not cover more than a given period.
- *Production rate*: How much a resource can manufacture of a product per time unit. The reciprocal would be how much time the production of one unit consumes of capacity. (A value equal to zero means that a product cannot be made at the resource).
- *Changeover (or setup) time*: Time needed to prepare a production resource (it is usually assumed that this time consumes capacity from the resource, also known as internal setup). It can depend on the product type sequence (sequence dependent setup) or not (sequence independent setup).
- *Backlogging*: Maximum quantity of a product (derived from customer orders) that cannot be made at the desired time bucket but can be manufactured in future periods.
- *Capacity*: Regular capacity available from a resource. Usually number of hours or days.
- *Overtime*: Maximum number of hours (or days) that can be used as overtime per time period per resource.

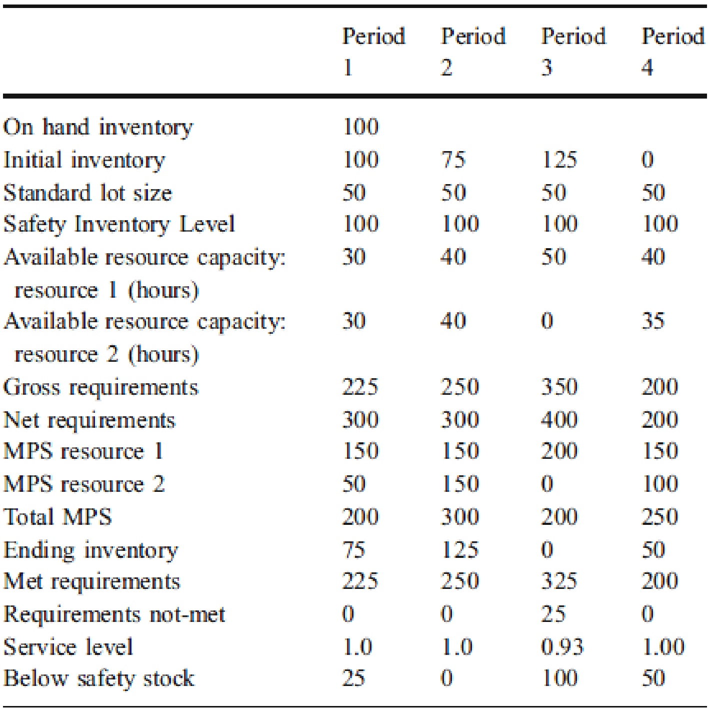
Main Output Parameters

- *Beginning inventory*: Quantity of a product available at the beginning of a time period. In the first time bucket, it equals the on-hand inventory; for the remaining periods, it equals the ending inventory of the previous period.
- *Ending inventory*: Quantity of products available at the end of a time period
- *Net requirement*: Represents what should be manufactured. It will be in fact manufactured if there is enough capacity. It is directly calculated from gross requirement, initial inventory levels, maximum inventory, subcontracted, minimum lot size, and standard lot size.

- *Master production schedules (MPS) solution*: Contains the final results with the product quantities to be manufactured, by which resources, through the planning horizon.
- *Requirement not met*: Shows in absolute term how much of the gross requirements will not be met by the master plan. This quantity can become backlogging and be transferred to future periods, if allowed by the scheduler.
- *Service Level*: A percentage representing how much of the gross requirement (demand and orders) will be met by the MPS.

Vieira & Ribas (2004) illustrate the complexity in the creation of a master plan through simple example, illustrated in Table 1.

Table 1 A simplified example of master production[4]



	Period 1	Period 2	Period 3	Period 4
On hand inventory	100			
Initial inventory	100	75	125	0
Standard lot size	50	50	50	50
Safety Inventory Level	100	100	100	100
Available resource capacity: resource 1 (hours)	30	40	50	40
Available resource capacity: resource 2 (hours)	30	40	0	35
Gross requirements	225	250	350	200
Net requirements	300	300	400	200
MPS resource 1	150	150	200	150
MPS resource 2	50	150	0	100
Total MPS	200	300	200	250
Ending inventory	75	125	0	50
Met requirements	225	250	325	200
Requirements not-met	0	0	25	0
Service level	1.0	1.0	0.93	1.00
Below safety stock	25	0	100	50

The example describes the master plan for a single product, without detailing the available capacity from resources 1 and 2. One can see that some of the total MPS values do not correspond to the net requirements due to the finite capacity restrictions not shown there. From this example, one can see that the main input data for the MPS are on hand inventory, safety inventory level, standard lot size, and gross requirements (certainly production rates or capacity needed per product per resource is another type of input, but this is not shown in that example). Net requirements indicate what would be made if there were enough capacity. Net requirements are calculated considering the gross requirements, inventory levels, and

[4] Source: (Vieira & Ribas, 2004)

production lot sizes. The last row in the table presents the quantity below the expected safety inventory level. Certainly, with a larger number of products, productive resources and capacity availability and consumption rates, and more planning periods, the master production schedule would not have been so easily created.

In general, the strategy of MPS creation is different from one company to other depending on the production approach, the variety of material produced, and the markets served by the company. In this context, three classic types of MPS approaches have been identified: make-to-stock (MTS), make-to-order (MTO), and assemble-to-order (ATO)[5].

The **make-to-stock** company produces in batches, carrying finished goods inventories for most, if not all, of its end items. Firms that make to stock are often producing consumer products. In this business environment, the MPS is the production statement of how much and when each end item is to be produced.

The **make-to-order** company, in general, carries no finished-goods inventors and builds each customer order as needed. In the MTO firm, the MPS unit is typically defined as the particular end item or set of items comprising a customer order. The definition is difficult since part of the job is to define the product; that is, design take place as construction take place. Production often starts before a complete product definition and bill of materials have been determined.

The **assemble-to-order** firm is typified by an almost limitless number of possible end item configurations, all made from combination of basic components and subassemblies. Customer delivery time requirements are often shorter than total manufacturing lead times, so production must be started in anticipation of customer orders. The large number of end item possibilities makes forecasting extremely difficult and stocking end items very risky. As a result, the assemble-to-order firm tries to maintain flexibility, starting basic component or subassemblies into production, and not starting final assembly until a customer order is received. In this firm, the MPS unit is stated in planning bills of material, such as an average lift truck of some model series. The MPS unit (planning bills) has its component as a set of common parts and options.

The primary difference between MTS, MTO, and ATO firms is in the definition of the MPS units. However, many MPS techniques are useful for any kind of MPS unit definition.

[5] The MPS description in MTS, MTO, & ATO is cited from *Vollmann et al.* (1997)

2.2 MRP II Framework

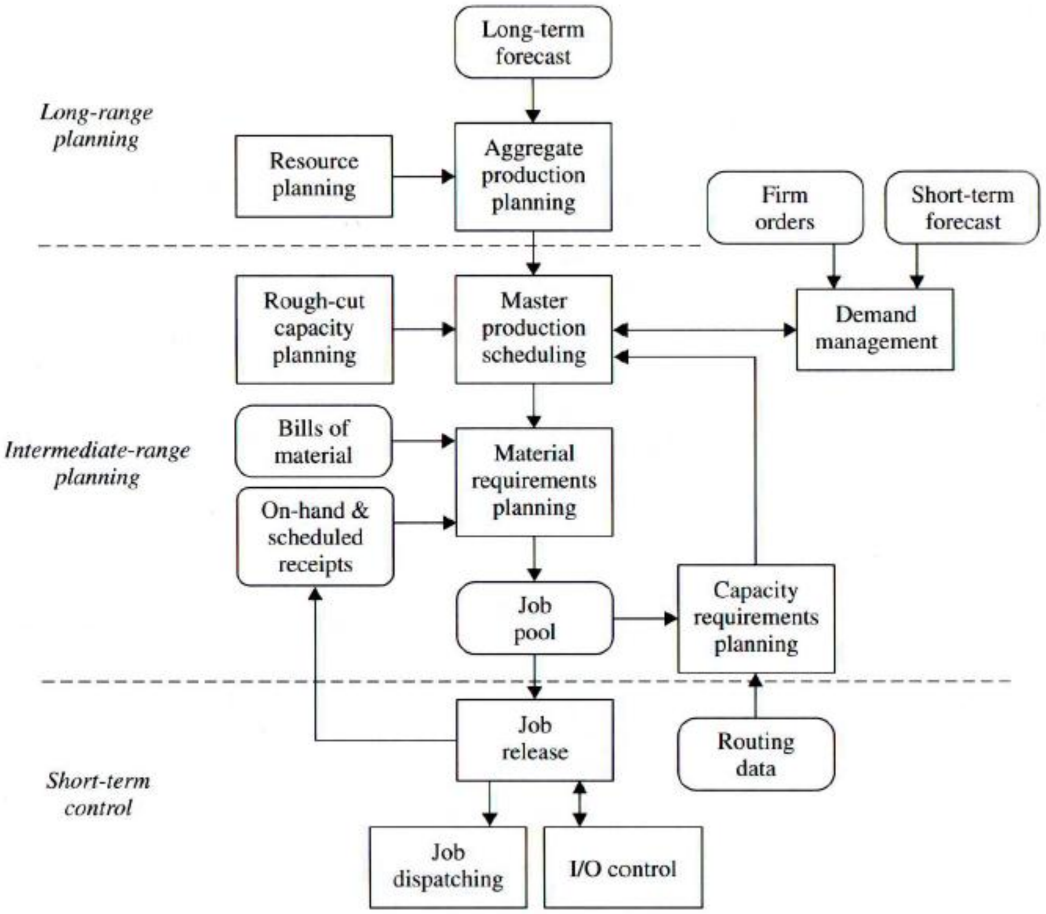

Long-range planning
Intermediate-range planning
Short-term control
Long-term forecast
Resource planning
Aggregate production planning
Firm orders
Short-term forecast
Rough-cut capacity planning
Master production scheduling
Demand management
Bills of material
On-hand & scheduled receipts
Material requirements planning
Job pool
Capacity requirements planning
Job release
Routing data
Job dispatching
I/O control

Figure 3 Relationship of MPS to other manufacturing planning and control activities[6]

Figure 3 depicts an instance of the MRP II hierarchy, which is also adopted by modern Enterprise Resource Planning (ERP). This scheme intends to show the elements in manufacturing planning and control systems that relate directly to master production schedules. The master scheduling block represents the focal point of all the production planning control activities. The demand management supplies all of the demand requirements and customer orders to the master production schedules. Likewise, the aggregate production planning, which represent production's role in the company's strategic business plan, indicates the operating constraints within which the MPS function must work. Furthermore, the Rough-cut capacity planning (RCCP) provides an analysis of master production schedule to determine the existence of manufacturing facilities that represents potential bottlenecks in production flow. At the end, the output of MPS drives the material requirement planning (MRP) as representation of tactical production planning.

[6] Source: *Hopp & Spearman* (2008), Factory Physics.

2.2.1 Aggregate Production Plan

In newer term, *sales and operation planning* (SOP) is often used to name the aggregate production planning. The change in terminology fosters the understanding that we are concerned with coordinating several critical activities in the company instead of only producing plans for manufacturing functions. The production planning or sales and operation planning includes the sales/marketing objectives, the cost and schedule objectives in manufacturing, and the inventory/financial objectives.

The production plan represents a strategic “game plan” for the company, which reflects the desired aggregate output from manufacturing. In some firms, the production plan is simply stated in term of the monthly or quarterly sales dollar output for the company as a whole, or for individual plants or business. In other firms, the production plan is stated in terms of the number of units to be produced in each major product line monthly for the next year. The units of manufacturing activities used in preparing the production plan vary considerably among companies. In addition to sales dollar output and the overall units per month for major product grouping, measures such as direct labor hours and pounds of product produced also used.

The production plan is knob on the business. It provides guidelines and constraints within which manufacturing is expected to operate. The production plan represents an agreement between marketing, manufacturing, and finance as to what (in aggregate) will be produced and made available for sale to the customers. This agreement forms the basis for budgetary and market planning for the company. A change in one aspect of a company’s strategic plans, such as the sales plan, the production plan, or the financial plan, necessarily means a change in the other two plans. Once the production plan is set, manufacturing has its task clearly defined – “to hit the production plan”. In many firms, the production plan is a key yardstick for measuring the performance of the plant managers. Likewise, once the production plan is established and agreed upon by top management, sales has its work defined – “to make the sales budget”. A sales budget that is not met (or is exceeded) means that unplanned finished-goods inventory (or excessive back orders) will be created.

The production planning activity at some firms is a highly structured process, often conducted on a monthly basis. This process often begins with the creation of a new sales forecast covering a year or more (long-term forecast). Next, any increase or decreases in inventory or backlog level are decided in order to determine total production requirements. Once the total demand to be placed in manufacturing in determined, the way in which manufacturing will meet this demand, that is, the production plan, can be decided. It is important to note that the production plan should equal the sales forecast in total over a designated period, perhaps a year. However, the timing of production may vary considerably from the sales forecast, creating planned increase (or decrease) in inventory levels. The steps involved in preparing the production plan occasionally take place over a period of several weeks or more, involving the top executives – the president, the vice-president manufacturing, the vice-president of marketing, and the vice president of finance.

The production plan feeds directly into the master production scheduling activity (see Fig. 2). The production plan provides guidelines within which the master production process takes place.

2.2.2 Demand Management

The demand management (Figure 3) encompasses forecasting, order entry, order promising (ATP), customer order service, physical distribution, and other customer-contract-related activities. Demand management also concerns other sources of demand for manufacturing capacity, including service-part demands, intercompany requirements, and pipeline inventory stocking. All quantities and timing for demands must be planned and controlled.

For many firms, planning and control of demand quantities and timing are a day-to-day interactive dialogue with customers. For other firms, particularly in the process industries, the critical coordination is in scheduling large inter and intra-company requirements. For still others, physical distribution is critical, since the factory must support a warehouse replenishment program, which can differ significantly from the pattern of final customer demand.

Demand management is a gateway module in Manufacturing Planning and Control (MPC) providing the link to the marketplace. Activities performed here provide coordination between manufacturing and the marketplace, sister plants, and warehouses. Through demand management, a channel communication between MPC systems and customer is maintained. Specific demand initiates actions throughout MPC, which ultimately result in product delivery and consumption of materials and capacities.

The process of converting the long-term aggregate forecast to a detailed forecast while tracking individual customer orders is one of demand management tasks. The output of the demand management module is a set of actual customer order plus a forecast of anticipated orders. As time progresses, the anticipated orders should be consumed by actual orders.

2.2.3 Rough-Cut Capacity Planning

The rough-cut capacity planning activity involves an analysis of the master production schedule to determine the existence of critical manufacturing facilities that are potential bottlenecks in the flow of production. Such analysis is typically made during master schedule review or whenever adjustments are made to the master production schedule. Every firm has several critical steps in production process that need to be carefully monitored as changes in the mix of product occur in the master schedule. At some firms, final assembly or finishing operation represent critical operations to be closely watched. In other firms, the critical operation involves a key machine that is operating on a three-shift basis. At still other firms, the critical operation may be a vendor who supplier a key raw material nor readily available from other sources. Rough-cut capacity planning identifies the impact of the MPS on these critical operations.

The rough-cut capacity analysis is typically performed much less frequently and covers a longer time span than the weekly shop load reports produced from capacity requirement planning (CRP) system using shop floor control and MRP data. The rough-cut analysis also involves the use of simpler capacity planning techniques and far less detailed information that the CRP analysis. This is, rough-cut capacity planning typically involves the use of bills of capacity or simple planning factors estimates instead of time-phased MRP and routing file data. The simple procedure, while less accurate, permits a quick analysis to indicate whether the MPS is feasible in view of the current capacity of the company. It also permits a rapid analysis of several alternatives that may be posed by top management in reviewing the MPS. Such analyses sometimes lead to a decision to make changes in the master schedule in order to accommodate the capacity limitation of the key facilities, or a decision to change the MPS and to implement adjustment to capacity involving overtime, subcontracting, alternate routing of work, employment level changes, and so forth. These changes are often planned over a considerable period of time, covering a quarter or more into the future.

In spite of providing quick analysis, RCCP has some limitations. Besides considering only critical resources, the RCCP assumes that the demand can be met without regard to how the work is scheduled within the process center (i.e., without any induced idle time). In this way, RCCP provides an optimistic estimate of what can be done. Another limitation of rough cut is that it ignores work-in-process and work completed. During comparing the required capacity and the planned capacity, the RCCP does not look at the work-in-process (WIP) or at what work was completed. This WIP netting does not take place until material requirement planning, capacity requirement planning, and shop floor control system are run. Thus, RCCP data is often invalid on the short term.

The rough-cut capacity planning block is shown Figure 3 as feeding information into the MPS function. This reflects its basic objective of keeping the MPS realistic in term of available capacity.

2.2.4 Material Requirement Planning

The basic function of MRP is revealed by its name – to plan material requirements. MRP is used to coordinate orders from within the plant and from outside. Outside orders are called purchase orders, while orders from within are called jobs. The main focus of MRP is on scheduling jobs and purchase orders to satisfy material requirements generated by external demand.

MRP deals with two basic dimensions of production control: quantities and timing. The system must determine appropriate production quantities of all types of items, from final products that are sold, to components used to build final products, to inputs purchased as raw materials. It must also determine production timing (i.e., job start times) that facilitates meeting order due dates.

In many MRP systems, time is divided into buckets, although some systems use continuous time. A bucket is an interval that is used to break time and demand into discrete chunks. The demand that accumulates over the time interval (bucket) is all considered due at the beginning of the bucket. Thus, if the bucket length is 1 week and during third week (bucket) there is demand for 200 units on Monday, 250 units on Tuesday, 100 on Wednesday, 50 on Thursday, and 350 on Friday, then demand for the third bucket is 950 units and is due on Monday morning. In the past, when data processing was more expensive, typical bucket sizes were one week or longer. Today, most modern MRP system use daily buckets, although there are still many systems using weeks.

MRP works with both finished products, or end items, and their constituent parts, called lower-level items. The relationship between end items and lower-level items is described by the bill of material (BOM). Demand for end items generates dependent demand for lower-level items. All demand for end items is independent demand, while typically most demand for lower-level items is dependent demand. However, there can be independent demand for lower-level items in the form of spare parts, part for research and quality tests, and so on.

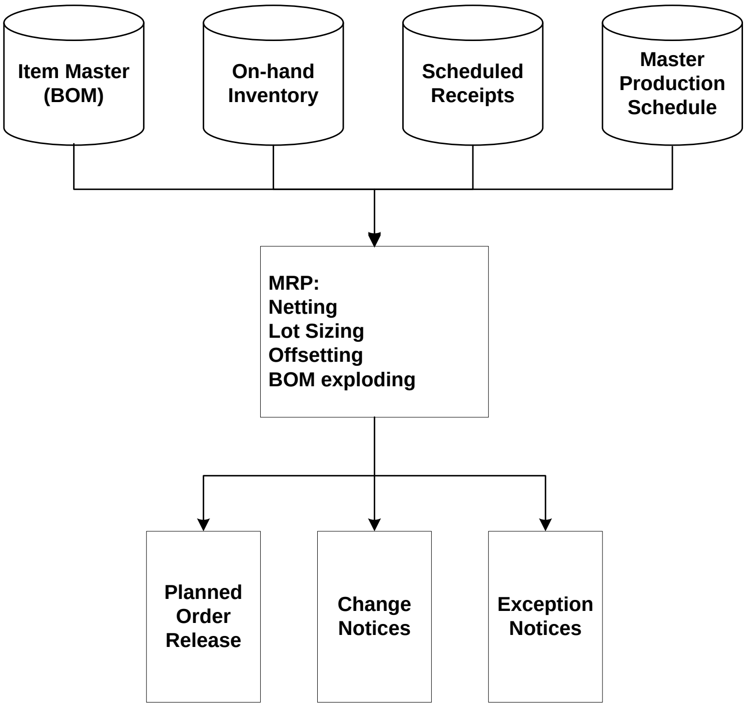


Figure 4 Schematic of MRP[7]

In addition to the BOM information, MRP requires information concerning independent demand, which comes from the master production schedule (MPS). The MPS contains gross requirements, the current inventory status known as on-hand inventory, and the status of outstanding orders (both purchased and manufacturing) known as scheduled receipts. Figure 4 depicts the schematic of MRP which describes the process, input and output of MRP.

For each level in the bill of material, beginning with end items, MRP does the following for each part:

1. *Netting*: Determine net requirements by subtracting on-hand inventory and any scheduled receipts from the gross requirements. The gross requirements for level-zero items come from the MPS, while those for lower-level items are the results of previous MRP iterations or are independent demand for those parts (e.g., spares). If the projected-on-hand becomes less than zero, there is a material requirement.
2. *Lot Sizing*: Divide the netted demand into appropriate lot sizes to form jobs
3. *Time phasing*: Offset the due dates of the jobs with lead times to determine start times.
4. *BOM Explosion*: Use start times, the lot sizes, and the BOM to generate gross requirements of any required components at the next level(s).
5. *Iterate*: Repeat these steps until all levels are processed.

[7] Source: (Hopp & Spearman, 2008), Factory Physics

The basic inputs to MRP are a forecast of demand for end items, the associated bills of material, and the current inventory status, plus any data needed to specify production policies. These data come from three sources: (1) the item master file, (2) the master production schedule, (3) the inventory status file.

The item master file is organized by part number and contains, at a minimum, a description of the part, bill-of-material information, lot-sizing information, and planning lead times. The minimum BOM data for a part are components and quantities that are directly required to make the part.

The master production schedule is the source of demand for the MRP system. It gives the quantity and due dates for all parts that have independent demand. This will include demand for all end items as well as external demand for lower-level parts (e.g., demand for spare parts). The minimum information contained in the master production schedule is a set of record containing a part number, a need quantity, and due date for each purchase order.

On-hand inventory data are stored by part number and contains information describing the part, where it is located and how many are currently on hand. On-hand inventory includes raw material stock, “crib” stock (i.e. inventory that has been processed since being raw material and kept within the plant), and assembly stock. On-hand inventory may also contain information about allocation that indicates how many parts are reserved for specific jobs.

Scheduled Receipts file contains all previously released orders, either purchase orders or manufacturing jobs. A scheduled receipt (SR) is a planned order release that has actually been released. For purchased parts, this involves executing a purchase order (PO) and sending it to a vendor. For manufactured parts, this entails gathering all necessary routing and manufacturing information, allocating the necessary inventory for the job, and releasing the job to the plant.

The output of MRP system includes planned order releases, change notice, and exception reports. Planned order releases eventually become the jobs that are processed in the plant. A ***planned order release*** (POR) contains at least three pieces of information: (1) the part number, (2) the number of units required, and (3) the due date for the job. ***Change notices*** indicate modification of existing jobs, such as changes in due dates or priorities. ***Exception reports***, as in any large management information system, are used to notify the users that there are discrepancies between what is expected and what will transpire. Such report might indicate job count differences, inventory discrepancies, imminently tardy jobs, and the like.

2.2.5 Capacity Requirement Planning

Capacity requirement planning (CRP) differs from the rough-cut planning procedures in four respects.

1. The CRP utilizes the time-phased material plan information produced by an MRP system.
2. The CRP system's gross-to-net feature takes into account production capacity already stored in the form of inventories of both components and assembled products.
3. The shop-floor control system accounts for the current status of all work-in-process in the shop. It considers only the remaining work on open shop orders.
4. The CRP takes into account demand for services parts, other demand that may not be accounted for in the MPS, and any additional capacity that might be required by MRP planner reacting to scrap, item record errors, on so on.

In spite of its name, capacity requirement planning does not generate finite capacity analysis. Instead, CRP performs what is called *infinite forward loading*[8]. CRP predicts job completion times for each process center, using given fixed lead times, and then computes a predicted loading over time. These loading are then compared against the available capacity, but correction is not made for an overloaded situation.

Note that even when load exceeds capacity, CRP assumes that the time to go through the process center does not change. Of course, we know that it will take longer to get through a heavily loaded process center than a lightly loaded one. Hence, all the estimates of CRP beyond such an overloaded condition will be in error. Therefore, CRP is typically not a good predictor of load conditions except in the very near term. Another problem with the CRP is that it tells the planner only that there is a problem; it offers nothing about what caused the problem or what can be done to alleviate it. To determine this, the planner must first obtain a report that disaggregates the load to determine which jobs are causing the problem, and then must use pegging to track the cause back to demand on the MPS. This can be quite tedious.

Thus, in spite of its hopeful introduction and worthy goals, there are fundamental problems with CRP. First, there are enormous data requirements, and the output is voluminous and tedious. Second is the fact that it offers no remedy to an overloaded situation. Finally, since the procedure uses infinite loading and many modern system can perform true finite capacity loading, fewer and fewer companies are seriously using CRP.

[8] Unlike MRP and CRP works, true finite capacity analysis does not assume a fixed lead time. Instead the time to go through a manufacturing operation depends on how many other jobs are already there and their relative priority.

2.3 Fuzzy Multi Objective Decision Making

Making decisions is part of our daily lives. A major concern is that almost all decision problems have multiple, usually conflicting, criteria. In classic decision-making process, the methods generally assume that all criteria and their respective weights are expressed in crisp value. However, many of the decision making problems in the real world take place in an environment in which the goals, the constraints, and the consequences of possible actions are not known precisely (Bellman & Zadeh, 1970). In some cases, criteria even are only be expressed qualitatively or by using linguistic terms. These situations imply that a real decision problem is very complicated and thus often seems to be little suited to mathematical modeling because there is no crisp definition (Zimmermann & Zysno, 1985).

It is not presumption that the classic decision-making methods such as the classic MADM is inapplicable in certain areas particularly when the decision situation involves fuzzy data. In general, the term “fuzzy” commonly refers to a situation in which the attribute or goal cannot be defined crisply, because of the absence of well-defined boundaries of the set of observation to which the description applies. For instance, available labor hours and available material may be “around 1530” hours and “about 1200”units, respectively. Due to this fact, the fuzzy approach which is conceptually proposed by Zadeh, is being one of the most rapidly developing methodologies for solving multi objective decision making in modern problem.

Indeed, the MPS development involves many vague parameters such as fuzzy resource capacity, fuzzy customer demand and fuzzy production rate. This condition make decision maker difficult to define the objective crisply and at the end the actual optimum solution cannot be obtained properly. Concerning this issue, the fuzzy multi-objective linear programming should be an appropriate approach for dealing with the MPS problem.

2.3.1 Fuzzy Set Theory

In 1965, L.A. Zadeh published his famous paper “Fuzzy Sets” in Information and Control providing a new mathematical tool which enables us to describe and handle vague or ambiguous notions such as “a set of all real numbers which are much greater than 1”, “a set of beautiful women”, or “the set of tall men”. Since then, fuzzy set theory has been rapidly developed by Zadeh himself and numerous researchers, and an increasing number of successful real applications of this theory in a wide variety of unexpected fields have been appearing. The main idea of fuzzy set theory is quite intuitive and natural: Instead of determining the exact boundaries as in an ordinary set, a fuzzy set allows no sharply defined boundaries because of a generalization of a characteristic functions to a membership function.

Definition 1 (Fuzzy Sets)

In general, a fuzzy set initiated by Zadeh (1965) is defined as follows:

Let *X* denotes a universal set. Then a fuzzy set $\tilde{L}$ of *X* is defined by its membership function

$$\mu_L : X \rightarrow [0,1], \tag{2.1}$$

which assign to each element $x \in X$ a real number $\mu_L(x)$ in the interval [0, 1] where the value of $\mu_L(x)$ at *x* represents the grade of membership of *x* in $\tilde{L}$. Thus, the nearer the value of $\mu_L(x)$ is unity, the higher grade of membership of x in $\tilde{L}$.

A fuzzy set $\tilde{L}$ can be characterized as a set of ordered pairs of element x and grade $\mu_L(x)$ and is often written as

$$\tilde{L} = \{(x,\ \mu_L(x)) \mid x \in X\} \tag{2.2}$$

(Example 2-1)

A realtor wants to classify the house he offers to his clients. One indicator of comfort of these houses is the number of bedrooms in it. Let *X* = {1, 2, 3, 4… 10} be the set of available types of houses described by x is number of bedrooms in a house. Then the fuzzy set "comfortable type of house for a 4-person family" may be described as

$$\tilde{L} = \{(1, 0.2), (2, 0.5), (3, 0.8), (4, 1), (5, 0.7), (6, 0.3)\}$$

A fuzzy set is denoted by an ordered set of pairs, the first element of which denotes the element and the second the degree of membership. In other literatures one may find different ways of denoting fuzzy sets.

Definition 2 ($\boldsymbol{\alpha}$ − level Set)

The α level set of a fuzzy set $\tilde{L}$ is defined as an ordinary set $\tilde{L}_\alpha$ for which the degree of its membership function exceed the level α.

$$\tilde{L}_\alpha = \{x \mid \mu_A(x) \geq \alpha \ and\ x \in X\}, and\ \alpha\ \in [0,1] \tag{2.3}$$

(Example 2-2)

Let look at example 2-1.

$$\tilde{L}_{0.7} = \{3, 4, 5\};\ \ \tilde{L}_{0.5} = \{2, 3, 4, 5\}$$

2.3.2 Fuzzy Decision

In a decision-making process, the principles ingredients include a set alternatives/decisions, a set of constraints on the choice between alternatives, and a performance/objective function which associates with each alternative the gain (or loss) resulting from the choice of that alternative. When this decision-making process is used in a fuzzy, *Bellman and Zadeh* (1970) proposed that the symmetry erases the difference between them and makes it possible to relate in a relatively simple way the concept of a decision to those of the goals and constraints of a decision process.

Assume that objective(s) and constraints in an imprecise situation can be represented by fuzzy sets. For an illustration, suppose that we have a fuzzy goal $\tilde{G}$ and a fuzzy constraint $\tilde{C}$ in a decision space *X* expressed as follows.

$\tilde{G}$: *x* should be substantially larger then 10, with

$$\mu_G(x) = \begin{cases} [1+(x-10)^{-2}]^{-1} & x \geq 10 \\ 0 & x < 10 \end{cases} \tag{2.4}$$

$\tilde{C}$: *x* should be in the vicinity of 15

$$\mu_C(x) = [1+(x-15)^4]^{-1} \tag{2.5}$$

Then, with the assumption of the symmetry we may make decisions which satisfy both the constraint "and" the goal. That is: $\tilde{G}$ and $\tilde{C}$ are connected to another by the operator "and" which corresponds to the intersection of fuzzy sets. This implies that the combined effect of the fuzzy goal $\tilde{G}$ and the fuzzy constraint $\tilde{C}$ on the choice of alternatives may be represented by the intersection $\tilde{G} \cap \tilde{C}$, with the membership function (Figure 5).

$$\mu_{G\cap C}(x) = \mu_G(x)\ \wedge \mu_C(x) = \min\{\mu_G(x),\ \mu_C(x)\} \tag{2.6}$$

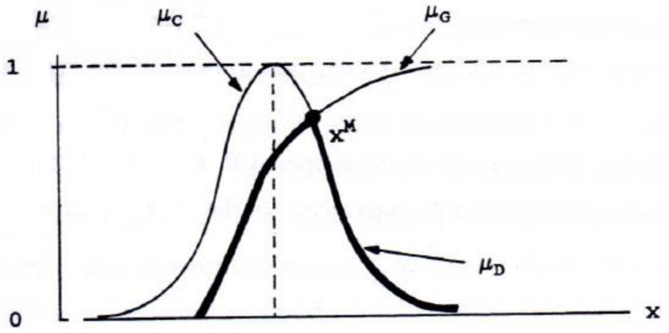


Figure 5 The relationship of $\tilde{G}$, $\tilde{C}$, and $\tilde{D}$

Then *Bellmann and Zadeh* (1970) proposed that a fuzzy decision may be defined as the fuzzy set of alternatives resulting from the intersection of the goals and the constraints. That is: the decision $\widetilde{D} = \tilde{G} \cap \tilde{C}$ is fuzzy set resulting from the intersection of $\tilde{G}$ and $\tilde{C}$, and has its membership function $\mu_D = \mu_G \cap \mu_C$ (Figure 5).

Maximizing decision then can be defined as follows:

$$\mu_D(x^M) = \max\ \mu_D(x) \quad for\ x \in X \tag{2.7}$$

If $\mu_D(x)$ has a unique maximum at x^M, then the maximizing decision is uniquely defined crisp decision which can be interpreted as the action which belongs to all fuzzy sets representing either constraints or objective(s) with the highest possible degree of membership. From the above analysis, it is obvious that the decision $\widetilde{D}*$ is obtained by using the max-min operator.

Suppose that the goals are defined as fuzzy sets $\tilde{G}_1, \tilde{G}_2, \ \ldots, \tilde{G}_k$ in $Y = \{y \mid y \in Y\}$ while the constraints $\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_m$ are defined as fuzzy set in $X = \{x \mid x \ \in X\}$. Now, given a fuzzy set $\tilde{G}_i$ in Y, we can then find a fuzzy set $\tilde{C}_i$ in X which induces $\tilde{G}_i$ in Y. Specially, with $y = f(x)$ the membership function of $\tilde{G}_i$ is given by the equality:

$$\mu_G(x) = \mu_{G_i}(f(x)), \ \forall_i \tag{2.8}$$

The decision $\widetilde{D}$, then, can be expressed as the intersection of $\tilde{G}_1, \tilde{G}_2, \ \ldots, \tilde{G}_k$ and $\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_m$. By using Equation 2.8, we can then obtain the following:

$$\mu_D(x) = \mu_{G1}(f(x)) \wedge \ldots \wedge \mu_{Gk}(f(x)) \wedge \mu_{C1}(x) \wedge \ldots \wedge \mu_{Cm}(x) \tag{2.9}$$

where $f: X \to Y$. In this way, the case where the goals and the constraints are defined as fuzzy sets in different spaces can be reduced to the case where they are defined in the same space. The maximizing decision x^* as shown in Equation 2.7 can then be obtained.

2.3.3 Fuzzy and Non-Fuzzy Linear Programming

Linear programming (LP) models shall be considered as a special kind of decision model: the decision space is defined by the constraints; the "goal" (utility function) is defined by the objective function; and the type of decision is assumed under certainty. The classical model of linear programming can be stated as

$$\begin{aligned} &maximize \quad f(x) = \ Cx \\ &subject\ to \quad\ Ax \leq b \\ &\qquad\qquad\qquad x \geq 0 \end{aligned} \tag{2.10}$$

Let us now depart from the classical assumption that all coefficients of C, A, and b are crisp numbers, where $C = \left(c_{1,}\, c_{2,}\, c_{3,} \ldots c_n\right); A = \left(a_1, a_2, a_3, \ldots a_j\right)$ and b is a constraint value.

If we assume that the LP-decision has to be made in fuzzy environments, quite a number of possible modifications of Equation 2.10 exist. First of all the decision maker might really not want to actually maximize or minimize the objective function. Rather he might want to reach some aspiration levels which might not even be defined crisply. Thus he might want to “improve the present cost situation considerably,” and so on.

Secondly, the constraint might be vague in one of the following ways: The $\leq$ sign might not be meant in the strictly mathematical sense but smaller violations might well be acceptable. This can happen if the constraints represent aspiration levels as mentioned above or if, for instance, the constraints represent sensory requirement (taste, color, smell, etc) which cannot adequately be approximated by a crisp constraints. Of course, the coefficients of the vectors b or c of the matrix A itself can have a fuzzy character either because they are fuzzy in nature or because perception of them is fuzzy.

Finally the role of the constraints can be different from that in classical linear programming where the violation of any single constraint by any amount renders the solution infeasible. The decision maker might accept small violations of constraints but might also attach different (crisp or fuzzy) degrees of importance to violations of different constraints. Fuzzy linear programming offers a number of ways to allow for all those types of vagueness.

Before we develop a specific model of linear programming in a fuzzy environment it should have become clear, that by contrast to classical linear programming “fuzzy linear programming” is not a uniquely defined type of model but that many variations are possible, depending on the assumptions or features of the real situation to be modeled.

Let us now turn to a first basic model for “fuzzy linear programming”. In model Equation 2.10 we shall assume, that the decision maker can establish an aspiration level, z, for the value of the objective function he wants to achieve and that each of the constraints is modeled as fuzzy set. Outcome of fuzzy LP then becomes:

$Find\ x\ such\ that$

$$Cx \gtrsim z$$

$$Ax \lesssim b$$

$$x \geq 0 \tag{2.11}$$

Here $\lesssim$ denotes the fuzzified version of $\leq$ and has the linguistic interpretation "essentially smaller than or equal". $\gtrsim$ denotes the fuzzified version of $\geq$ and has the linguistic interpretation "essentially greater than or equal". The objective function in Equation 2.10 might have to be written as a minimizing goal in order to consider z as an upper bound.

We see that Equation 2.11 is fully symmetric with respect to objective function and constraints. *Zimmermann* (1976) expressed the problem as follows:

$$Bx \lesssim d$$

$$x \geq 0 \tag{2.12}$$

Where,

$$B = \begin{bmatrix} -C \\ A \end{bmatrix}, \quad d = \begin{bmatrix} -z \\ b \end{bmatrix} \tag{2.13}$$

Each of the *(m + 1)* rows of Equation 2.12 shall now be represented by a fuzzy set, the membership function of which are $\mu_i(x),\ i = 1,2,..m+1$. Following the rule of fuzzy decision, the membership function of fuzzy set "decision" of Equation 2.12 is

$$\mu_{D(x)} = \min_i\{\,\mu_i(x)\} \tag{2.14}$$

$\mu_i(x)$ can be interpreted as the degree to which x satisfies the fuzzy inequality $B_i x \leq d_i$ (where B_i denotes the i^{th} row of B).

Assuming that the decision maker is interested not in a fuzzy set but in a crisp "optimal" solution we could suggest the "maximizing solution" to Equation 2.14, which is the solution to the possible nonlinear programming problem

$$\max_{x\geq 0}\ \min_i\ \{\,\mu_i(x)\} = \max_{x\geq 0}\ \mu_D(x) \tag{2.15}$$

Now we have to specify the membership function $\mu_i(x)$. $\mu_i(x)$ should be 0 if the constraints (including objective function) are strongly violated, and 1 if they are very well satisfied (i.e. satisfied in the crisp sense); and $\mu_i(x)$ should increase monotonously from 0 to 1, that is:

$$\mu_i(x) = \begin{cases} 1 & B_i x \leq d_i \\ \in [0,1] & d_i < B_i x \leq d_i + p_i \\ 0 & B_i x > d_i + p_i \end{cases} \tag{2.16}$$

Using the simplest type of membership function we assume them to be linearly increasing over the “tolerance interval” p_i :

$$\mu_i(x) = \begin{cases} 1 & B_i x \leq d_i \\ 1 - (B_i x - d_i)/p_i & d_i < B_i x \leq d_i + p_i \\ 0 & B_i x > d_i + p_i \end{cases} \qquad (2.17)$$

The p_i are subjectively chosen constants of admissible violations of the constraints and the objective function. Substituting Equation 2.17 into Equation 2.15 yields, after some rearrangement (Zimmermann H. J., 1976) and with some additional assumptions,

$$\max_{x\geq 0} \ \min_{i} \left(1 - \frac{B_i x - d_i}{p_i}\right) \qquad (2.18)$$

By introducing the auxiliary variable λ this problem can be transformed into the following equivalent conventional linear programming problem:

$$\begin{aligned} &maximize && \lambda \\ &subject\ to && \lambda p_i + B_i x \leq d_i + p_i \qquad i = 1,2, \ldots., m+1 \\ & && x \geq 0 \qquad (2.19) \end{aligned}$$

where $\lambda = \min\ \{\mu_i(x)\}$. If the optimal solution to Equation 2.19 is vector (λ, x_0) then x_0 is the maximizing solution of Equation 2.15 of model in Equation 2.11 with assuming membership functions as specified in Equation 2.17. The Equation 2.19 is also often expressed in the following simple form:

$$\begin{aligned} &maximize && \lambda \\ &subject\ to && \lambda \leq \ \mu_i(x) \qquad i = 1,2, \ldots., m+1 \\ & && x \geq 0 \qquad (2.20) \end{aligned}$$

2.3.4 Fuzzy Multi-Objective Linear Programming

Zimmermann (1978) extended his fuzzy linear programming approach to the following multi-objective linear programming problem with k linear objective functions $z_i(x) = C_i x, i = 1, \ldots, k$:

$$\begin{aligned} &minimize && Z(x) = \{\tilde{z}_1(x), \tilde{z}_2(x), \ldots, \tilde{z}_k(x)\ \}^T \\ &subject\ to && Ax\ \leq b \\ & && x \geq 0 \qquad (2.21) \end{aligned}$$

Where $C_i = (c_{i1}, \ldots, c_{in}),\ i = 1, \ldots, k,\ \ x = (x_1, \ldots, x_n)^T,\ b = (b_1, \ldots, b_m)^T$ and $A = [a_{ij}]$ is an $m \times n$ matrix.

For each of the objective functions $z_i(x) = c_i x,\ i = 1, \ldots, k,$ of this problem, assume that the decision maker (DM) has a fuzzy goal such as "the objective function $z_i(x)$ should be substantially less than or equal to some value z_i." Then the corresponding linear membership function $\mu_i(z_i(x))$ is defined as

$$\mu_i(z_i(x)) = \begin{cases} 0 & z_i(x) \geq z_i^1 \\ (z_i^1 - z_i(x))/(z_i^1 - z_i^0) & z_i^0 \geq z_i(x) \geq z_i^1 \\ 1 & z_i(x) \leq z_i^0 \end{cases} \qquad (2.22)$$

Where z_i^0 and z_i^1 denotes the value of the objective function $z_i(x)$ such that the degree of membership function is 0 or 1 respectively. Figure 6 illustrates the graph of possible shape of the linear membership function.

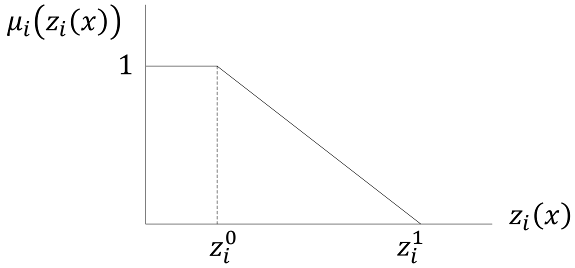


Figure 6 Non-increasing linear membership function

Using such linear membership functions $\mu_i(z_i(x)),\ i = 1, \ldots, k,$ and following the fuzzy decision of *Bellman and Zadeh* (1970), the original multi-objective linear programming can be interpreted as follows:

$$\begin{aligned} &maximize \quad \min_{i=1,\ldots,k} \{\mu_i(z_i(x))\} \\ &subject\ to \quad Ax \leq b, x \geq 0 \end{aligned} \qquad (2.23)$$

By introducing the auxiliary variable λ, it can be reduced to the following conventional linear programming problem

$$\begin{aligned} &maximize \ \ \lambda \\ &subject\ to \ \ \lambda \leq \mu_i(z_i(x)) \qquad i = 1,2, \ldots., k \\ &\qquad\qquad\quad Ax \leq b,\ x \geq 0 \end{aligned} \qquad (2.24)$$

In the case where the objectives and constraints are fuzzy, Equation 2.21 may change into the following maximization fuzzy multi-objective linear problem:

$$\begin{aligned} &maximize \quad Z(x) = \{\tilde{z}_1(x), \tilde{z}_2(x), \dots, \tilde{z}_k(x)\}^T \\ &subject\ to \quad \tilde{g}_r(x) = A_r(x) \leq \tilde{b}_r \qquad r = 1,2,\dots,h \qquad (fuzzy\ constraint) \\ &\qquad\qquad g_n(x) = A_n(x) \leq b_n \qquad n = h+1,\dots,t \quad (deterministic\ constraint) \\ &\qquad\qquad x \geq 0 \end{aligned} \tag{2.25}$$

Where $A_r(x) \leq \tilde{b}_r$ is equivalent to $A_r(x) \lesssim b_r$. The tolerance p_r for fuzzy constraint $\tilde{b}_r$ is given initially. In the fuzzy set theory, the fuzzy objective function and the fuzzy constraints are defined by their corresponding membership functions. For simplicity, let assume that the membership function of the fuzzy objectives $\mu_{zi}(x)$ are non-decreasing continuous linear function, and the fuzzy constraints $\mu_{gr}(x)$ are non-increasing continuous linear membership function as follows (Figure 7):

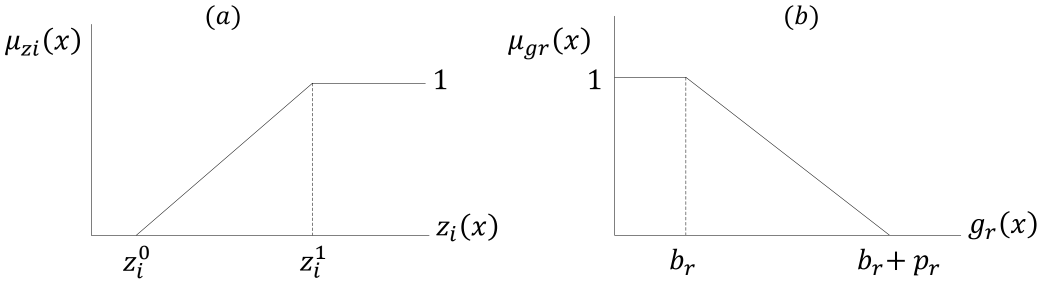


Figure 7 The membership function: (a) for fuzzy objectives (b) for fuzzy constraints

$$\mu_{zi}(x) = \begin{cases} 1 & z_i(x) \geq z_i^1 \\ (z_i(x) - z_i^0)/(z_i^1 - z_i^0) & z_i^0 \geq z_i(x) \geq z_i^1 \\ 0 & z_i(x) \leq z_i^0 \end{cases} \tag{2.26}$$

$$\mu_{gr}(x) = \begin{cases} 0 & g_r(x) \geq b_r + p_r \\ ((b_r + p_r) - g_r(x))/(p_r) & b_r \geq g_r(x) \geq b_r + p_r \\ 1 & g_r(x) \leq b_r \end{cases} \tag{2.27}$$

Using Zimmermann's approach, a fuzzy solution is given by the intersection of all the fuzzy sets representing either fuzzy objective or fuzzy constraints (Equation 2.6). Therefore, the fuzzy solution for the problem with k objectives and h fuzzy constraints (Equation 2.25) is

$$\begin{aligned} &maximize \quad \min_{i=1,\dots,k;r=1,\dots,h} \{\mu_{zi}(x), \mu_{gr}(x)\} \\ &subject\ to \quad Ax \leq b, x \geq 0 \end{aligned} \tag{2.28}$$

Or equivalent to the following crisp linear programming,

$$\begin{aligned} &maximize\ \ \lambda \\ &subject\ to\ \ \lambda \leq\ \mu_{zi}(x) \qquad i = 1,2, \ldots., k \\ &\qquad\qquad\ \ \lambda \leq\ \mu_{gr}(x) \qquad r = 1,2, \ldots., h \\ &\qquad\qquad\ \ Ax \leq b,\ x \geq 0 \end{aligned} \tag{2.29}$$

In this solution the relationship between constraints and objective functions in a fuzzy environment is fully symmetric (Zimmermann H. J., 1978). In other words, in this definition of the fuzzy decision, there is no difference between the fuzzy goals and fuzzy constraints. However, in some cases, a decision maker (DM) may want to assign different preference (importance) on each fuzzy objective and fuzzy constraints. This case can be handled by the weighted additive model which is described as follows:

The weighted additive model is widely used in vector-objective optimization problems: the basic concept is to use a single utility function to express the overall preference of DM to draw out the relative importance of criteria (Lai & Hwang, 1994). In this case, multiplying each membership function of fuzzy goals by their corresponding weights and then adding the results together obtain a linear weighted utility function.

The convex fuzzy model proposed by *Bellman and Zadeh* (1970), *Sakawa* (1993) and the weighted additive model, *Tiwari et al.* (1987) is

$$\mu_{D(x)} = \sum_{i=1}^{k} w_i \mu_{zi}(x) + \sum_{r=1}^{h} \beta_r\, \mu_{gr}(x) \tag{2.30}$$

$$\sum_{i=1}^{k} w_i + \sum_{r=1}^{h} \beta_r = 1 \qquad w_i, \beta_r \geq 0 \tag{2.31}$$

where, w_i and β_r are weighting coefficients that present the relative importance among the fuzzy goals and fuzzy constraints. The following crisp single objective programming is equivalent to the above fuzzy model:

$$maximize\ \ \sum_{i=1}^{k} w_i \lambda_i + \sum_{r=1}^{h} \beta_r \gamma_r$$

$$subject\ to$$

$$\lambda_i \leq\ \mu_{zi}(x) \qquad i = 1,2, \ldots., k$$

$$\lambda_r \leq \mu_{gr}(x) \qquad r = 1,2, \ldots., h$$

$$\lambda_i, \gamma_r \in [0,1]$$

$$w_i, \beta_r \geq 0$$

$$\sum_{i=1}^{k} w_i + \sum_{r=1}^{h} \beta_r = 1$$

$$Ax \leq b,\ x \geq 0 \qquad (2.32)$$

The Equation 2.32 is known also as *Fuzzy Additive Goal Programming*.

2.4 Genetic Algorithms

The theory of natural selection proposes that the plants and animals that exist today are the result of millions of years of adaptation to the demands of the environment. At any given time, a number of different organisms may co-exist and compete for the same resources in an ecosystem. The organisms that are most capable of acquiring resources and successfully procreating are the ones whose descendants will tend to be numerous in the future. Organisms those are less capable, for whatever reason, will tend to have few or no descendants in the future.

Evolutionary computation (EC) techniques abstract these evolutionary principles into algorithms that may be used to search for optimal solutions to a problem. In a search algorithm, a number of possible solutions to a problem are available and the task is to find the best solution possible in a fixed amount of time. In principle, the evolutionary computation encompasses three key steps in order for finding the best solution: Initialization, iterative variation and selection. The population is made to converge to optimal solution over iterations of random variation and selection.

2.4.1 Genetic Algorithm Terminology

The section discusses the basic terminology used in genetic algorithms (GA) to achieve a good enough solution for possible terminating conditions. These terminologies must be understood in order to be able to construct the appropriate genetic structure.

- *Individual* – an individual is a single solution. Individual combines two forms of solutions: (1) The chromosome, which is the raw ‘genetic’ information (genotype) that the GA deals with (2) The phenotype, which is the expressive of the chromosome in the terms of model. Figure

8 depicts the representation of genotype and phenotype. A chromosome (genotype) should in some way contain information about solution that it represents. The morphogenesis function associates each genotype with its phenotype. It simply means that each chromosome must define one unique solution, but it does not mean that each solution encoded by exactly one chromosome.

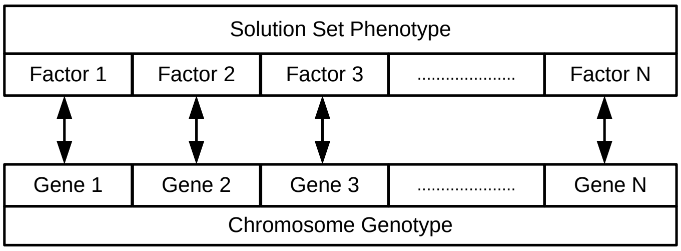


Figure 8 Representation of genotype and phenotype[9]

- *Chromosome* **–** It represents a legal solution to the problem and is subdivided into genes (Figure 9). A gene is the GA's representation of factor for a control factor. Each factor in the solution set corresponds to gene in the chromosome. Either one chromosome or a group of chromosomes can compose an individual.
- *Gene* **–** is the basic "instructions" for building Genetic Algorithms. The sequence of genes composes a chromosome. The bit string is often used to represent a gene, however, depending on the application, integers or real numbers are used.
- *Allele* **–** is possible value of a gene. For example, the Allele for a gene which is used the binary alphabet representation, is {0, 1}.
- *Genetic Locus* **-** It represents the position of a gene in chromosome.

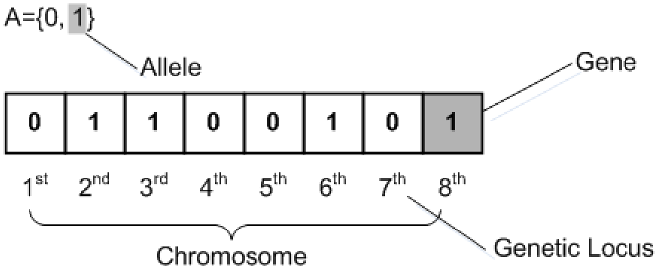


Figure 9 Representation of Chromosome, Gene, Allele and Genetic Locus[10]

- *Encoding* **-** is a process of representing individual genes. The process can be performed using bit, numbers, trees, array, list, or any other objects. The encoding depends mainly on solving the problem. For example, one can encode directly real or integer number.

[9] Source: (Sivanandam & Deepa, 2008)
[10] Source: (Soares & Vieira, 2008)

- *Fitness* **–** The fitness of an individual in a genetic algorithm is the value of an objective function for its phenotype. For calculating fitness, the chromosome has to be first decoded and the objective function has to be evaluated. The fitness not only indicates how good the solution is, but also corresponds to how close the chromosome is to the optimal one.
- *Population* **–** A population is a collection of individuals or chromosomes. A population consists of a number of individuals being tested, the phenotype parameters defining the individuals and some information about search space. For each and every problem, the population size will depend on the complexity of the problem. The larger the population size is, the easier it is to explore the search space, but it also consumes much time to converge to the optimum solution.
- *Search termination* – The iteration to search solution will be terminated based on the following parameters.
  - ✓ ***Maximum generation*** – The genetic algorithms stop when the specified numbers of generation's have evolved. Generation is a population which survives in certain period.
  - ✓ ***Elapsed Time*** – The genetic process will end when the specific time has elapsed.
  - ✓ ***No change in fitness*** – The genetic process will end if there is no change to population's best fitness for a specified number of generations.
  - ✓ ***Stall generation*** – The algorithm stops if there is no improvement in the objective function for a sequence of consecutive generation of length stall generations.
  - ✓ ***Stall time limit*** – The algorithm stops if there is no improvement in the objective function during an interval of time in seconds equal to stall time limit.

2.4.2 Breeding

The breeding process is the heart of the genetic algorithm. It is in this process, the search process creates new fitter individuals. In principle, the breeding cycle consists of three steps: (i) Selecting parents (ii) Reproducing new fitter offspring by crossover and mutation (iii) Replacing old individuals in the population with new ones.

a. Selection

Selection is the process of choosing two parents from the population for crossing. The main concern on this step is to decide how to perform selection i.e., how to choose individuals in the population that will create offspring for the next generation and how many offspring each will create. The purpose of selection is to emphasize fitter individuals in the population in hopes that their off springs have higher fitness. Chromosomes are selected from the initial population to be parents for reproduction. The problem is how to select these chromosomes. According to Darwin's theory of evolution the best ones survive to create new offspring. It implies on genetic algorithms that the chromosomes are picked out of the population according to their evaluation function. The higher the fitness function, the more chance an individual has to be selected.

Tournament is one of techniques used for selecting parents. In the tournament selection method, at each selection cycle, a number of *k* individuals are selected (having all population individuals the same probability to be chosen). The best individual from the *k* group goes on to the next phase (Michalewicz, 1996). This cycle repeats until the population is complete. The pseudo code for this method is as follows (Bäck, 1996):

```
Algorithm: Tournament
In: population P(t), nk number of individuals to be selected from P(t) to create the tournament group k
Out: P(t)' winner population from the various tournament cycles

SizeOf(P(t))' ← 0
For I ← 1 to SizeOf(P(t)) do
        k ← Random sample of nk individuals from P(t)
        bI ← Best Individual from k
        P(t)' ← P(t)' + bI
End for
Return P(t)'
```

b. Reproduction

Reproduction is a process to generate new fitter individuals for the next population. There are two basic operators in genetic algorithms to produce offspring: *crossover* and *mutation*. Crossover is supposed to exploit the current solution to find the better ones, whereas mutation is intended to maintain genetic diversity in the population. The mutation prevents the algorithm to be trapped in local minimum.

Crossover emulates the process of exchange of genetic fragment (material) among pairs of chromosomes. It takes more than one individuals (usually two) from the selected individuals (mating pool) as parent and produce a child from them. The traditional genetic algorithms uses single point crossover which cuts two chromosomes in one point and splices the two halves to create offspring (Figure 10). The dotted lines in the figures indicate the crossover point. Apart from single point crossover, many different crossover algorithms have been devised, often involving more than one cut point (Figure 10). It should be noted that adding further crossover points reduce the performance of the GA. The problem with adding additional crossover points is that building blocks are more likely to be disrupted. However, an advantage of having more crossover points is that the problem space may be searched more thoroughly.

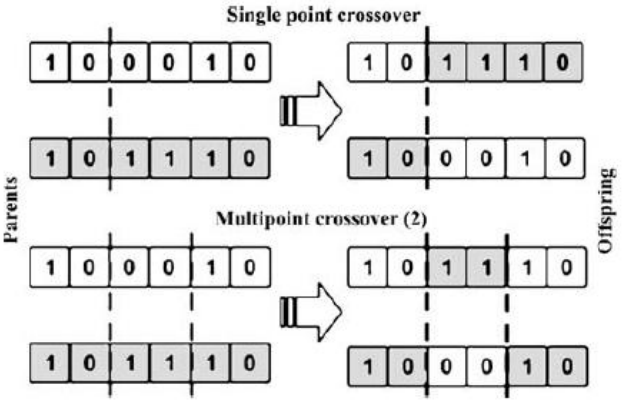


Figure 10 Crossover operation

Mutation alters one or more genes of the structure to a value randomly defined (Figure 11). This makes the possibility of reaching any point in the search space always feasible and escapes from the local optima's trap. One of mutation techniques often used is mutation probability. With mutation probability parameters (P_m), this technique determine how often parts of chromosome will be mutated. If there is no mutation, offspring are generated immediately after crossover without any change. If P_m is 100%, whole chromosome is changed, if it is 0%, nothing is changed. Mutation should not occur very often, because then GA will in fact change to random search.

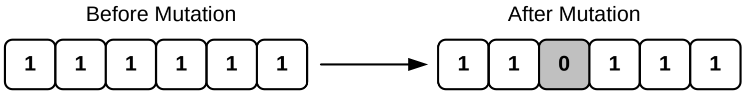


Figure 11 Mutation operation

c. Replacement

Replacement is last stage of any breeding cycle. Two parents are drawn from a fixed size population, they breed two children, but not all four can return to the population, so two must be replaced. Once off springs are produced, a method must determine which of current member of population, if any, should be replaced by new solutions. The technique used to decide which individual stay in a population and which are replaced in influencing convergence. Some of the techniques are random replacement, weak parent replacement and both parent. The first technique replaces two randomly chosen individuals in the population. In second technique, a weaker parent is replaced by a strong child. And, the last technique replaces both parents with new children.

2.4.3 Genetic Algorithms Logic

In general, the procedure of genetic algorithm is as follows: After deciding on the design of the genetic algorithm, it randomly or heuristically generates an initial population of individuals. Each individual represents a possible solution for the problem to be solved. The representation of the individual is codified through the chromosomes structure. Then each individual receives an adaptability (fitness) value calculated based on a given objective function. The fitness parameter tells us how good the solution (individual) is for the problem scenario. The next phase is to simulate the process of natural selection where the fittest individuals will have larger probability to survive and to generate offspring. In this stage, individuals with low fitness are extinguished and their places taken by the children individuals generated in the crossing, always keeping the size (number of individuals) of the population constant.

Having found the fittest individuals, the crossing (crossover) takes place, where the individuals' genetic information involved in the crossing will be combined generating offspring (children individuals). For the crossover (change of genetic material), the fittest individuals (the ones apparently generating the best solutions for the problem) have a higher chance of being selected. The mutation operator is applied to the offspring population, and, according to a very small probability, some of the new individuals will suffer mutation (a random and small change to its genetic material information). A new fitness value is calculated to the individuals that have suffered mutation. Created the descending population, the stopping criterion is checked. If a stopping criterion has been reached, the algorithm execution finishes, otherwise a new cycle begins. The pseudo code for genetic algorithm procedure is presented as follows (Bäck, 1996):

Algorithm: **genetic algorithm**

```
Generate the initial population P(0)
For each individual i in the current population P(t) do
    Evaluate individual i fitness
End for
While stop criteria not achieved do
    Select candidate individuals for crossover from population P(t)
    Apply the crossover operator in the selected individuals
    Apply the mutation operator in the Population P(t)*
        For each individual i in the current population P(t)* do
            Evaluate individual i fitness
        End for
End while
```

Chapter 3

Fuzzy Multi-Objective Decision Making for MPS Creation

Since introduced by L.A. Zadeh in 1965, the fuzzy mathematical programming has been applied in many research and practice fields including production planning and scheduling. *Lee* (1990), for example, has proposed the fuzzy set theory approach to solve the aggregate production planning (APP) problem. Moreover, *Lai et al.* (1994) also employed Carlsson & Korhonen's approach, which is extention of the fuzzy set theory, to solve production scheduling problem whose all coefficients are fuzzy. However, the application of fuzzy set theory to master production schedules problem is still uncommon. For this reason, this study proposes fuzzy multi-objective linear programming (FMOLP) to create master production schedule (MPS).

According to *Vieira et al.* (2003), manufacturing enterprises must have at least these objectives in mind, particularly, for MPS development: maximize customer service, maximize resource utilization and minimize inventory levels. In real practice, it is very hard to satisfy those objectives at once since they conflicts each other. One tries to minimize inventories, for instance, not having enough products to attend unexpected orders and ultimately may result in degradation of service level. The contrary is true; having inventory is acceptable in order to meet customer demand, however too much of it will increase costs.

Through extensive researches, various mathematical models has been developed to represent the MPS problem. One of them was presented by *Soares et al* (2008) that consider four objective measures: minimize inventory level (*EI*), requirement not met (*RNM*), inventory below safety stock (*BSS*) and overtime (*OC*). Indeed, this model has been successfully solved using several optimization techniques such as heuristic (*Vieira et al.,* 2006), simulated annealing (*Vieira et al.,* 2003), and genetic algorithms (*Soares et al.*, 2008). However, to provide alternative solution, this research resolves that mathematical model using different approach that is known as fuzzy multi-objective linear programming (FMOLP). The advantages of FMOLP compared with other previous techniques is that FMOLP:

1. Requires no min-max normalization

   Since the values of objectives (*EI, RNM, BSS, and OC*) are in completely different scale, the min-max normalization is required to put them in common scale (*Vieira et al.*, 2004). Using fuzzy multi-objective linear programming, this step can be skipped because each value of objectives is represented by the degree of membership whose interval is within range [0, 1].

2. Enable to manipulate freely objectives according to decision maker's preference

   In most multi-objective problem, the poor performance on one criterion cannot easily be balanced with a good performance on other criteria. This situation makes the DM's target difficult to be realized. By using FMOLP, the various mathematical techniques such as $\alpha - cut$, *preemptive goal, weighted additive* can be applied in order to control the objective value such that all DM's targets on each criterion can be met.

3. Considers the fuzzy objectives and productive resources

   Using FMOLP, the decision makers may simultaneously consider various conflicting objectives whose aspiration levels are imprecise. The aspiration levels of objectives ordinarily are obtained from historical data or DM's experience.

3.1 Mathematical Model of MPS

The master production schedule problem can be mathematically modeled as a mixed integer program as follows (*Soares et al., 2008*):

• K	Total quantity of different products (SKU)
• R	Total quantity of different productive resources
• P	Total number of planning periods
• TH	Total planning horizon
• OH_k	Initial available inventory (on-hand), at the first scheduling period
• GR_{kp}	Gross requirement for product k at period p
• BS_{kp}	Standard lot size for product k at period p
• NR_{kp}	Net requirement for product k at period p, considering infinity capacity
• SS_{kp}	Safety inventory level for product k at period p
• UR_{kr}	Production rate for product k at resource r (units per hour)
• AC_{rp}	Available capacity, in hours, at resource r at period p

Decision variable for the MPS are:

BN_{kpr} Quantity of standard lot sizes needed for the production of the product k at resource r, at period p (number of lots)

MPS_{kpr} Total quantity to be manufactured of the product k at resource r, at period p

$MPST_{kp}$ Total quantity to be manufactured of the product k at resource r, at period p (considering all available resources)

BI_{kp} Initial inventory level of the product k at period p

CUH_{rp} Capacity used from the resource r at period p

CUP_{rp} Percent rate obtained from the relation of the number of hours consumed from the resource r at period p, and the available number of hours to the same resource and period

GR_{kp} Gross requirement for product k at period p

RM_{kp} Total requirements met for product k at period p

RM_{kpr} Total requirements met for product k at period p at resource r

RNM_{kp} Requirements not met for product k at period p

SL_{kp} Service level, relation of the requirement met, RM_{kp} and the gross requirements for product k at period p

According to *Vieira et al.* (2003), the following objectives should be considered in the master production schedule creation.

- The overall average inventory level, considering all products and the entire planning horizon (*EI*) is

$$EI = \sum_{k=1}^{K} \left(\frac{\sum_{p=1}^{P} EI_{kp}}{TH} \right) \tag{3.1}$$

- The average requirements not met for all products and periods (*RNM*) is

$$RNM = \frac{\sum_{k=1}^{K} \sum_{p=1}^{P} RNM_{kp}}{TH} \tag{3.2}$$

- The average quantity below safety stock level, considering all products and periods (*BSS*) is

$$BSS = \frac{\sum_{k=1}^{K} \sum_{p=1}^{P} BSS_{kp}}{TH} \tag{3.3}$$

- The average over capacity needed, considering all resource and the entire planning horizon (*OC*) is

$$OC = \sum_{r=1}^{R}\sum_{p=1}^{P} OC_{rp} \tag{3.4}$$

Subject to:

$$BI_{kp} = \begin{cases} OH_k, \ se(p = 1) \\ EI_{k(p-1)}, \ se(p > 1) \end{cases} \tag{3.5}$$

$$EI_{kp} = \max\left[0, \left(MPS_{kp} + BI_{kp}\right) - GR_{kp}\right] \tag{3.6}$$

$$MPST_{kp} = \sum_{r=1}^{R} MPS_{kpr} \tag{3.7}$$

$$MPS_{kpr} = BN_{kpr} \times BS_{kpr} \tag{3.8}$$

$$RNM_{kp} = \max\left[0, \left(GR_{kp} - \left(MPST_{kp} + BI_{kp}\right)\right)\right] \tag{3.9}$$

$$BSS_{kp} = \max\left[0, \left(SS_{kp} - EI_{kp}\right)\right] \tag{3.10}$$

$$CUH_{rp} = \sum_{k=1}^{K} \frac{(BS_{kp}BN_{kp})}{UR_{kr}} \tag{3.11}$$

$$OC_{rp} = \max\left[0, \left(CUH_{rp} - AC_{rp}\right)\right] \tag{3.12}$$

3.2 Fuzzy Multi-Objective Linear Programming

Master schedule development is essentially a decision making process which can be stated as a linear programming model. It has goals to be achieved and constraint that defined the decision space. The classical model of multi-objective linear programming for MPS development can be expressed as follows.

$$Minimize\ Z_1 = \sum_{k=1}^{K}\left(\frac{\sum_{p=1}^{P} EI_{kp}(x)}{TH}\right)$$

$$Minimize\ Z_2 = \frac{\sum_{k=1}^{K}\sum_{p=1}^{P} RNM_{kp}(x)}{TH}$$

$$Minimize\ Z_3 = \frac{\sum_{k=1}^{K} \sum_{p=1}^{P} BSS_{kp}(x)}{TH}$$

$$Minimize\ Z_4 = \sum_{r=1}^{R} \sum_{p=1}^{P} OC_{rp}\ (x)$$

Subject to:

$$g(x) = CUH_{rp} - AC_{rp} \leq OL_{max}$$

$$x_{kpr} \geq 0\ and\ integer \qquad (3.13)$$

Z_1 is meant to minimize the average of inventory level, Z_2 minimize the requirement not met (maximize service level), Z_3 attempts to maintains inventory at the safety stock level, and Z_4 minimize overtime (maximize resource utilization). The deterministic constraint $g(x)$ limits overloaded capacity in each productive resource per time period. The overloaded capacity (overtime) at one productive resource must be not greater than OL_{max} (in hours).

Let assume that the decision makers want to reach some aspiration level which is not defined crisply rather than really intend to actual maximum or minimum objective function. This makes the LP-decision turn into fuzzy liner programming. *Amid et al.* (*2005)* has provided procedure to state the classical LP as a fuzzy multi-objective linear programming (FMOLP) and subsequently formulize the equivalent crisp single objective model for the FMOLP. Sequence of that procedure, which has been customized according to MPS case, is described as follows.

Step 1: Construct the fuzzy model of MPS creation problem according to the criteria and constraints of the decision makers (Equation 2.21).

Step 2: Determine the lower bound (Z_i^0) and (Z_i^1) upper bound of aspiration level (DM's target) for each objective. The limit of aspiration level (Z_i^0, Z_i^1) can be obtained by either solving multi-objective as single objective problem or retrieving from historical data (DM's experience).

Step 3: For the objective functions and fuzzy constraints, find the membership function according to Equation 2.22

Step 4: Formulate the equivalent crisp model of the fuzzy optimization problem according to Equation (2.24). For weighted additive model, assign previously DM's preference, and then use fuzzy convex decision-making to build crisp model (Equation 2.32).

Step 5: Solve the crisp model by using genetic algorithm to find the optimal solution *x*.*

Following the above steps, the formulation of fuzzy multi-objective linear programming (FMOLP) for MPS problem can be described as below.

Step 1:

If the decision maker (DM) has a fuzzy goal such as the objective function $\tilde{Z}_i(x)$ should be substantially less than or equal to some aspiration level Z_i^*, then the FMOLP for the MPS problem (Equation 3.13) can be formulated as follows.

Find $x = [x_{111}, x_{112}, x_{113}, \dots, x_{kpr}]$ to satisfy

$$\tilde{Z}_1(x) = \sum_{k=1}^{K} \left(\frac{\sum_{p=1}^{P} EI_{kp}(x)}{TH} \right) \lesssim Z_1^*$$

$$\tilde{Z}_2(x) = \frac{\sum_{k=1}^{K} \sum_{p=1}^{P} RNM_{kp}(x)}{TH} \lesssim Z_2^*$$

$$\tilde{Z}_3(x) = \frac{\sum_{k=1}^{K} \sum_{p=1}^{P} BSS_{kp}(x)}{TH} \lesssim Z_3^*$$

$$\tilde{Z}_4(x) = \sum_{r=1}^{R} \sum_{p=1}^{P} OC_{rp}\,(x) \lesssim Z_4^*$$

$subject\ to.$

$$g_2(x) = CUH_{rp} - AC_{rp} \leq OL_{max} \qquad (deterministic\ constraint)$$

$$x_{kpr} \geq 0\ and\ integer \tag{3.14}$$

Z_i^* $(i = 1,2,3,4)$ are aspiration level that a decision maker want to achieve. Sign " $\lesssim$ " on the objective function shows fuzzy inequality which has the linguistic interpretation “essentially smaller than or equal”.

Step 2 & 3:

Let assume that Z_i^0 and Z_i^1 denotes the value of the objective function $Z_i(i = 1,2,3)$ such that the degree of membership function is 0 or 1 respectively. Using non-increasing linear function (Figure 12.a), the membership function for objectives Z_1, Z_2, Z_3 are expressed as Equation 3.15. For the objective function Z_4, the triangular membership function (Figure 12.b) is used to express the degree of achievement level. If a and b denote the limit of the admissible violation of overtime objective Z_4, then the membership function of objective can be formulated as Equation 3.16.

$$\mu_{Zi}(x) = \begin{cases} 0 & Z_i(x) \geq Z_i^1 \\ (Z_i^1 - Z_i(x))/(Z_i^1 - Z_i^0) & Z_i^0 \geq Z_i(x) \geq Z_i^1 \quad ; i = 1,2,3 \\ 1 & Z_i(x) \leq Z_i^0 \end{cases} \qquad (3.15)$$

$$\mu_{Z4}(x) = \begin{cases} 1 - \left(Z_4^1 - Z_4(x)\right)/a & Z_4^1 - a \leq Z_4(x) \leq Z_4^1 \\ 1 - (Z_4(x) - Z_4^1)/b & Z_4^1 \geq Z_4(x) \geq Z_4^1 + b \\ 0 & else \end{cases} \qquad (3.16)$$

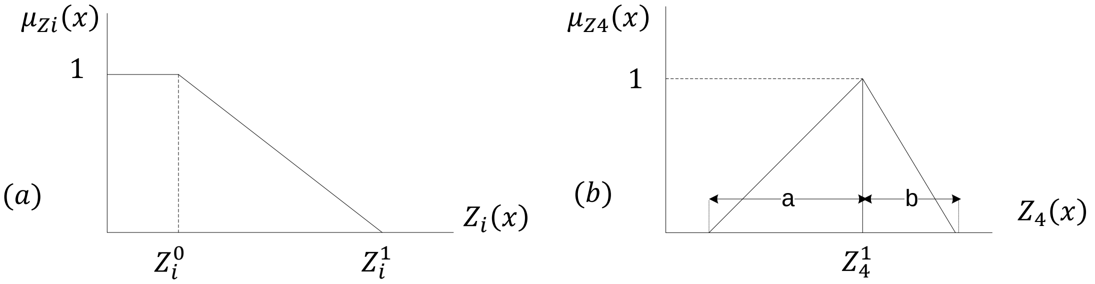


Figure 12 (a) Membership function for fuzzy objectives Z_1, Z_2, Z_3 (b) membership function for fuzzy objective Z_4

<u>Step 4:</u>

Considering that the fuzzy goals and fuzzy constraints are treated equivalently which implies that all decision criteria have equal importance. According to *Bellman and Zadeh* (1970), the equivalent crisp single objective programming for Equation (3.14) can be described as follows:

$$maximize \quad \lambda = \min\{\mu_{Z1}(x), \mu_{Z2}(x), \mu_{Z3}(x), \mu_{Z4}(x)\}$$

$$subject\ to:$$

$$\lambda \leq (Z_1^1 - Z_1(x))/(Z_1^1 - Z_1^0)$$

$$\lambda \leq (Z_2^1 - Z_2(x))/(Z_2^1 - Z_2^0)$$

$$\lambda \leq (Z_3^1 - Z_3(x))/(Z_3^1 - Z_3^0)$$

$$\lambda \leq 1 - \left(Z_4^1 - Z_4(x)\right)/a$$

$$\lambda \leq 1 - (Z_4(x) - Z_4^1)/b$$

$$CUH_{rp} - AC_{rp} \leq OL_{max}$$

$$\lambda \in [0,1]$$

$$x \geq 0 \; and \; integer \tag{3.17}$$

Step 5:

The last step is to solve the equivalent crisp model by using genetic algorithm (see Chapter 4).

3.3 Case Illustration

In order to give clear illustration about the above fuzzy models, a production scenario is presented to be solved. All numerical examples, simulation and planning in the coming chapters will refer to this production scenario as case illustration.

3.3.1 Production Scenario

Let consider a typical manufacturing system which manages four production lines (which can be machines or production lines) R_1, R_2, R_3, and R_4 and have available capacity per period 8, 8, 7 and 8 hours respectively. The manufacture system processes four products with identification number $K_1, K_2, K_3 \; and \; K_4$. The average of production rate for products at respective resources is presented by Table 2. Table 3 presents the safety stock that has to be kept at the end of each time period, the initial inventory (on-hand) for each product and standard production lot size. Table 4 depicts the gross requirement for all products within planning horizon. As shown by Figure 15, the profile of gross requirement is cyclic on which the demand is uptrend at the beginning of period, and subsequently turns down after period 8th. Inappropriate planning decision for this type of demand may lead to either excessive inventory or poor customer satisfaction. This scenario considers a MPS for a planning horizon of ten periods or equivalent to two weeks calendar.

Table 2 Average of production rate for products of respective resource (units/hour)

	R_1	R_2	R_3	R_4
K_1	400	100	150	200
K_2	150	200	400	150
K_3	200	350	150	100
K_4	400	100	300	350

Table 3 Configuration of production scenario (initial inventory/OH, safety stock, lot size)

	OH	s. stock	Lot size
K_1	440	200	50
K_2	350	180	50
K_3	220	300	50
K_4	500	250	50

All products can be manufactured in any available productive resources (production line) and afterward the finished products will be delivered to the corresponding packaging station as shown by Figure 13. It is assumed that the disturbance (machine failure, labor shortage, etc.) won occur during operation. The production rate and customer requirement are not known precisely so that the crisp value of objective function cannot be determined exactly. As a way out, the values of objective functions are defined as aspiration level, which is essentially imprecise number. Therefore, the fuzzy model must be employed instead of classical LP.

Table 4 Gross requirements (units) for all products

Prod\Period	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
K_1	1470	1300	1710	1090	1440	2000	1860	2500	1560	1140
K_2	1320	1520	1560	1660	1530	1590	2000	2460	2200	1740
K_3	1360	1430	1120	1590	1740	1680	2080	2910	2810	1710
K_4	1450	1100	1480	1410	1130	1520	1950	2120	1580	1880
Total	**5600**	**5350**	**5870**	**5750**	**5840**	**6790**	**7890**	**9990**	**8150**	**6470**

Now, one can realize how complex this MPS problem is, not only because of parameter fuzziness but also the number of permutation that should be checked out to ensure the optimum solution. Definitely, this task costs much effort and time, particularly, if it involves more products and productive resources and the problem is solved using manual spreadsheet instead.

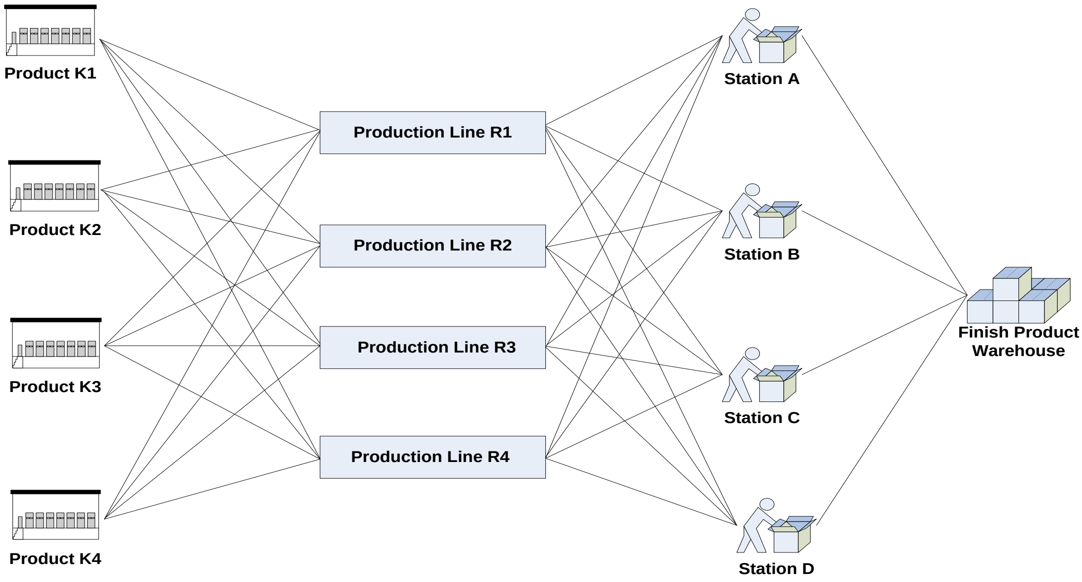


Figure 13 The material flow of manufacturing system

3.3.2 Fuzzy Solution

For evaluation, the model FMOLP will be applied to solve the given production scenario. First of all, the lower and upper bound of aspiration level for all objectives must be defined. The de-fuzzified value of inventory level, requirement not met, inventory below safety stock and overtime are presented in Table 5. It is important to note that the aspiration level must be reasonable values to avoid infeasible solution. The admissible violation parameters for overtime objective ($a\ and\ b$) are set as 6 and 3 hours respectively, whereas the maximum overtime that is allowed for each productive resource is assumed 4 hours ($OL_{max} = 4$).

Table 5 The lower and upper bound of aspiration level defined by Decision Makers

	Minimum membership degree ($\boldsymbol{\mu = 0}$)	**Maximum membership degree ($\boldsymbol{\mu = 1}$)**	**Minimum membership degree ($\boldsymbol{\mu = 0}$)**
Average of inventory level ($\boldsymbol{Z_1}$)	-	5250	50
Requirement not met ($\boldsymbol{Z_2}$)	-	3200	0
Inventory level below safety stock ($\boldsymbol{Z_3}$)	-	2000	10
Additional resource capacity/Overtime ($\boldsymbol{Z_4}$)[11]	0	6	9

Having substituted the parameter values into Equation 3.17 and assuming the objectives and constraints are treated equivalently or without relative importance, the equivalent crisp model for the MPS problem is obtained as follows:

$$maximize\quad \lambda = \min\{\mu_{Z1}(x),\ \mu_{Z2}(x)\ ,\mu_{Z3}(x),\ \mu_{Z4}(x)\}$$

$$subject\ to:$$

$$\lambda \leq (5250 - Z_1(x))/5200$$

$$\lambda \leq (3200 - Z_2(x))/5200$$

$$\lambda \leq (2000 - Z_3(x))/1990$$

$$\lambda \leq 1 - (Z_4(x) - 6)/3$$

$$\lambda \leq 1 - (6 - Z_4(x))/6$$

$$CUH_{rp} - AC_{rp} \leq 4$$

[11] a=6 and b = 3

$$\lambda \in [0,1]$$

$$x \geq 0 \; and \; integer$$

$$r = 1,2,3,4; \quad p = 1,2,\dots,10 \qquad (3.18)$$

Solving the above crisp single objective using the MATLAB's genetic algorithm toolbox, the degree satisfaction and achievement level of objectives for the optimum solution are obtained as follows.

$$\mu_D(x) = 0.774, \;\; \mu_{Z1} = 0.775, \;\; \mu_{Z2} = 0.786\,, \quad \mu_{Z3} = 0.774, \quad \mu_{Z4} = 0.855$$

$$Z_1 = 1221, \; Z_2 = 685, \; Z_3 = 459, \; Z_4 = 6.43$$

One can see that the required overtime for optimum solution is 6.43 hours, which differ from the DM's expectation ($Z_4^1 = 6$). This underlines one of fuzzy model advantages that DM must not know precisely some information, which is in this case the required productive resource, to get optimum solution. The best MPS solution (x^*) is presented in Table 6. The value "zero" indicate that there are no products to be manufactured in the related resource and at the corresponding time period.

Table 6 The best MPS solution found

Prod.	**Resource**	**Period**									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$\boldsymbol{K_1}$	R_1	350	450	1350	900	150	450	1300	400	1300	750
	R_2	250	300	50	200	250	100	0	250	50	100
	R_3	400	200	200	100	200	500	100	450	200	150
	R_4	250	150	450	100	200	0	400	450	150	150
	Total MPS	1250	1100	2050	1300	800	1050	1800	1550	1700	1150
$\boldsymbol{K_2}$	R_1	150	250	0	250	250	350	350	400	150	450
	R_2	250	200	750	650	300	400	300	250	200	450
	R_3	300	950	500	400	500	550	750	550	550	350
	R_4	250	300	250	150	450	400	550	400	450	550
	Total MPS	950	1700	1500	1450	1500	1700	1950	1600	1350	1800
$\boldsymbol{K_3}$	R_1	350	1100	200	500	350	450	250	500	700	300
	R_2	250	700	300	150	650	450	800	750	1300	400
	R_3	100	0	300	350	650	250	500	550	700	300
	R_4	300	100	300	400	300	450	150	100	250	150
	Total MPS	1000	1900	1100	1400	1950	1600	1700	1900	2950	1150
$\boldsymbol{K_4}$	R_1	2050	0	150	250	1750	100	0	250	150	500
	R_2	350	200	250	150	200	250	400	200	300	400
	R_3	250	100	300	300	100	250	300	350	250	450
	R_4	200	800	0	50	150	350	450	550	50	500
	Total MPS	2850	1100	700	750	2200	950	1150	1350	750	1850

3.3.3 Evaluation

The high varying demand often leads to the inefficient use of productive resources. This phenomenon will cause poor performance measure of manufacturing system if it is not anticipated properly. In term of master schedule, this problem is indicated by excessive inventory, poor customer service or high over-utilized productive resource. Indeed, various approaches have been employed to overcome the high customer demand variability in MPS creation. One of the methods is load leveling (Proud, 1999). Principally, this method levels some amount of demand to the previous period, in which the capacity is still available (Figure 14). However, this technique may increase inventory cost, particularly, if the demand is shifted to the very early in advance of time period.

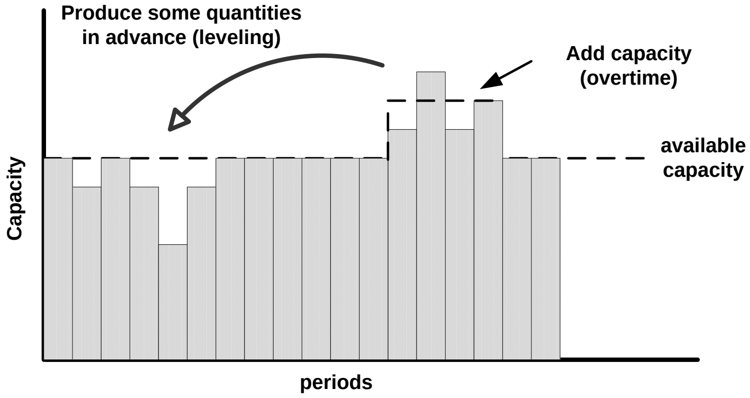


Figure 14 Overcome varying demand in MPS

Another approach, which is widely used in practical, is to add some amount of resource capacity (overtime), especially, as peak load (Figure 14). This technique is ordinarily able to reduce inventory level, but consequently it may raise production cost. The challenge of this method is to determine the required capacity needs to be added, in which periods and at which resources the additional capacity (overtime) should be assigned such that the inventory and production costs are balanced. An optimization technique for MPS creation is considered having excellent intelligence if it can overcome those problems without much scheduler's intervention.

The proper assignment of overtime (quantity and position) can definitely reduce inventory level. In contrary, the inappropriate placement of overtime will not only lead ineffective use of resource but also may cause high inventory. The ability of FMOLP model to determine and put additional resource capacity will be tested. For this purpose two model configurations are presented. At the first configuration, the decision maker configures the aspiration level for overtime objective as follows $(a = 6, Z_4^1 = 6, b = 3)$. In the second configuration, the aspiration level of overtime objective function is set to zero ($a = 0, Z_4^1 = 0, b = 1$). The outcomes of both configurations are presented in Table 7.

Table 7 The MPS solution with and without overtime assignment

	$\mu_D(x)$	μ_{Z1}	μ_{Z2}	μ_{Z3}	μ_{Z4}	EI	RNM	BSS	OC
$Z_4^1 \cong 6\ hrs$	0.77437	0.77437	0.78594	0.77437	0.85516	1221	685	459	6.4345
$Z_4^1 \cong 0\ hrs$	0.55212	0.55212	0.55344	0.70754	0.7619	2379	1429	592	0.2381

The outcome exhibits that the overall degree satisfaction can be improved considerably by adding resource capacity as around 6 hours. One can see that the inventory level (EI) and requirement not met (RNM) go down almost 100%. This underlines hypothesis that the overtime assignment can reduce the inventory level while maintaining high customer satisfaction. In addition, this result also proves that the FMOLP can put the given additional capacity on the right resource and time. As shown by Figure 15, the additional capacity is required as peak load and generally the overtime profile reflects the gross requirement pattern.

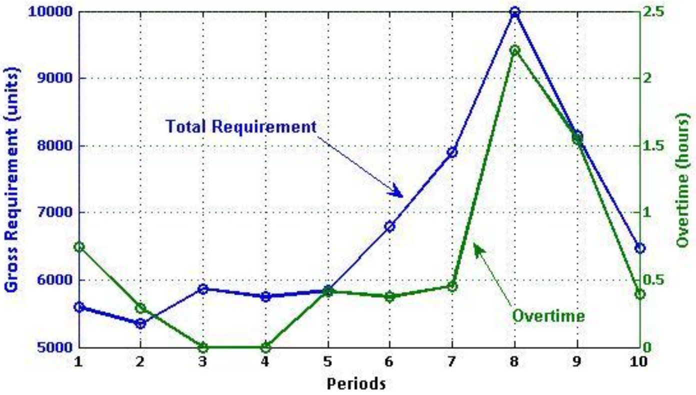


Figure 15 The profile of total gross requirement and overtime

3.4 Weighted Additive Model

In a decision making process, a decision maker may require to give different preference to the objectives functions, according to the company policy or planning strategy. For instance, the top level managements may prefer to have high service level although this costs poor performance of other criteria such as to having high inventory level. This kind of strategy can be implemented by formulizing the fuzzy weighted additive model, which is proposed by *Tiwari et al.* (1987). The model enables decision maker to assign Constanta multiplied to each objective.

To build the fuzzy weighted additive model, let assume that $w_i (i = 1,2,3,4)$ are the weighting coefficients that denote the relative importance between the fuzzy objectives. Using the convex fuzzy model proposed by *Bellman and Zadeh* (1970) and *Sakawa* (1993), the crisp single objective programming (Equation 3.18) will turn into the following forms.

$$maximize \quad w_1\lambda_1 + w_2\lambda_2 + w_3\lambda_3 + w_4\lambda_4$$

$$subject\ to:$$

$$\lambda_1 \leq (Z_1^1 - Z_1(x))/(Z_1^1 - Z_1^0)$$

$$\lambda_2 \leq (Z_2^1 - Z_2(x))/(Z_2^1 - Z_2^0)$$

$$\lambda_3 \leq (Z_3^1 - Z_3(x))/(Z_3^1 - Z_3^0)$$

$$\lambda_4 \leq 1 - (Z_4(x) - Z_4^1)/b$$

$$\lambda_4 \leq 1 - \left(Z_4^1 - Z_4(x)\right)/a$$

$$CUH_{rp} - AC_{rp} \leq OL_{max}$$

$$w_1 + w_2 + w_3 + w_4 = 1$$

$$\lambda_i \in [0,1] \quad ; i = 1,2,3,4 \quad ; r = 1,2,3,4 \ ; p = 1,2,\ldots,10$$

$$x \geq 0\ and\ integer \qquad (3.19)$$

Assuming that all objectives have equal importance $(w_1 = w_2 = w_3 = w_4 = 0.25)$, the optimum solution for the above crisp model is obtained at the degree of overall satisfaction $\mu_D(x) = 0.86278$ with the achievement level as follows.

$$\lambda_1 = 0.791,\ \lambda_2 = 0.821,\ \lambda_3 = 0.842,\ \lambda_4 = 0.998$$

$$Z_1 = 1138,\ Z_2 = 574,\ Z_3 = 325,\ Z_4 = 5.988$$

Having compared with the previous fuzzy formulation (Equation 3.18), this weighted additive model result in better performance since the degree of overall satisfaction does not simply impersonate the membership degree of the worst objective. To show the improvement of achievement level of objective functions due to the relative importance, the various configurations of weighting factor are tested and presented in Table 8. It shows that the achievement level changes proportionally to the DM's preference. The higher the weighting factor, the higher degree of achievement level is obtained. It indicates that the achievement level of objective function is consistent with DM's preference.

In addition, the result also shows that the improvement of achievement level in one objective must be balanced with poor performance on other objectives. For instance, the customer service degree as much as 99% can be achieved by maintaining very high inventory level (4630 unit/period). In contrary, having very low inventory (57 unit/period) increase requirement not met (1090 units/period). This fact also emphasizes that the master production schedule involves multi objectives which are conflicting each other.

Table 8 The achievement level with different importance

w_1	w_2	w_3	w_4	λ_1	λ_2	λ_3	λ_4	$\boldsymbol{EI}$	$\boldsymbol{RNM}$	$\boldsymbol{BSS}$	**OC**
Equal Importance											
0.25	0.25	0.25	0.25	0.791	0.821	0.842	0.998	1138	574	325	5.988
More importance to inventory level (EI)											
0.3	0.2	0.2	0.3	0.856	0.746	0.787	0.996	800	812	434	6.012
0.5	0.2	0.2	0.1	0.942	0.759	0.684	0.986	349	772	638	6.042
0.7	0.1	0.1	0.1	0.999	0.659	0.566	0.998	57	1090	873	5.988
More importance to requirement not met (RNM)											
0.2	0.3	0.3	0.2	0.693	0.829	0.885	0.997	1645	547	239	5.982
0.1	0.5	0.2	0.2	0.271	0.953	0.877	0.998	3842	149	255	6.006
0.1	0.6	0.2	0.1	0.173	0.979	0.932	0.979	4351	66	146	5.875
More importance to inventory below safety stock (BSS)											
0.2	0.2	0.4	0.2	0.671	0.822	0.879	1.000	1759	569	250	6.000
0.1	0.2	0.6	0.1	0.425	0.833	0.908	0.984	3039	534	193	6.047
0.1	0.1	0.7	0.1	0.148	0.905	0.946	0.670	4481	303	117	6.988

3.5 Sensitivity Analysis

The achievement level of objective function is influenced by the interval of aspiration level (Z_i^0 and Z_i^1). Assigning unreasonable value of aspiration level can lead to infeasible solution. The various approaches have been suggested by many researchers to ensure the feasible solution. One of them is proposed by *Zimmermann* (1978) that suggested using the individual optima as lower bound (Z_i^0) and upper bound (Z_i^1) of aspiration level for each objective. This approach has been implemented by *Amid et al.* (2005) & *Kumar et al.* (2005) to solve the supplier selection problem in fuzzy environment. They obtained the lower and upper bound through solving multi-objective problem as single objective using, each time, only one objective (ignoring all others) and satisfy the corresponding constraints.

Another approached is employed by *Roy & Maiti* (1998) in their study. They let decision makers to choose freely any aspiration values, which is ordinarily obtained from historical data or experience, such that the feasible solution can be found. To have knowledge of the behavior of variation aspiration interval on each objective, *Roy et al. (1998)* presented the sensitivity analysis. Since this study adopts similar approach to determine lower and upper limit of aspiration level, the same analysis will be also carried out. This test is also meant to ensure the range of feasible aspiration level.

In this test, the variation of aspiration interval for objective Z_1, Z_2, and Z_3 are obtained by varying incrementally the upper bound of aspiration level (Z_i^1) while keeping the lower bound of aspiration level (Z_i^0) constant. For objective Z_4, the expected overtime is set constant ($Z_1^4 = 6$) while the maximum tolerance interval is varying from 1 to 6. All input parameters of MPS do not change except the maximum acceptable violation $(Z_i^1, a,\ b; i = 1,2,3,4)$. In addition, the equal importance for all objectives and constraints $(w_1 = w_2 = w_3 = w_4 = 0.25)$ is considered. The following variables denote the maximum acceptable violation for each objective.

Z_1^1	The maximum acceptable violation of inventory level (***EI***)
Z_2^1	The maximum acceptable violation of requirement not met (***RNM***)
Z_3^1	The maximum acceptable violation of Inventory below safety stock (***BSS***)
$a = b$	The maximum tolerance interval of overtime objective

3.5.1 Sensitivity due to EI

The sensitivity analysis on fuzzy goals with equal weights due to the change in the tolerance limits on inventory level is evaluated. In this case, the value of lower bound Z_1^0 is set constant (*50 units*) and the value of upper bound Z_1^1 is varying from 4250 to 12250 incrementally in steps of 1000 units for each run. The solutions for different values of Z_1^1 are tabulated in Table 9.

Table 9 Effect of variations in Z_1^1

Z_1^1	*EI*	*RNM*	*BSS*	*OC*	$\mu_D(x)$	*Membership degree*			
						μ_{Z1}	μ_{Z2}	μ_{Z3}	μ_{Z4}
4250	1092	619	311	6.023	0.84982	0.751	0.806	0.848	0.992
5250	1138	574	325	5.988	0.86278	0.791	0.821	0.842	0.998
6250	1863	603	256	6.012	0.84789	0.707	0.811	0.876	0.996
7250	1820	544	270	6.011	0.85417	0.754	0.830	0.869	0.996
8250	2000	562	267	6.024	0.86237	0.762	0.824	0.871	0.992
9250	1842	538	260	6.077	0.87142	0.805	0.832	0.874	0.974
10250	1836	532	269	5.982	0.88138	0.824	0.833	0.869	0.997
11250	2060	454	246	6.017	0.88853	0.820	0.858	0.881	0.994
12250	2375	433	234	5.982	0.88964	0.809	0.865	0.887	0.997

It shows that the degree of achievement level of inventory level (μ_{Z1}) tends upward as the maximum acceptable violation (Z_1^1) increase. However, this does not automatically lead to the improvement of inventory, in contrary, the inventory even increase slightly. Moreover, the higher acceptable violation (Z_1^1) seems improving the performance of other objectives, especially, customer satisfaction level (RNM) and inventory target (BSS). This is reasonable since typically the customer service level can be enhanced by having high inventory level.

3.5.2 Sensitivity due to RNM

This result of sensitivity analysis on fuzzy goals with equal weights due to Requirement Not Met (*RNM*) is presented in Table 10. In this test, the value of lower bound Z_2^0 is set zero and the upper bound of fuzzy objective requirement not met (Z_2^1) is changed incrementally with step 500 units for each run. The result indicates that the variation of tolerance limit on Z_2^1 does not have any considerable change on the achievement level of all objectives although it improves slightly the overall degree of satisfaction (μ_D). In general, the values of $EI, RNM, BSS\ and\ OC$ are relatively constant (only varying in certain small range). For example, the inventory value is varying between *1026* and *1248* units.

Table 10 Effect of variations in Z_2^1

Z_2^1	*EI*	*RNM*	*BSS*	*OC*	$\mu_D(x)$	*Membership degree*			
						μ_{Z1}	μ_{Z2}	μ_{Z3}	μ_{Z4}
3200	1138	574	325	5.988	0.86278	0.791	0.821	0.842	0.998
3700	1256	622	338	6.006	0.85829	0.768	0.832	0.835	0.998
4200	1189	578	259	5.988	0.87906	0.781	0.862	0.875	0.998
4700	1246	544	260	5.988	0.88166	0.770	0.884	0.874	0.998
5200	1026	596	344	6.035	0.87949	0.812	0.885	0.832	0.988
5700	1089	643	290	6.011	0.88568	0.800	0.887	0.859	0.996
6200	1164	643	336	6.000	0.87956	0.785	0.896	0.836	1.000
6700	1248	557	289	6.053	0.88211	0.769	0.916	0.859	0.982

3.5.3 Sensitivity due to BSS

The change of achievement level of objectives function due to the variation of tolerance limit on inventory below safety stock (BSS) is evaluated in this section. As the previous sensitivity test, the lower bound of aspiration level Z_3^0 is set constant at 10 units and the maximum acceptable violation Z_3^1 of BSS is varying from 2000 to 5500 with incrementally in step of 500 units. The outcome of analysis is presented in Table 11. It exhibits that the variation Z_3^1 gives significant change on the achievement level of inventory (EI), particularly, as the value Z_3^1 reach up to 4000 units. In general, the increasing of tolerance limits on BSS gets the inventory tend to decrease, in contrary, the requirement not met (RNM) and inventory below safety stock (BSS) tend to increase. It also implies that the low inventory level lead to poor customer satisfaction. One can see that this result is opposite of the variation of inventory level (Z_1^1). This test also results in better overall degree of satisfaction (μ_D).

Table 11 Effect of variations in Z_3^1

Z_3^1	*EI*	*RNM*	*BSS*	*OC*	$\mu_D(x)$	*Membership degree*			
						μ_{Z1}	μ_{Z2}	μ_{Z3}	μ_{Z4}
2000	1138	574	325	5.988	0.86278	0.791	0.821	0.842	0.998
2500	1290	645	325	5.982	0.85762	0.761	0.798	0.873	0.997
3000	1094	597	358	5.994	0.87382	0.799	0.813	0.883	0.999
3500	1001	664	382	6.006	0.87546	0.817	0.792	0.893	0.998
4000	1013	662	443	5.970	0.87361	0.815	0.793	0.891	0.995
4500	945	659	478	6.012	0.87883	0.827	0.794	0.895	0.997
5000	832	651	486	6.029	0.88621	0.849	0.796	0.905	0.994
5500	734	693	450	6.000	0.89294	0.868	0.783	0.919	1.000
6000	705	648	489	6.023	0.89670	0.874	0.797	0.920	0.995

3.5.4 Sensitivity due to OC

The sensitivity analysis due to the change in tolerance limits on fuzzy objective overtime is also conducted and presented in Table 12. In this test, the left tolerance (a) and right tolerance (b) interval of objective (see Figure 14.b) is set according a rule, where the value of Z_4^1 is constant $(Z_4^1 = 6)$ and a is equal to b. Those variables varying from 1 to 6 with incrementally in step of 1 as well as. The sensitivity test exhibits that the variation of tolerance limit on overtime gives relatively no influence to other objectives.

Table 12 Effect of variations in a and b

$a = b$	*EI*	*RNM*	*BSS*	*OC*	$\mu_D(x)$	*Membership degree*			
						μ_{Z1}	μ_{Z2}	μ_{Z3}	μ_{Z4}
1	1221	684	381	6.029	0.83620	6.029	0.786	0.813	0.970
2	1031	680	364	6.017	0.86788	0.811	0.787	0.882	0.991
3	1052	675	395	6.018	0.86541	0.807	0.789	0.871	0.994
4	1227	522	322	6.006	0.87617	0.773	0.836	0.895	0.998
5	1138	591	414	6.029	0.86625	0.791	0.815	0.865	0.994
6	1204	668	374	6.011	0.84611	0.778	0.791	0.817	0.998

3.6 $\alpha - Cut$ Analysis

The adjustment of relative importance is probably not enough to satisfy all DM's target. For example (see Table 13), by assigning the following weighting factors ($w_1 = 0.1; w_2 = 0.6; w_3 = 0.2; w_4 = 0.1$), DM may obtain good performance on customer satisfaction (*RNM=66*) and inventory below safety stock (*BSS=146*). Unfortunately, this costs very high inventory on another site (*EI = 4351*). Decision makers may consider reducing the inventory level objective somehow. In most cases, however, it is not easy to balance the poor performance on one criterion with a good performance on other criteria.

To deal with this case, we can reformulate the FMOLP model, such that the achievement level of membership function should not be less than an allowed value. The $\alpha - cut$ approach can be utilized to ensure that the degree of achievement for any goals and fuzzy constraint should not be less than a minimum allowed value α. To do so, the weighted additive model should be modified by adding new constraints of $\lambda_i \geq \alpha$ and $\gamma_r \geq \alpha, \alpha \in [\alpha^-, \alpha^+]$ to other system constraints (Equation 3.20). For illustration, let assume that the membership degree of λ_1 must not less than $\alpha - cut$. Then, the crisp model can be formulized as follows.

$$
\begin{aligned}
&maximize \quad w_1\lambda_1 + w_2\lambda_2 + w_2\lambda_3 + w_4\lambda_4 \\
&subject\ to: \\
&\qquad \boldsymbol{\lambda_1 \geq \alpha}^{12} \\
&\qquad \lambda_1 \leq (Z_1^1 - Z_1(x))/(Z_1^1 - Z_1^0) \\
&\qquad \lambda_2 \leq (Z_2^1 - Z_2(x))/(Z_2^1 - Z_2^0) \\
&\qquad \lambda_3 \leq (Z_3^1 - Z_3(x))/(Z_3^1 - Z_3^0) \\
&\qquad \lambda_4 \leq 1 - (Z_4(x) - Z_4^1)/b \\
&\qquad \lambda_4 \leq 1 - \left(Z_4^1 - Z_4(x)\right)/a \\
&\qquad CUH_{rp} - AC_{rp} \leq OL_{max} \\
&\qquad w_1 + w_2 + w_3 + w_4 = 1 \\
&\qquad \lambda_i, \gamma_1 \in [0,1] \qquad ; i = 1,2,3,4 \\
&\qquad x \geq 0\ and\ integer \qquad (3.20)
\end{aligned}
$$

[12] This constraint restricts the corresponding membership value not less than $\boldsymbol{\alpha}$

This approach requires that the decision makers have to choose reasonable value for α to avoid getting infeasible solution (Chen S. H., 1985). For example, let take one result from Table 13. With the following DM's preference $(w_1 = 0.1; w_2 = 0.6; w_3 = 0.2; \beta_1 = 0.1)$, the achievement degree of membership function of inventory level (λ_1) is obtained as *0.173*. It implies that the value of Z_1 is *4351*, which is comparable with *50* $(\mu_{Z1}(x) = 1)$. This achievement may not be enough to satisfy decision maker (DM) in term of the inventory objective function, therefore, DM may consider improving the achievement level of this objective. In this example, α^- is 0.173 and α^+ is 0.573. The variable α is changing from α^- to α^+ in steps of 0.05. Table 13 presents all optimum solutions S1 to S9 related to $\alpha - cut$ levels.

Table 13 Optimal solution related to $\alpha - cut$ level

Parameters	*S1*	*S2*	*S3*	*S4*	*S5*	*S6*	*S7*	*S8*	*S9*
$\alpha - \mathrm{Cut}$	0.173	0.223	0.273	0.323	0.373	0.423	0.473	0.523	0.573
$\lambda_1 = \mu_{Z1}(x)$	0.173	0.228	0.280	0.331	0.377	0.424	0.473	0.527	0.576
$\lambda_2 = \mu_{Z2}(x)$	0.979	0.964	0.958	0.948	0.937	0.921	0.892	0.854	0.875
$\lambda_3 = \mu_{Z3}(x)$	0.932	0.906	0.907	0.902	0.888	0.871	0.854	0.815	0.849
$\lambda_4 = \mu_{Z1}(x)$	0.979	0.970	0.979	0.979	0.994	0.950	0.997	0.998	0.982
$Z_1 = \mathrm{EI}$	4351	4060	3794	3529	3289	3046	2789	2509	2254
$Z_2 = \mathrm{RNM}$	66	116	132	165	199	254	344	467	401
$Z_3 = \mathrm{BSS}$	146	197	194	205	232	267	301	378	310
$Z_4 = \mathrm{OC}$	5.875	6.089	5.875	5.875	6.0179	6.148	5.982	6.006	6.053

In S1 – S9, simultaneously according to DM's preference and increasing $\alpha - cut$ level from 0.173 to 0.573, it is shown by Figure 16 that as the degree of achievement level of inventory objective (μ_{EI}) increases, both of the degree of achievement level of customer service (μ_{RNM}) and inventory target (μ_{BSS}) objectives drops. It implies that performance of objective EI can be enhanced by decreasing the performance of objectives RNM and BSS.

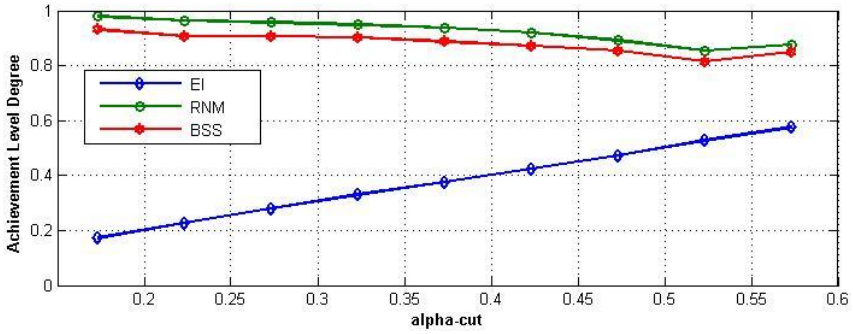


Figure 16 The achievement level with $\alpha - cut$

3.7 Comparison Fuzzy and Non-Fuzzy Solution

Vieira *et al.* (2004) and Soares *et al.* (2008) has solved the same MPS problem using simulated annealing and genetic algorithms technique respectively. To provide benchmarking, the result of Soares's study will be cited here as non-fuzzy solution. That study proposes the following equation as objective function.

$$Minimize\, Z = c_1 EI + c_2 RNM + c_3 BSS + c_4 OC \tag{3.21}$$

where, the coefficient $c_1, c_2, c_3\; and\; c_4$ are used to indicate the importance of each MPS performance measure. The objective function is meant to minimize the average of ending inventory level (EI), the requirement not met (RNM), inventory below safety stock (BSS) and over capacity (OC). Since the values of objectives goals on (3.24) have completely different ranges, min-max normalization is performed to put them on interval (0, 1). Thus, the above objective function changes into the following equation.

$$Z_n = c_1 \frac{EI}{EI_{max}} + c_2 \frac{RNM}{RNM_{max}} + c_3 \frac{BSS}{BSS_{max}} + c_3 OC \tag{3.22}$$

Where, $EI_{max}, RNM_{max}, and\; BSS_{max}$ is the maximum values of corresponding goals which are estimated from pre-processing step (warm-up period) in genetic algorithms runs. Eventually, to mimics the Darwin's natural selection theory, one considers this following equation as a measure of individual fitness. With this fitness function, the fittest individual is the one with smallest Z_n.

$$fitness = \left[\frac{1}{1 + Z_n}\right] \tag{3.23}$$

To examine the proposed genetic structure, *Soares at al.* (2008) consider a MPS creation for planning horizon of seven periods, four products and four productive resources. Each one of these resources has 35 hours of available capacity during a week. There was no initial inventory for any product and it considered 1000 units to be kept as safety inventory at the end of each period. The standard production lot size was 500 units for all products. Gross requirement for all products are shown on Table 14 and Table 15 shows productive rate for products at respective resources.

For fuzzy solution (FMOLP), the maximum acceptable violations of objectives $Z_1^1,\; Z_2^1,\; and\; Z_3^1$ are given as 15250, 9200, and 5000 units respectively and the minimum aspiration levels for them are zero ($Z_1^0 = Z_2^0 = Z_3^0 = 0$). The expected overtime is around 3 hours with lower and upper tolerance interval of constraint 3 hours ($Z_4^1 = a = b = 3$). The weighted additive model considers different importance for each objective ($w_1 = w_2 = w_3 = 0.3\ and\ w_4 = 0.1$).

Table 14 Gross requirement for benchmarking

Period	1	2	3	4	5	6	7
K_1	2700	3300	4000	4300	5300	7000	6000
K_2	4200	4500	4950	6300	6300	9050	8500
K_3	6400	6000	6400	8400	7600	8000	8000
K_4	2500	2150	1850	1600	4600	7000	8000

Table 15 Productive rates (units/hours)

	R_1	R_2	R_3	R_4
K_1	100	100	100	100
K_2	150	150	150	150
K_3	200	200	200	200
K_4	500	500	500	500

As a final result, the comparison of performance measures between fuzzy solution and non-fuzzy solution for the given production scenario is presented by Table 16. The non-fuzzy solution yields highest inventory level and overtime. It seems that this approach may be not able to assigned properly overtime ("where and when" question is not addressed accurately). Theoretically, overtime should be able to reduce inventory level if it is placed on appropriate resource and right time. In contrary, the fuzzy solution yields the lower inventory level and simultaneously keeps customer satisfaction high. The required overtime is also lower than that in non-fuzzy solution. It seems that model FMOLP can effectively address when the additional capacity must be substituted, how much it is required and at which resource should be added. The only limitation of fuzzy solution found in this test is that the inventory below safety stock (BSS) is relatively high compared with non-fuzzy solution. However, the objective BSS can be improved by using $\alpha - cut$ approach if it is necessary. Of course, it will cost the performance reduction of other criteria.

Table 16 The comparison between fuzzy and non-fuzzy solution

	EI *(units/period)*	RNM *(units/period)*	BSS *(units/period)*	OC *Hours*
Non-Fuzzy Solution	5226	986	585	4.33
FMOLP	4943	842	821	3

Chapter 4

MPS-Genetic Algorithm

The genetic algorithm is a method for solving optimization problems that based on natural selection. The genetic algorithm repeatedly modifies a population of individual selection. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them produce the children for the next generation. Over successive generations, the population evolves toward an optimal solution. The genetic algorithm can be applied to solve variety of optimization problems that are not well suited for standard optimization algorithms, including the problem in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear. In this study, the genetic algorithm is used to solve the crisp model which is equivalent to FMOLP for MPS problem (see Chapter 3). The prominent difference between genetic algorithm and standard algorithm is that the genetic algorithm (GA) uses population of solutions rather than a single solution for searching. It improves the chance of reaching the global optimum and also helps in avoiding local stationary point. In addition, the genetic algorithm generates the next population by stochastic computation that involves random choice instead of deterministic computation.

4.1 Genetic Algorithm Implementation Using MATLAB

MATLAB has variety of functions useful to the genetic algorithm practitioners and those wishing to experiment with the genetic algorithm for the first time. The genetic algorithm toolbox is collection of routines, written mostly in m-files, which implement the functions of GA.

4.1.1 Data Structure

MATLAB essentially supports only one data type, a rectangular matrix of real or complex numeric elements. The main data structure in the genetic algorithm toolbox is population, objective function and fitness value (*Sivanandam*, 2003).

Population data structure stores a collection of individuals in single matrix size $N_{ind} \times L_{ind}$, where N_{ind} the number of individuals in the population is and L_{ind} denotes the length of the genotypic representation of those individuals. Each row corresponds to an individual's genotype, consisting of base-*n*, values.

$$Population = \begin{array}{c} \left[\begin{array}{ccccc} g_{1,1} & g_{1,2} & g_{1,3} & \cdots & g_{1,Lind} \\ g_{2,1} & g_{2,2} & g_{2,3} & \cdots & g_{2,Lind} \\ g_{3,1} & g_{3,2} & g_{3,3} & \cdots & g_{3,Lind} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ g_{Nind,1} & g_{Nind,2} & g_{Nind,3} & \cdots & g_{Nind,Lind} \end{array}\right] \end{array} \begin{array}{c} Individual\ 1 \\ Individual\ 2 \\ Individual\ 3 \\ \cdots \\ Individual\ N_{ind} \end{array}$$

An individual represent single solution which consists of genotype representation (chromosome) and phenotype (see Figure 8). If the chromosome uses bit string, the decision variables (phenotypes) are obtained by applying some mapping from the chromosome representation into the decision variable space. Here, each string contained in the chromosome structure decodes to a row vector of order N_{var}. The collection of decision variables is stored in a numerical matrix of size $N_{ind} \times N_{var}$. Again, each row corresponds to a particular individual's phenotype. An example of the phenotype data structure is given below, where *BIN2REAL* is used to represent an arbitrary function, which maps the genotype to phenotype.

$$Phenotypes = \ BIN2REAL(Chromosome) \qquad \qquad \%\ genotype\ to\ phenotypes$$

$$Phenotypes = \left[\begin{array}{ccccc} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,Nvar} \\ x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,Nvar} \\ x_{3,1} & x_{3,2} & x_{3,3} & \cdots & x_{3,Nvar} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{Nind,1} & x_{Nind,2} & x_{Nind,3} & \cdots & x_{Nind,Nvar} \end{array}\right] \begin{array}{c} Individual\ 1 \\ Individual\ 2 \\ Individual\ 3 \\ \cdots \\ Individual\ N_{ind} \end{array}$$

The actual mapping between the chromosome representation and their phenotypic values depends upon the *BIN2REAL* function used. It is perfectly feasible using this representation to have vectors of decision variables in different types. For example, it is possible to mix integer, real-valued and binary decision variables in the same phenotypes data structure.

An ***objective function*** is used to evaluate the performance of phenotype in the problem domain. Objective function values can be scalar or, in the case of multi-objective problems, vector. Note that objective function is not necessarily the same as the fitness values. Objective function values are stored in a numerical matrix size of $N_{ind} \times N_{obj}$, where N_{obj} is the number of objectives. Each row corresponds to a particular individual's objective vectors.

$$Objective = OBJFUN(Phenotype) \qquad \qquad \% \, objective \, function$$

$$Objective = \begin{array}{cc} \begin{bmatrix} y_{1,1} & y_{1,2} & y_{1,3} & \cdots & y_{1,Nobj} \\ y_{2,1} & y_{2,2} & y_{2,3} & \cdots & y_{2,Nobj} \\ y_{3,1} & y_{3,2} & y_{3,3} & \cdots & y_{3,Nobj} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{Nind,1} & y_{Nind,2} & y_{Nind,3} & \cdots & y_{Nind,Nobj} \end{bmatrix} & \begin{matrix} Individual\ 1 \\ Individual\ 2 \\ Individual\ 3 \\ \cdots \\ Individual\ N_{ind} \end{matrix} \end{array}$$

Fitness values are derived from objective function value through a scaling or ranking function. Fitness is non-negative scalars and is stored in column vectors of length N_{ind}. An example is shown below where *RANKING* is an arbitrary fitness function.

$$Fitness = RANKING(Objective) \qquad \qquad \% \, fitness \, function$$

$$Objective = \begin{array}{cc} \begin{bmatrix} y_{1} \\ y_{2} \\ y_{3} \\ \cdots \\ y_{Nind} \end{bmatrix} & \begin{matrix} Individual\ 1 \\ Individual\ 2 \\ Individual\ 3 \\ \cdots \\ Individual\ N_{ind} \end{matrix} \end{array}$$

4.1.2 Toolbox Function

The genetic algorithm and direct search toolbox is a collection of function that extend the capabilities of the optimization toolbox and MATLAB numeric computing environment. This toolbox includes routines for solving optimization problems using *genetic algorithm* and *direct search*. These algorithms solve a variety of optimization problems that lie outside of the optimization toolbox.

The genetic algorithm uses three main types of rules at each step to create the next generation from the current population:

- *Selection rules* select the individuals, called parents that contribute to the population at the next generation.
- *Crossover rules* combine two parents to form children for the next generation.
- *Mutation rules* apply random changes to individual parent to form children.

At the command line, the genetic algorithm function (*ga*) can be called by using the following syntax.

$$[x \quad fval] = ga(@objfun, nvars, options)$$

Where,

- $@objfun$ is a handle to the objective function
- $nvars$ is the number of independent variables for the objective function (decision variables)
- $options$ is a structure containing options for the genetic algorithm. If there is no argument passed, *ga* uses its default options (MathWorks, 2004).

The result are given by,

- x – Point at which the final value is attained
- $fval$ – Final value of the objective function

Toolboxes are set of standard library functions, which consist of predefined algorithms. The functions which compose the genetic algorithm toolbox are presented in Table 17.

Table 17 The functions of genetic algorithm toolbox[13]

Solvers	
ga	Genetic algorithm solver
gatool	Genetic algorithm GUI
Accessing options	
gaoptimset	Create/modify a genetic algorithm options structure
gaoptimget	Get options for genetic algorithm
Fitness Scaling	
fitscalingshiftlinear	Offset and scale fitness to desired range
fitscalingprop	Proportional fitness scaling
fitscalingrank	Rank based fitness scaling
fitscalingtop	Top individuals reproduce equally
Parent Selection	
selectionremainder	Remainder stochastic sampling without replacement
selectionroulette	Choose parents using roulette wheel
selectionstochunif	Choose parents using stochastic universal sampling (SUS)
selectiontournament	Each parent is the best of a random set
selectionuniform	Choose parents at random
Crossover (recombination) Functions	
crossoverheuristic	Move from worst parent to slightly past best parent
crossoverintermediate	Weighted average of the parents
crossoverscattered	Position independent crossover function
crossoversinglepoint	Single point crossover
crossovertwopoint	Two point crossover

[13] Source: (Sivanandam, 2008)

Mutation Functions	
mutationgaussian	Gaussian mutation
mutationuniform	Uniform multi-point mutation

Plot Functions	
gaplotbestf	Plots the best score and the mean score
gaplotbestindiv	Plots the best individual in every generation as a bar plot
gaplotdistance	Averages several samples of distances between individuals
gaplotexpectation	Plots raw scores vs the expected number of offspring
gaplotgenealogy	Plot the ancestors of every individual
gaplotrange	Plots the min, mean, and max of the scores
gaplotscordiversity	Plots a histogram of this generations scores
gaplotscores	Plots the scores of every member of the population
gaplotselection	A histogram of parents
gaplotstopping	Display stopping criteria levels

Output Functions	
gaoutputgen	Displays generation number and best function value in a separate window
gaoutputoptions	Prints all of the non-default options settings

MATLAB lets users to customize the above genetic algorithm functions according to user's requirement. In this case, the functions for initial population, crossover and mutation will be customized such that they fit to MPS problems.

4.2 Genetic Structure of MPS Problem

This study adopts the genetic structure proposed by *Soares et al.* (2008). The proposed MPS-GA genetic representation considers several chromosomes in an individual. It is different from most representations found in the literature, which consider a single chromosome represented by a single bit vector structure. Figure 17 illustrate the conceptual model for the GA structure implemented for a scenario with three products, four resources, and three periods.

The proposed structure uses an alphabet to represent an individual as set of integer positive numbers. Each sphere in the structure, which assumes a possible value of the alphabet (allele), represents a gene. A set of genes makes a chromosome, which represents the distribution of quantities to be made at the various available resources for a given product at a specific time period. This way, the reference to a certain productive resource is given by the genetic locus. A set of chromosomes composing the chromosome group represents the total distribution of quantities to be made of all the products at every resource, in a given time period. The complete

MPS individual is made of a set of chromosome group. The length of this set is given by the number of time periods in the master plan horizon.

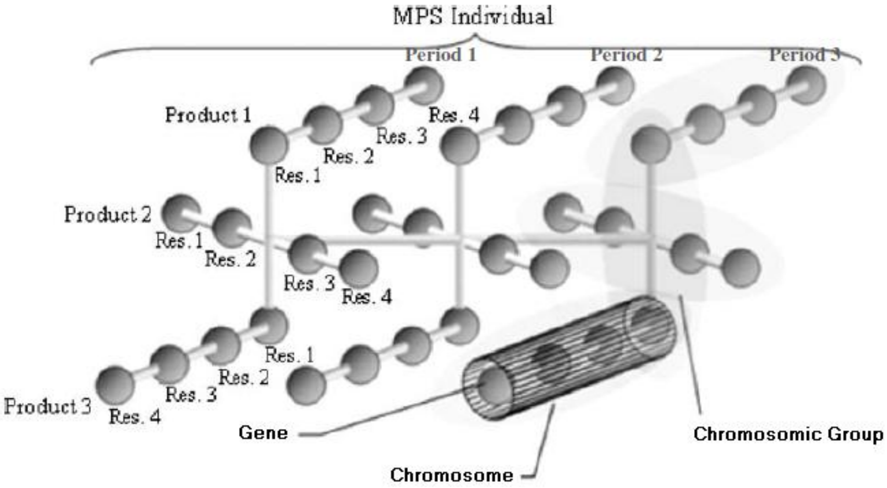


Figure 17 A conceptual genotype representation for the MPS-GA developed[14]

Since MATLAB's genetic algorithm solver accepts only a column matrix size of $1 \times N_{vars}$ as decision variables, the above 3D chromosome structure must be flattened to a 1D structure. N_{vars} is the number of gene in single MPS individual (the length of individual), which is equal to $N_{res} \times N_{prod} \times N_{per}$. The variables N_{res}, N_{prod}, $and\ N_{per}$ denote the number of resources, the number of products and the number of planning horizon respectively. Figure 18 depicts the flattened genetic structure.

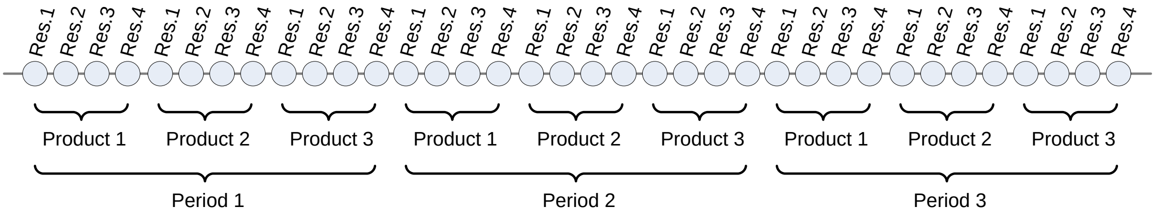


Figure 18 The 1D chromosome structure

The set of MPS individuals making a population will evolve according to the GA configuration in search for the best individual (the solution of master schedule). The MPS final solution will be represented by the decoding and extraction of information from the genotype constituting the individual phenotype.

[14] Source: (Soares & Vieira, 2008)

The phenotype – the problem solution – will have, for every product, resource and time period:

- The net requirements
- The total quantities to be made (planning/MPS quantity)
- Initial and ending inventory levels
- Requirement not met
- Service levels
- Used Capacity
- Overtime capacity needed
- Average inventory level
- Average requirement not met
- Average quantity below safety inventory level

4.3 GA's Function Customization

MATLAB's genetic algorithm toolbox provides three options as population types: *double vector, bit string, or custom.* In the MPS problem, the first two of types cannot be implemented since the chromosome representation requires integer number as allele. Therefore, the *custom* option, which let users to defined own population type, is chosen. If *custom* is selected, users must write own initial population creation, mutation, and crossover that work with the population type, and specify these function in the fields *creation function, mutation function, and crossover function*, respectively.

4.3.1 Initial Population Creation Function

The size of the population (number of individuals) and the way the initial population is created have a significant influence in the performance of the algorithm and to the quality of the results. The ideal situation would be to have the greatest possible diversity of individuals to better through the search space. Regarding to MPS genetic algorithm, the heuristic approach is considered as the best way to generate initial population (Soares & Vieira, 2008).

This approach fills up the most possible diversity, with values always respecting the standard lot size restriction. Consider a hypothetical situation where the gross requirement for a given product at a certain time period (time bucket) is 3,000 units, the standard lot size is 500 units, and there are four possible productive resources available to make the product, the alleles for the first individual in the population in the first period would be "{0;500;1,000;1500}". The second individual would have "{2,000; 3,000; 2,500; 500}", and so on, sequentially for all individuals in the population.

The pseudo code of population creation function for multi resources, multi products and multi periods may be written as follows.

Algorithm: **Customized Function for Initial Population Creation (*CustCreation*)**
In: Size of population (N_{ind}) and Gross requirement (*GR)*
Out: Initial population (*P(t))*

```
SizeOf($P(t)$) ← 0
$PH$ ← the number of planning horizon
$K$ ← the number of products
$R$ ← the number of resources
$LS$ ← production lot-size
For $p$ ← 1 to $PH$ do
    For $k$ ← 1 to $K$ do
        $GR(p, k)$ ← gross requirement of product $k$ at period $p$
        $\chi \in [0, 1, 2, \dots, round(GR(p, k)/LS)]$
        $IP(p, k)$ ← $RandInt(N_{ind}, R, \chi)$
        $P(t)$ ← $P(t) + IP(p, k)$
    End for
End for
Return $P(t) * LS$
```

χ denotes a uniform random integer variable respecting the standard production lot-size. *IP(p, k)* is output of function *RandInt* which represents the initial population corresponding to product k and period p. *RandInt* generates random integer with size of $N_{ind} \times R$ and the number must be member of χ.

4.3.2 Mutation Function

The mutation operator of genetic algorithm was introduced by Holland as a "background operator" that occasionally changes single bits (gene) of individuals by inverting them (Holland, 1975). In analogy to the natural model, the mutation probability $P_m \in [0, 1]$ per bit is usually very small in genetic algorithms. Some common settings are $P_m = 0.001$ (De Jong, 1975), $P_m = 0.01$ (Grefenstette, 1986), and $P_m \in [0.005, 0.01]$ (*Schaffer et al.,* 1989). In the latter study, *Bäck* (1996) additionally formulate heuristic rules indicating that performance is decreasing both for large population size ($\mu > 200$) combined with large mutation probability ($P_m > 0.05$) as well as for small population size ($\mu < 200$) combined with small mutation probability ($P_m < 0.002$).

For MPS genetic algorithm, the appropriate combination of populatio size ($\mu > 100$) and mutation probability $0.001 \leq P_m \leq 0.01$ is chosen (*Soares et al.*, 2008). To generate new offspring, one or more genes of the structure is altered by uniform random integer number respecting to the production lot-size. The new offspring will be evaluated whether it satisfy constraints or not. The parent individual will survive in the next generation if the offspring lies outside of constraint space. As implementation, the default mutation function in MATLAB's genetic algorithm is customized with the following algorithm. *sP* denotes a mutated individual and *ReplaceGen* is a function to replace genes of selected genetic locus with random integer member of *X*.

Algorithm: **Customized Mutation Function (*CustMutation*)**
In: parent individuals (pI)
Out: mutation children (cI)

```
$P_m$ ← 0.01
SizeOf($cI$) ← 0
$X$ ← set of uniform random integer respecting to production lot-size
$Nvar$ ← the number of decision variables
For $i$ ← 1 to SizeOf(pI) do
    $sP$ ← $pI$(i)
    $\delta \in [0, 1]$ ← set of uniform random number length of $Nvar$
    mPoints ← genetic locus of $\delta$ whose gene less than $P_m$
    If SizeOf(mPoints)>0 Then
        sP(mPoints) ← ReplaceGen($X$)
        If IsFeasible($sP$) Then cI(i) = sP
                      Else cI(i) = pI(i)
        End If
    End If
End for
Return cI
```

4.3.3 Crossover Function

While mutation in genetic algorithms serves as operator to reintroduce "lost alleles" into the population, the crossover operator is emphasized as the most important search operator of genetic algorithm. The idea forming the background of crossover is that useful segments of different parents should be combined in order to yield new individual which benefits from advantageous gene combinations of both parents. This way, longer and longer segments of high fitness are expected to emerge, finally leading to good overall solution (Holland, 1975).

Crossover in genetic algorithm is always a sexual operator that with probability p_c selects two parent individuals form the population, recombines them to form two new individuals, and discard one of the results at random. Common proposed settings for the crossover probability are $p_c = 0.6$ (De Jong, 1975), $p_c = 0.95$ (Grefenstette, 1986), and $p_c \in [0.75, 0.95]$ (*Schaffer et al., 1989*).

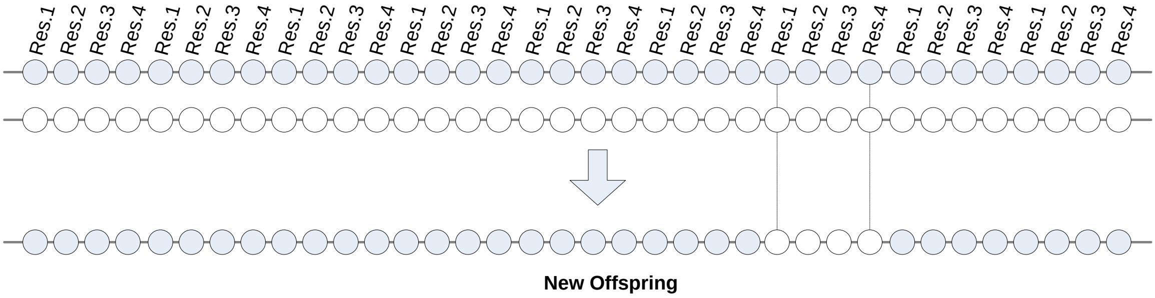


Figure 19 The example of two-points crossover for MPS genetic algorithm

In case of MPS genetic algorithm, *Soares et al.* (2008) has evaluated various configuration of genetic algorithm and suggested to have crossover probability $0.6 \le p_c \le 0.99$ and to apply crossover of just one chromose in one group of an individual (Figure 19). It can be shown that this is a particular case of a two points crossover in the 1D flattened structure. Two individuals are selected randomly to produce new feasible offspring. If the offspring does not satisfy constraint, the parent individuals will not be replaced. The algorithm of crossover operator for MPS genetic algorithm is presented below.

Algorithm: **Customized Crossover Function (*CustCrossover*)**
In: selected parent individuals (pI)
Out: new children (cI)

```
$p_c$ ← 0.95
SizeOf($cI$) ← 0
$N_{var}$ ← the number of decision variables
$R$ ← the number of resources
For i ← 1 to SizeOf(pI)/2 step 2 do
    parent1 ← pI(i)
    parent2 ← pI(i+1)
    $\tau \in [1,2,\ldots,N_{var}]$ ← first genetic locus of selected chromosome in parent1
    newOffspring ← parent1($1:\tau-1$)+ parent2($\tau:\tau+R$)+ parent1($\tau+R+1:end$)
        If IsFeasible(newOffspring) Then
            cI(i) = newOffspring
        End If
    End If
End for
Return cI
```

4.4 MPS-GA Evaluation

To give illustration of the implementation of MPS genetic algorithm, the crisp linear programming equivalency of FMOLP (Equation 3.19) will be solved. The goal of equation 3.19 is to maximize the intersection between membership function of objectives $\mu_D(x)$. Since the Genetic Algorithm and Direct Search Toolbox only minimize the objective or fitness function, one can minimize $\{-\mu_D(x)\}$ to obtain maximum optimum solution. In other word, the optimum solution is one with lowest the degree of overall satisfaction instead of highest one. Assuming DM's preference $(w_1 = w_2 = w_3 = 0.3; \beta_1 = 0.1)$, the objective or fitness function for genetic algorithm may be written in m-file as below. It is imperative to note that the objective function returns negative value.

```
$function\ f_{opt} = \boldsymbol{objFunc(x)}$
    $\mu_{Z1}(x) = \cdots$  $\%membership\ function\ of\ objective\ Z_1$
    $\mu_{Z2}(x) = \cdots$  $\%membership\ function\ of\ objective\ Z_2$
    $\mu_{Z3}(x) = \cdots$  $\%membership\ function\ of\ objective\ Z_3$
    $\mu_{g1}(x) = \cdots$  $\%membership\ function\ of\ constraint\ g_1$
    $f_{opt} = -(0.3 * \mu_{Z1}(x) + 0.3 * \mu_{Z2}(x) + 0.3 * \mu_{Z3}(x) + 0.1 * \mu_{g1}(x))$
$End$
```

Before parent selection process, the algorithm scores each member of the current population by computing its fitness value and subsequently scales the raw fitness scores into a more usable range of values. The selection function assigns a higher probability of selection to individuals with higher scaled fitness values. In this case, *Rank* function is selected to scale fitness value. In principle, the fitness scaling function, *Rank*, scales the raw fitness scores based on the rank of each individual instead of its score. The rank of an individual is its position in the sorted scores: the rank of the fittest individuals is 1, the next fittest is 2, and so on. The rank scaling function assigns scaled value of an individual with rank *n* according to formula $1/\sqrt{n}$.

In addition, elitism strategy is also applied in the MPS genetic algorithm. The elitism strategy specifies the number of individuals that are guaranteed to survive to the next generation. The number of elite children should be integer and less than or equal to population size. For MPS problem, the elite count is defined as one individual.

Once the fitness function has been saved as m-files, the genetic algorithm solver can be invoked. The input parameter values of genetic algorithm for solving the given MPS problem are presented in Table 18. The obtained solution from genetic algorithm for five time running is logged in Table 19. The solutions are different for each running due to the randomness of genetic algorithm operator (selection, crossover and mutation).

Table 18 The GA configuration parameter values to MPS problem

Parameters	Adjustment Value
Fitness function	*@objFunc*
Number of variable	160
Population size	> 160
Mutation rate(P_m)	0.01
Crossover rate (p_c)	0.95
Selection method	Tournament – using group of 4
Elitism strategy	Yes, maintaining only the best individual from the current generation
Fitness scaling	Rank based fitness scaling
Population type	double vector
Creation function	Custom Function <*CustCreation*>
Crossover function	Custom Function <*CustCrossover*>
Mutation function	Custom Function <*CustMutation*>
Plot function	• Plots the best score and the mean score <*gaplotbestf*> • Plots the min, mean, and max of the scores <*gaplotrange*> • Averages of distances between individuals <*gaplotdistance*>
Population diversity control	No need for MPS problem
Stopping criteria	• generation = 600 • stall generation = 100 • stall time limit = 100 • Time limit = infinite

Table 19 Solution by genetic algorithm

Running	μ_{Z1}	μ_{Z2}	μ_{Z3}	μ_{Z1}	Overall Degree (μ_D)	Running Time (s)
1	0.7525	0.9728	0.9387	0.9960	0.89880	238 (3.97 min)
2	0.7452	0.9984	0.9366	0.9881	0.90290	239 (3.98 min)
3	0.7298	0.9718	0.9206	0.9941	0.88609	237 (3.95 min)
4	0.7111	0.9775	0.9221	0.9921	0.88244	237 (3.95 min)
5	0.7173	0.9791	0.9422	0.9921	0.89078	239 (3.98 min)
6	0.7228	0.9637	0.9055	0.9921	0.87686	240 (4.00 min)

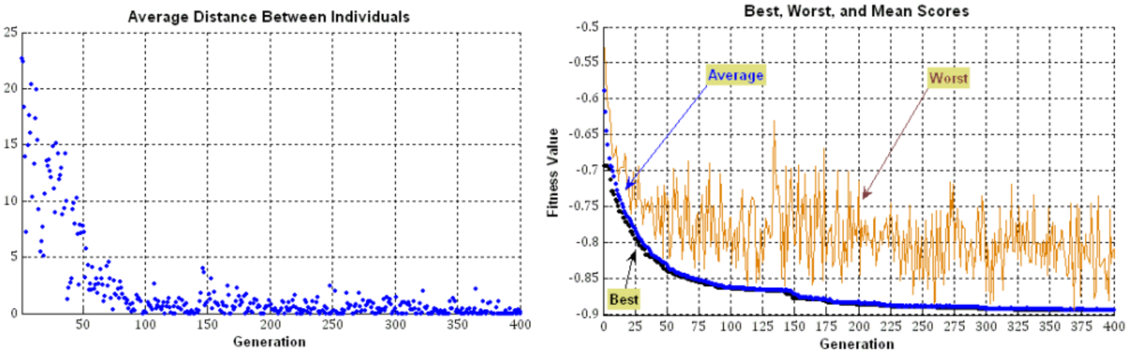


Figure 20 The example of average distance between individuals (left). The example of fitness value evolution (right)

The Figure 20(left) shows that average distance between individuals in one population decrease approaching to zero as the generation growths. It indicates that as generation evolves the overall population converges to optimum solution as well. The plot on Figure 20 (right) depicts the fitness evolution for the best individual, the average from all individuals, and the fitness of the worst solution (individual) in the population.

One can see that the average fitness for the population was very close to the best individual fitness, this indicates that may be the population has individuals that were too similar (low diversity). If this occurs, the use of higher mutation rate and/or extermination strategy could improve the quality of the solution found. According to *Chambers* (1995), the loss of population diversity can be partially decreased by the application of a re-initialization strategy, that extermination is one of variant of this strategy. Extermination basis on the fact that natural catastrophes happen and when they occur only the fittest individuals survive. In regarding to a population processed by a GA, when an extermination event occurs, only the fittest individual survives.

4.5 Benchmarking

To ensure the reliability and efficiency of the proposed genetic structure, the result of algorithm will be compared with Differential Evolution Algorithm (DE), which is another variant of Evolutionary algorithms (Bäck, 1996). The reliability means that the point with minimal/maximal function value found in search process is sufficiently close to the global minimum minimum/maximum point. The efficiency refers to the processing time required by algorithm to find global solution. Although this benchmarking cannot ensure that the obtained solutions of both algorithms are definitely global optimum, but at least, it can verify that the proposed genetic algorithm can provide relatively better solution than similar approach.

4.5.1 Differential Evolution Algorithm (DE)

In general, the differential evolution imitates also evolution theory, where the fittest individual survives and the rest extinct, but, DE is much simpler than genetic algorithm in term of reproduction. The differential evolution algorithm works with two populations P and Q of the same size N. Each individual (x_i) in old population P produces a mutated offspring (y) with mutant factor (u). The fitness value of parent individual $f(y)$ and new offspring $f(x_i)$ is compared. If the offspring is more fitness than parent individual and meet the given constraints, the offspring (y) is kept and inserted into the new population Q instead of (x_i). After completion of the new population Q, all individuals in old population P is replaced by Q and the search continues until stopping condition is met. The logic of differential evolution algorithm may be written as follows.

Algorithm: **Differential Evolution Algorithm**

```
1.  $P$ ← Initial Population consist of N individuals
2.  SizeOf($Q$) ← 0
3.  Repeat
4.          For $i$← 1 to SizeOf($P$) do
5.                  $x_i$ ← $P(i)$
6.                  compute a mutant vector $u$
7.                  create offspring $y$ by the crossover of $u$ and parent individual $x_i$
8.                  If not IsFeasible($y$) then $y = x_i$ End if
9.                  If $x_{new}$ more fit then insert $x_{new}$ into $Q$
10.                             Else insert $x_i$ into $Q$
11.                 End if
12.         End for
13.         P ←Q
14. Until stopping condition
```

There are several variants how to generate the mutant factor (u). In this case, two methods are introduced: (1) *Rand* method which generates u by adding the weighting different of two points (Equation 4.1) and (2) *Best* method that generates mutant factor u according to Equation 4.2.

$$u = r_1 + F(r_2 + r_3) \tag{4.1}$$

$$u = x_{fit} + F(r_1 + r_2 - r_3 - r_4) \tag{4.2}$$

Where, r_1, r_2, r_3 and r_4 are arbitrary points taken randomly from P (not coinciding with the current x_i), x_{fit} denote the fittest individual in population P and F is input parameter defined by decision makers. The element of new offspring $(y_j, j = 1,2,3,..d)$ is built up by the crossover of its parent x_i and u using the following rule.

$$y_j = \begin{cases} u_j & if\ U_j \le C \quad or\ j = l \\ x_{ij} & if\ U_j > C \quad and\ j \ne l \end{cases} \tag{4.3}$$

Where, $l \in j$ and $U_j \in [0,1]$ is uniform random number, and $C \in [0,1]$ is input parameters influencing the number to be exchanged by crossover. Equation 4.3 ensures that at least one element of x_i is changed even if $C = 0$.

The different value of input parameters $(F\ and\ C)$ are used in statement on line 6 and 7 of algorithm. The value of F and C is defined by control parameter values (h), which is chosen randomly with probability $q_h, h = 1,2,\dots,H$. The probability can be changed according to success rate of the setting in preceding steps of search process. The h setting is successful if the generated offspring is more fitness than its parent. If n_h is the current number of the h setting successes, the probability q_h can be evaluated simply as the relatively frequency

$$q_h = \frac{n_h + n_0}{\sum_{k=1}^{H}(n_k + n_0)} \tag{4.4}$$

where, $n_0 > 0$ is a constant. The setting $n_0 \geq 1$ prevents a dramatic change in q_h by one random successful use of the h parameter setting. In order to avoid the degeneration of process the current value of q_h are reset to their starting values $q_h = 1/H$ if any probability q_h decrease below a given limit $\delta > 0$. The competition provides a self-adaptive mechanism of setting control parameter appropriate to the problem actually solved.

4.5.2 DE Evaluation

This algorithm was tested on the crisp linear programming (Equation 3.19) with relative importance $(w_1 = w_2 = w_3 = 0.3; \beta_1 = 0.1)$. Since it is a global minimization, the fitness function will be $\{-\mu_D(x)\}$ instead of $\{\mu_D(x)\}$. The same genetic structure (Figure 18) is employed in this algorithm. The parameters values that are controlling competition of the setting are set up as follows:

- Competition control – $n_0 = 2;\ \delta = {}^{1}/_{(5H)};\ H = 5$
- Population size – $N = 50$
- The number of decision variables – $d = 160$
- Maximum objective function evaluation – $max\_evals = 3000$
- The tolerance of convergence – $max\_eps = 1e - 5$

The input parameters (*F and* C) and mutant factor generator (*u*) is determined based on the control parameters h (Table 20). For example, $(F = 0.5; C = 0$ for $h = 1)$, $(F = 0.5; C = 0.5$ for $h = 2)$ and so on. The mutant vector *u* is generated according to equation 4.1 in case *h* values are obtained as 1 – 9 and equation 4.2 if *h* values are within range 10 – 18.

Table 20 Input parameters according to control setting

h	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
F	0.5	0.5	0.5	0.8	0.8	0.8	1	1	1	0.5	0.5	0.5	0.8	0.8	0.8	1	1	1
C	0	0.5	1	0	0.5	1	0	0.5	1	0	0.5	1	0	0.5	1	0	0.5	1

The search for the global optimum was stopped if the following condition is met.

1. The number of objective function evaluations exceeds the upper limit ($max\_evals \times d$).
2. The difference of best and worst fitness values less than the tolerance of convergence ($f_{max} - f_{min} < max\_eps$).
3. Stall generation, which is the number of generation without improvement, is met ($s\_gen = 300$).

Table 21 shows the solution of equation 3.19 obtained from the differential evolution algorithm for six times running. Likely genetic algorithm, the differential evolution gives also different solution for each run due to randomness. To reproduce the same result, one can reset the state of random number generator. Figure 21 depicts the example evolution of the best, mean and worst fitness value corresponding to generation.

Table 21 Solution by differential evolution algorithm

Running	μ_{Z1}	μ_{Z2}	μ_{Z3}	μ_{g1}	**Overall Degree** (μ_D)	**Running Time (second)**
1	0.8063	0.9872	0.9332	0.9921	0.91722	673 (11.2 min)
2	0.7717	0.9947	0.9553	0.9960	0.91611	456 (7.6 min)
3	0.8117	0.9912	0.9367	0.9841	0.92031	506 (8.4 min)
4	0.7785	0.9966	0.9392	0.9980	0.91407	478 (7.7 min)
5	0.7994	0.9775	0.9357	0.9960	0.91338	446 (7.4 min)
6	0.8023	0.9903	0.9377	0.9980	0.91889	588 (9.8 min)

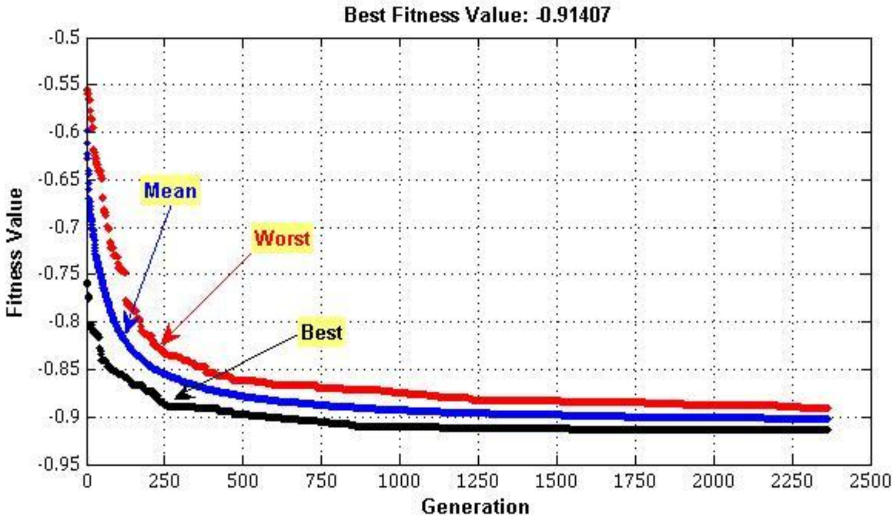


Figure 21 The example of fitness values evolution

4.5.3 Comparison GA and DE

Although both algorithms are inspired by the same idea, that is Darwin's evolution theory, they show completely different behavior, particularly, in processing time. One can see (Table 21) that the differential evolution requires double processing time or even more than that of the genetic algorithm (Table 19) to find optimum solution. It means that in this case the GA is much more efficient than DE algorithm. In addition, the genetic algorithm shows good stability in term of processing time, which is indicated by lower standard deviation (Table 22). For six times run, the processing time required by GA is only varying within 3.95 and 4.00 minutes. In contrary, the processing time of DE algorithm varies in large range. One times the DE algorithm converges to optimum solution in 7.4 minutes, but, in another time, it requires 11.2 minutes to reach the global solution.

Despite its inferiority in term of efficiency, Table 22 shows that the DE algorithm can achieve higher average of the best fitness value. It means that the DE is more reliable than genetic algorithm. Another thing which makes DE algorithm superior is that, in spite of less population size, it likely covers larger searching space so that the better fitness value can be found. It's reasonable because the DE algorithm has evolved as 2350 generations before it converge to optimum solution. It is almost five times compared with generation evaluated by GA.

However, this study prefers to use genetic algorithm because of its rate and stability. In other applications or real practices, in which time is not crucial factor, one may consider implementing the DE algorithm to solve the MPS problem.

Table 22 Comparison of optimum solutions of GA and DE

Performance Measure	GA	DE
Average of the best fitness value	0.88964	0.91666
Standard deviation of the best fitness value	0.0099	0.0027
Average processing time	3.97 minutes	8.75 minutes
Standard deviation of processing time	0.02	1.47
Population size	160	50

Chapter 5

Master Schedule Frameworks

In “top-down” manufacturing system, master production schedule (MPS) is a vital function of production planning. MPS is a device for top management to control over manufacturing resources. By altering the MPS, management can alter inventory levels, lead-times, capacity demand, and so on. To manage inventory and production properly, a *valid* and *realistic* MPS is required (Sheikh, 2003). *Valid* means that resources needed for execution will be available. In other word, the material due date equal to the material need dates and the planned capacity equal to the required capacity (Proud, 1999). *Realistic* means that they satisfy DM’s target or management’s strategies and policies.

In integrated manufacturing system, MPS incorporating with other functions composes production planning and control (Figure 3). The terminology of MPS framework in this study refers to the picture (Sari, 1997). The tight cooperation of inter-function is presented in this framework, where output of one planning function drives downstream planning functions. For instance, the productive resource profile estimated by rough-cut capacity planning (RCCP) becomes one of MPS input parameters, and subsequently the MPS output drives material requirement planning (MRP), and so on. This type of structure makes the performance of one function induces other ones. Therefore, the development of *valid* and *realistic* MPS is not merely about MPS optimization technique but also related to the surrounding functions, which is affecting and affected by MPS, as an integrated system. Moreover, better understanding for actual shop-floor situation such as the variability of production parameters (processing time, set-up time, and queue time) is also essential for development of valid and realistic MPS.

Based on the thought, a development methodology of valid and realistic MPS, which is called “Closed-Loop MPS”, is proposed. The closed-loop MPS transplants the fuzzy approach (Chapter 2) into the real MPS framework. Once can say that this method “disarms” the top-down management approach which is prejudged as one of root causes of MPS unrealism due to lack of detail information on shop-floor level. In top down system, downstream functions are unable to fix the deficiency of their upstream function. In contrary, this approach allows lower level planning (execution) gives feedback to MPS in case of infeasible implementation.

5.1 Closed-Loop MPS

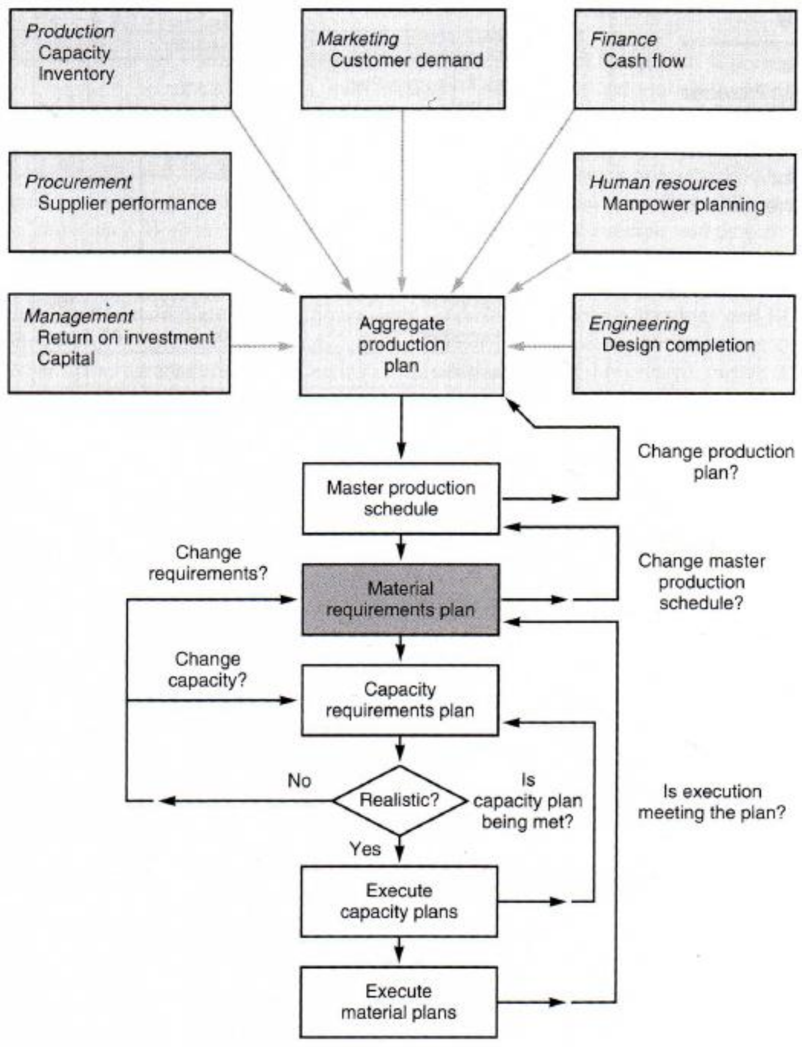


Figure 22 Iterative planning process[15]

The creation of an accurate and realistic master production schedule is essential for all types of manufacturing system. Since the MPS is the driver of all the production planning in downstream level. According to *Sheikh* (2003), the development of actual scheduling is usually iterative, with a preliminary schedule being drawn up, checked for problems, and revised them. After a schedule has been determined, the following points are checked:

1. Does the schedule meet the production plan and feasible to implement?

[15] Source: (Heizer & Render, 2006)

2. Does the schedule meet the end item demand forecast?
3. Does the schedule provide for flexibility and backups in case disturbance occurs?
4. Does the schedule violate any other constraints regarding equipment, lead-times, supplier, facilities, and so forth? And so on.
5. Does the schedule conform to organizational policy?

Problem in any one of these areas may force a revision of the schedule and a repeat of the previous step. The iterative process continues until all questions are answered and the planning goal is met. The same though has been also shared by *Heizer et. al* (2006), which proposed a concept of how valid and realistic master production schedule can be generated through iterative process (Figure 22). First of all, the master schedule must be in accordance with a production plan. The production plan sets the overall level of output in broad terms (for example, product families, standard hours, or dollar volume). The production plan also includes a variety of inputs, including fluctuations, supplier performance, and other considerations. Each of these inputs contributes in its own way to the production plan as shown in Figure 22.

As the planning process moves from the production plan to execution, each of the lower-level plan must be feasible. When one is not, feedback to the next higher level is used to make the necessary adjustment. One of the major strength of MRP is its ability to determine precisely the feasibility of a schedule within aggregate capacity constraints. However,the verification of production planning is still required, particularly, before execution. This step will ensure that no infeasible production planning is released to shop-floor (manufacturing system). Unfortunately, *Heizer et. al* (2006) did not go detail of how to validate the production plan on execution level.

According to *Kochhar et al.* (1998), there are some major difficulties associated with the creation of an valid and realistic MPS. Those difficulties are briefly discussed as below.

1. *Invalid Data*. The validity of an MPS can be considerably affected by any errors in the input parameters such as inaccurate forecasting, error in inventory data, and on-hand balance.
2. *Overstatement*. Overstating an MPS result in a large quantity of “past due” (*Berry et al.,* 1987). As nothing can be made yesterday, the past due quantities become a problem. For any valid and realistic MPS, all these past due quantities must be eliminated.
3. *Uncertainty*. Frequent changes made to an MPS can result in an unstable and uncertain master schedule. However, it is sometimes necessary to change an MPS due to a change in customer demand. The effect of these change must be analyzed in order to determine the best of accomodating them (*Berry et al.,* 1987).

4. *Execution*. An MPS is a plan to make an amount of product in a given time period. Real benefits only come when the plan is executed well (*Plossl et al.*, 1983). It is often the case that the execution of an MPS cannot be carried out according to plan for one reason or another such as machine failure, blocked buffer or high variability of processing time.
5. *Lack of on-time decision making*. Since the MPS is considered the backbone of a production planning system, it is the responsibility of the master scheduler to take the right actions at the right time to overcome the problems that may occur during the master production scheduling process.

To overcome the potential problems, *Kochhar et al.* (2008) proposed a method called "knowledge-based system". In principle, the knowledge-based system approach adopts also iterative process. This approach integrates the capabilities of knowledge-based systems and what-if analysis. The knowlegde based contains facts and data concerning a given manufacturing system as well as rules for making appropriate management decisions. For a given MPS the knowledge-based system interprets the results of the what-if analysis. If there are problems that make the MPS unrealistic, the knowledge-based system search for feasible solutions and recommends them for the scheduler to consider. Following the selection of solutions to the identified problems, the what-if simulation can be carried out again to analyze the effectiveness of these solutions before implementation.

By combining a knowledge-based system with what-if simulation facilities, almost all potential problems described above can be resolved. This approach can successfully analyze the effect of adding new product, adding new orders for existing products, off-load some orders and push or pull due date on tentative MPS. Subsequently, it suggests such solutions that MPS is feasible to implement. Nevertheless, the aprroach overlooks the effect of unpredictable event (i.e. machine failure, varying production parameters) that may occurs as execution. In fact, the uncertainty can disturb MPS and make it no longer realistic. Therefore, in order to bridge the gap of existing research, this study uses discrete event simulation (DES) to help create an valid and accurate MPS. Using discrete event simulation, one can simulate unpredictable event (uncertainty) or dinamic behaviour of manufacturing system and analyze its effect on the given MPS.

Figure 23 depicts a methodology for creation a valid and realistic MPS proposed in this study. This methodology is called also "Closed-Loop MPS" since it involves iterative process to obtain a valid and realistic MPS. Based on the input parameters retrieved from Sales & Operation Planning (SOP) and Demand Management, MPS generator proposes a tentative MPS solution. Using fuzzy multi-objective linear programming (Chapter 3), the MPS generator considers four objectives: minimizing inventory level, maximizing customer satisfaction, minimizing inventory below safety stock and maximizing use of resources. A scheduler (decision maker) may set such optimization parameters that the DM's target can be achieved. The tentative MPS drives MRP I to determine the required material based on bill of material (BOM), on-hand inventory and scheduled receipts information.

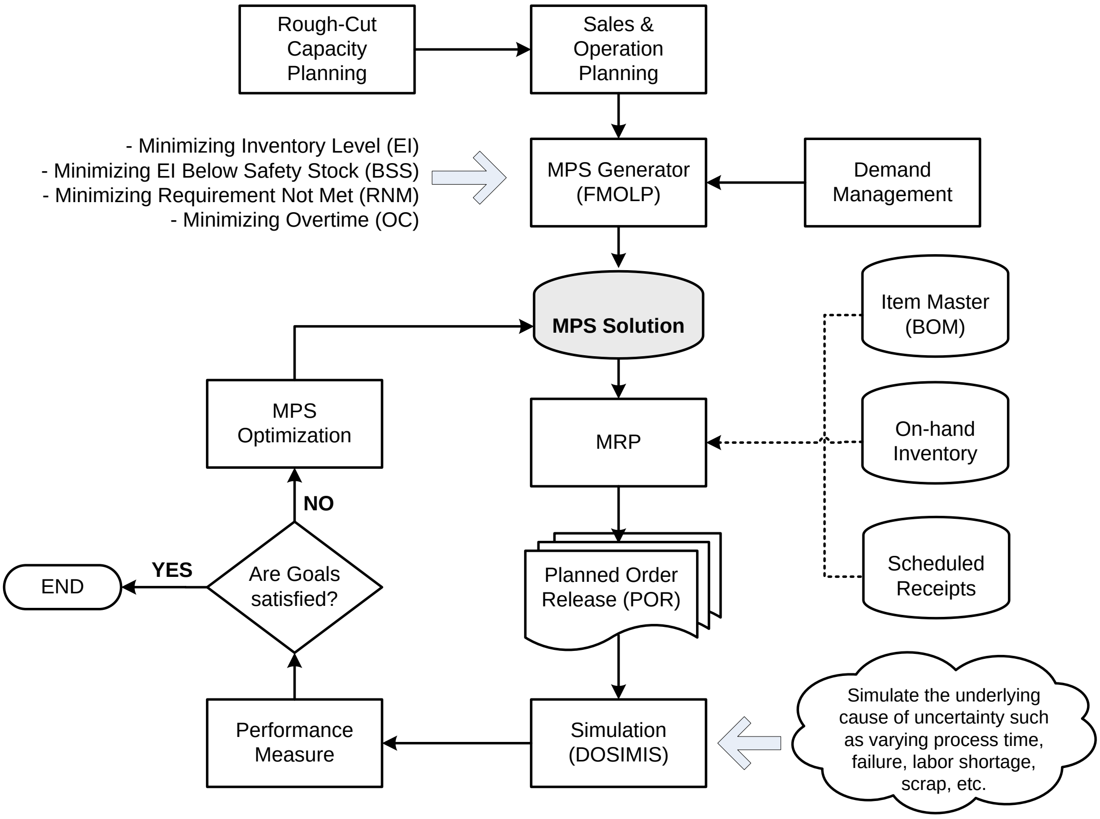

Rough-Cut Capacity Planning
Sales & Operation Planning
- Minimizing Inventory Level (EI)
- Minimizing EI Below Safety Stock (BSS)
- Minimizing Requirement Not Met (RNM)
- Minimizing Overtime (OC)
MPS Generator (FMOLP)
Demand Management
MPS Solution
Item Master (BOM)
MPS Optimization
MRP
On-hand Inventory
NO
YES
END
Are Goals satisfied?
Planned Order Release (POR)
Scheduled Receipts
Simulate the underlying cause of uncertainty such as varying process time, failure, labor shortage, scrap, etc.
Performance Measure
Simulation (DOSIMIS)

Figure 23 Schema of closed-loop MPS

Planned order request (POR) will be one of outcomes of MRP I process. POR contains information associated with production such as part number, quantitiy to be manufactured, work center, production start date, duedate, and also customer number if MPS is created for MTO environment. In standard production planning, a scheduler can alter any data in POR and make necessary adjustment if necessary as long as they are not yet converted into production order (PO). PO's are used to control production within company and also to control cost accounting. It defines material to be processed, at which location, at what time, resource to be used, how much work is required and how the order cost to be settled. Once a PO is generated, production process can be initiated according to the production order information.

In this approach, the execution of POR will be simulated before they are converted into production order (PO). For this purpose, a model of manufacturing system (shop-floor) is developed using DES and all generated PORs are exported into it. The simulation will mimic the real process of POR execution in shop-floor. During execution, stochastic situation such as varying processing time, unplanned machine failure, etc. is introduced. The outcome of simulation will be analyzed and performance indicators, which *EI, RNM, BSS* and *OC* are measured. If DM's targets are not met, an adjustment of tentative MPS is required. Following the adjustment, the existing MPS solution is altered and MPS is run again to generate new POR. The planning process continues until DM's target is satisfied. The final output of this process should be a valid and realistic master production schedule.

The advantages of this methodology, which uses simulation model to validate a tentative MPS, are described as follows:

1. The simulation model gives complete overview of manufacturing system for either top management or other authorized busines function such as marketing department. Such that the top management and marketing people does not make an MPS as their "wish list" (*Kochhar et al., 1998*). Finally, it promote good communication between the marketing management and the manufacturing management.
2. The use of simulation model for verification overcomes the drawbacks of rough cut capacity planning (RCCP) in ERP system which considers only critial resource. Since the simulation model is able to mimic the detail of process centers including one with potential bottleneck.
3. The underlying causes of uncertainty can be also modeled so that the effect of uncertainy on the proposed MPS can be recognized in advance. In this way, the required capacity and planned capacity can be balanced more properly (Proud, 1999).

5.2 Information System of MPS

An information platform is required to realize the closed-loop MPS. In general, the activities of MPS creation using the proposed methodology can be categorized into three main layers: production planning logic (PPL), optimization and simulation (Figure 24). The PPL function includes rough-cut capacity planning (RCCP), sales and operation planning (SOP), demand management and material requirement planning (MRP I). The optimization function is meant to generate the initial solution of MPS, make MPS adjustment and compute the performance measure. And, the last function is simulation which aims to model manufacturing system. Since it is difficult to find a software package that is able to perform all functions, the information system of MPS is built of several software packages: Dosimis-3, SAP R/3 and MATLAB. Dosimis-3 is discrete event simulation software that is superior for modeling a material flow system. SAP

R/3 is a leading enterprise resource planning (ERP) world-wide, of which production planning module is one of its strength point. And, MATLAB is a high-performance tool for technical computing, with which optimization and computation can be easily delivered. Conceptually, the integration layer of three software packages is presented in Figure 24.

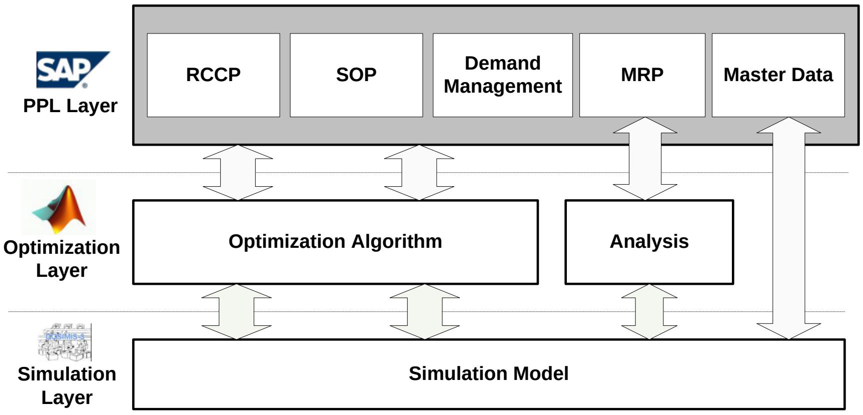


Figure 24 Integrated system for closed-loop MPS

5.2.1 Integration Model

According to *Brown* (2000), the functionality integration of several software package is categorized as functional or application integration. The application addresses the combination of disparate computer system into one system that uses a common set of data structures and rules to deliver the required functionality for a business area. In practice, the integration model is employed as tool to carry out an integration project. The integration model is to brigde the gap between the business view and the technical view as the business people and the technical people has different view of world. It helps place the technology in the background and bring the business use to the foreground. *Brown* (2000) has provided the various templates of integration model for different application area including ERP integration. Figure 25 depicts the integration model for closed-loop MPS. The steps in the integration model is described sequentially as follows:

1. Top management makes sales and operation planning for medium to long term planning (monthly, quarterly or yearly) according to internal requirement, forecasted customer demand, productive resource profile.
2. The SOP and Demand management will be basis for a master scheduler to determine the values of MPS input parameter such as gross requirement, target stock level, production rate, etc. (Chapter 2.1).

3. Using fuzzy multi-objective linear programming (FMOLP) and based on the MPS input, the proposed MPS solution will be generated (Chapter 3). The objective values of this solution $(EI^*, RNM^*, BSS^*, OC^*)$ are considered as DM's target.
4. The outcome of step 3 is a tentative, which means that the MPS solution may be changed if it is infeasible to implement.
5. The detail requirement of material and capacity will be computed as MRP run and planned order request (POR) is one of outcomes of this step.
6. At least, POR contains information of part number, quantity and due date (Chapter 2.2.4).
7. The execution of POR in manufacturing system is simulated by using discrete event simulation. This simulation ensure that the POR can be completed on time.
8. The simulation model of manufacturing system is developed based on the set of production data which is imported from SAP.
9. The production master data is a collection of tables which store information of production system such as routing, work center, processing time and so on.
10. Once the simulation model is run, the output data of simulation can be analized and performance measure is computed $(EI_i\ , RNM_i\ , BSS_i\ , OC_i;\ \ i = number\ of\ iteration)$.
11. If the performance measure are not equal or close to DM's target, the tentative MPS must be adjusted. Load leveling technique is used to adjust the tentative MPS (Chapter 7.2).
12. The iteration stops until the performance measure is equal or close to DM's target.

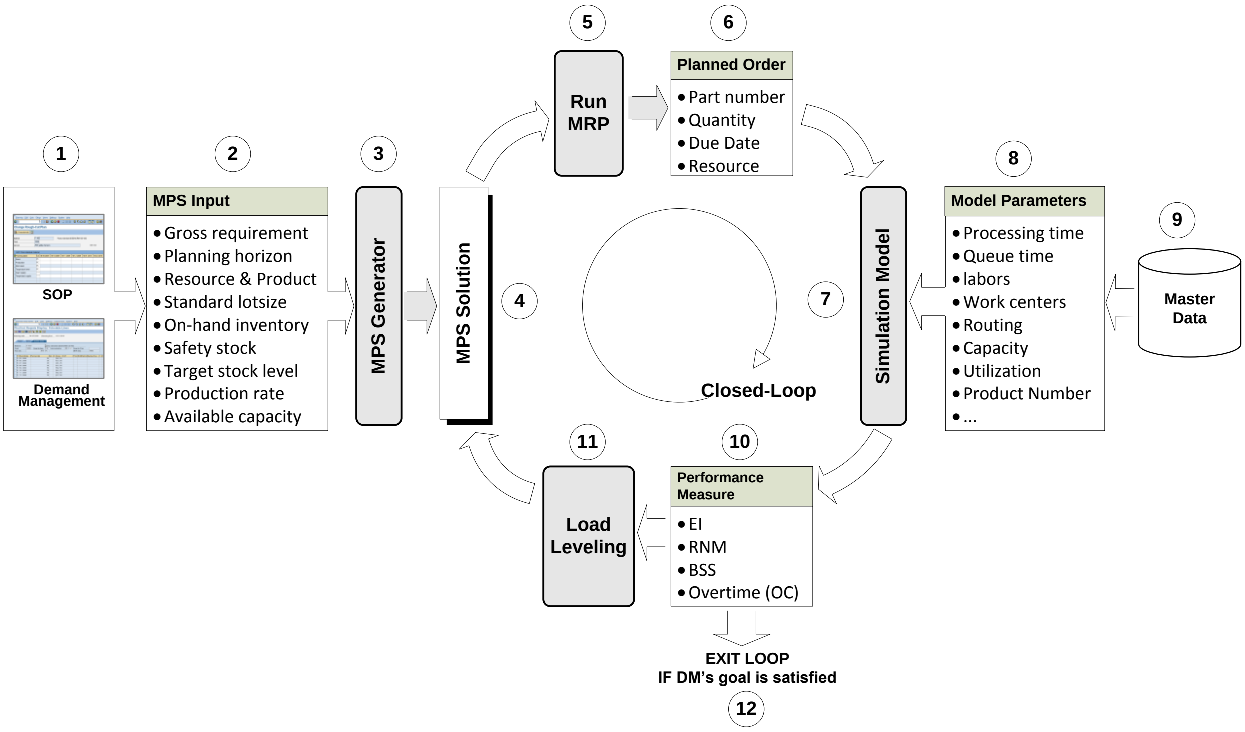


Figure 25 Integration model for MPS

5.2.2 Business Process Modeling

A business process is a collection of related or structured tasks or activities that produce a particular purpose (can be a service or product). Business Process Modeling Notation (BPMN) which is developed by Business Process Management Initiative (BPMI) is chosen to describe the sequential process of MPS creation. BPMN was developed to provide a notation that is readily understandable by all business stakeholders (*White*, 2004); the business analyst that create the initial draft of the processes, the technical developers responsible for implementing technology, and the business people who manage and monitor those process. From business process model, one can identify who (what) are involved in the process, what their role in the process (which activities/task) they must perform), and when those activities must be completed. Figure 26 depicts the business process model for MPS development. The agents involved in this process are scheduler or decision maker, SAP R/3, MATLAB and Dosimis-3. The document or data that is required or produced are master data, planned order (POR) and production order (PO).

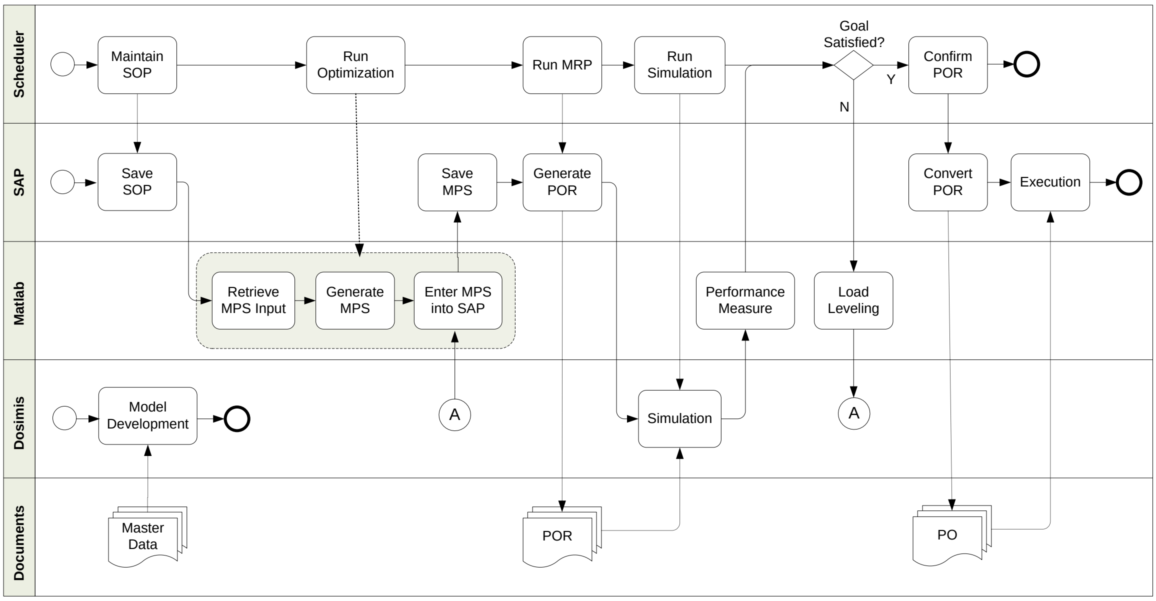


Figure 26 Business process for MPS development

5.3 Data Exchange & Mapping

Data exchange and data mapping is a common topic in system integration. According to the OECD glossary of statistical terms, the data exchange is the process of sending and receiving data in such a manner that the information content or meaning assigned to the data is not altered during the transmission. Since mostly each system has different data structure and type, the data mapping is required as data exchange.

5.3.1 System Interface

In order to enhance the system integrity, the modern software packages provide system interfaces that enable them to integrate or communicate with external systems in standard and robust manner, likewise, the three software packages used in this study: SAP R/3, MATLAB and DOSIMIS-3.

SAP's Interface

SAP is a business application solution which manages comprehensively financial, manufacturing, sales & distribution, and human resources. SAP R/3 system provides numerous interface methods as shown Figure 27. This in only brief summary of the available R/3 interface, for a more detailed description see *Buck-Emden et al.* (1996) and the R/3 documentation. *Buck-Emden (1996)* divide R/3 interface up into *object-oriented, high-level, medium-level* and *low-level* interface.

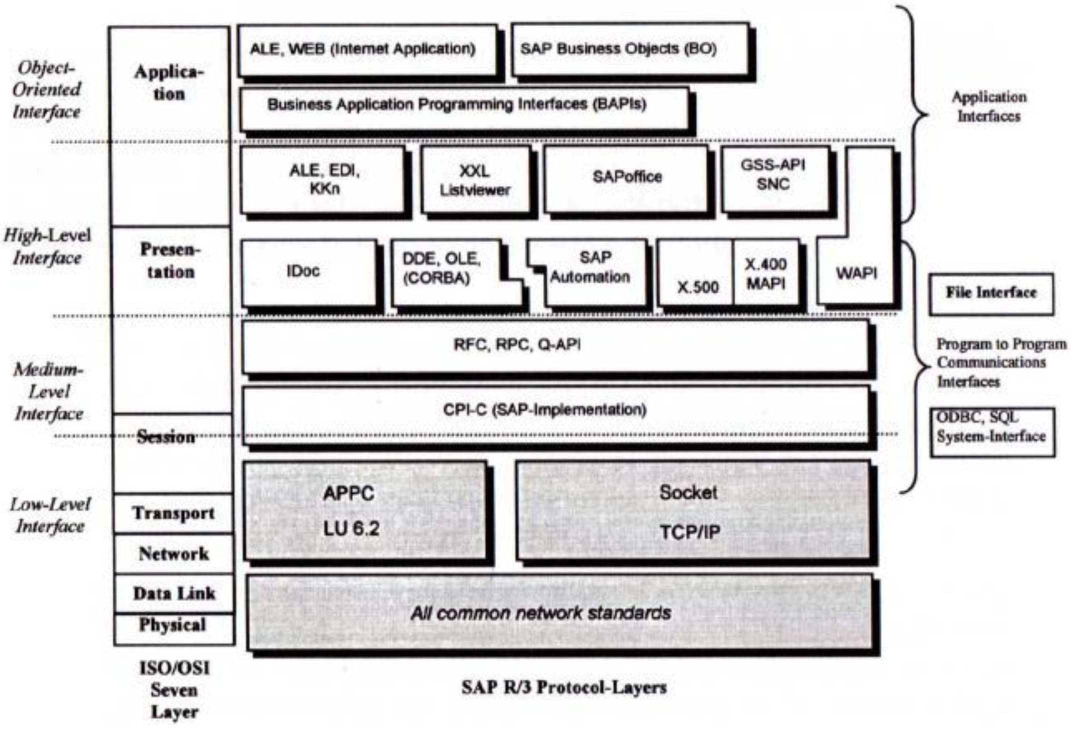


Figure 27 SAP R/3 Protocol-Layers[16]

Dosimis-3's Interface

Dosimis-3 is a modular-oriented graphical interactive standard simulator, which allows simulation of time-discrete material flow system. The interfaces provided by Dosimis-3 can be described as follows:

[16] Source: (Moser, 1999)

- *Excel-Interface*. It enables users to parameterize Dosimis-3 modules externally and in tabular form as well as to calculate module parameters by Excel formula.
- *COM-Server*. It let users to automate process during project. COM interface can be connected to VBA-Excel or Visual C++.
- *Program Interface (API) for decision table*. The type of interface allows simulation model to interact with external programs which is compiled into DLL file. The DLL file will be called from decision table as simulation running. As template, the project *EtFunktion,* which is developed with C/C++, exists in the project workspace *UserLibrary* of Dosimis-3.

MATLAB

MATLAB (version 7.0) stands for **Mat**rix **Lab**oratory, which is designed for numerical computation. MATLAB can be interfaced easily with external system. Table 23 presents the built-in external interface function provided by MATLAB.

Table 23 MATLAB external interface functions

Interface	Description
Generic DLL Interface Function	Functions that enable you to interact with external shared library (.dll) files.
C MAT-File Functions	Functions that allow you to incorporate and use MATLAB data in your own C programs.
C MX-Functions	Array access and creation functions that you use in your C MEX-files to manipulate MATLAB arrays.
C MEX-Functions	Functions that you use in your C MEX-files to perform operations back in the MATLAB environment.
C Engine Functions	Functions that allow you to call MATLAB from your own C programs.
Fortran MAT-File Functions	Functions that allow you to incorporate and use MATLAB data in your own Fortran programs.
Fortran MX-Functions	Array access and creation functions that you use in your Fortran MEX-files to manipulate MATLAB arrays
Fortran MEX-Functions	Functions that you use in your Fortran MEX-files to perform operations back in the MATLAB environment.
Fortran Engine Functions	Functions that allow you to call MATLAB from your own Fortran programs.
Java Interface Functions	Functions that enable you to create and interact with Java classes and objects from MATLAB.
COM Functions	Functions that create COM objects and manipulate their interfaces.
DDE Functions	Dynamic Data Exchange functions that enable MATLAB to access other Windows applications and vice versa.
Serial Port I/O Functions	Functions that enable you to interact with devices connected to your computer's serial port.

Having explored all interface methods of each system, one can determine the most suitable interface implemented for this system integration. Considering complexity and flexibility, the interface structure as shown in Figure 28 is proposed in this study.

Program language C/C++ is chosen as interface among the systems because the language is able to communicate with all of them. Using C/C++, one can develop a program that control and manipulate objects in simulator (Dosimis-3) through program interface (API), get the data from SAP through BAPIs and simultaneously invoke the genetic algorithm solver in MATLAB. Thus, the program is compiled into DLL file and subsequently is called from decision table as simulator at run. In this case, the business logic is implemented in project *EtFunktion*.

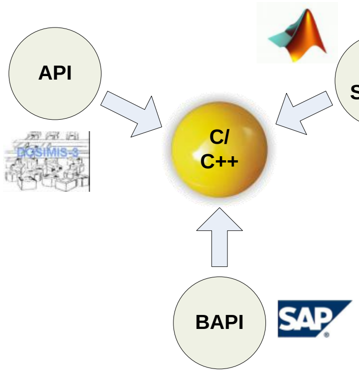


Figure 28 Interface of SAP R/3, Dosimis-3, MATLAB

5.3.2 Mapping POR and Work Schedule

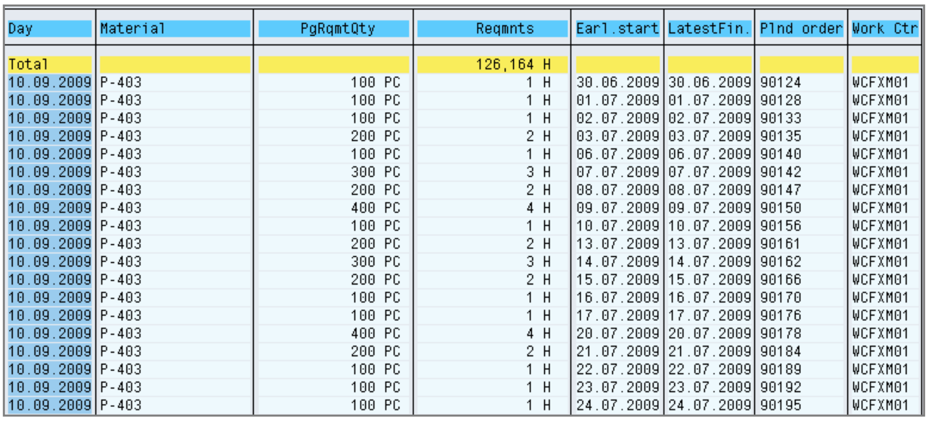
This study uses the free access of SAP IDES.6 provided by a SAP-Consultant (www.consolut.net). Due to security reason, the users are prohibited to access the BAPIs. Therefore, the POR cannot be imported automatically into Dosimis-3. Alternatively, the production data required is exported manually from SAP to a dummy database and thus, other systems (Dosimis-3 and MATLAB) retrieve the required data from that database. Dosimis-3 as simulation environment imports two types of data from SAP R/3. Firstly is master data that is required to build the simulation model. It contains the information of work center, processing time, routing, labor, etc (Figure 25). Secondly is planned order request (POR) data which is consumed as simulation run. The most crucial part in this integration is to map planned order (POR) of SAP R/3 and work schedule of Dosimis-3 because they have different data structure and format.

Planned Order (SAP R/3 version)

All planned orders for a production scenario and a specified work center can be reviewed using transaction code (t-code) CM38. As shown Figure 29, the planned orders are formatted in tabular form. Each row represents one planned order. At least, the columns of planned order table sequentially consist of:

- *Day* is the date in which the planning period starts.

- *Material* is unique key (number) to identify a material.
- *PgRqmQty* represents the quantity to be produced for corresponding planned order.
- *Reqmnts* represents capacity required to produce the given quantity (hrs)
- *Earl. start* means the earliest date, on which operation can start.
- *Latestfin.* means the later date, on which operation execution should finish.
- *Plnd order* identifies a planned order for the given material.
- *Work Ctr* is a unique key to identify the work center.
- *Lot Size* represents the production lot size (not shown)



Day	Material	PgRqmtQty	Reqmnts	Earl.start	LatestFin.	Plnd order	Work Ctr
Total			126,164 H				
10.09.2009	P-403	100 PC	1 H	30.06.2009	30.06.2009	90124	WCFXM01
10.09.2009	P-403	100 PC	1 H	01.07.2009	01.07.2009	90128	WCFXM01
10.09.2009	P-403	100 PC	1 H	02.07.2009	02.07.2009	90133	WCFXM01
10.09.2009	P-403	200 PC	2 H	03.07.2009	03.07.2009	90135	WCFXM01
10.09.2009	P-403	100 PC	1 H	06.07.2009	06.07.2009	90140	WCFXM01
10.09.2009	P-403	300 PC	3 H	07.07.2009	07.07.2009	90142	WCFXM01
10.09.2009	P-403	200 PC	2 H	08.07.2009	08.07.2009	90147	WCFXM01
10.09.2009	P-403	400 PC	4 H	09.07.2009	09.07.2009	90150	WCFXM01
10.09.2009	P-403	100 PC	1 H	10.07.2009	10.07.2009	90156	WCFXM01
10.09.2009	P-403	200 PC	2 H	13.07.2009	13.07.2009	90161	WCFXM01
10.09.2009	P-403	300 PC	3 H	14.07.2009	14.07.2009	90162	WCFXM01
10.09.2009	P-403	200 PC	2 H	15.07.2009	15.07.2009	90166	WCFXM01
10.09.2009	P-403	100 PC	1 H	16.07.2009	16.07.2009	90170	WCFXM01
10.09.2009	P-403	100 PC	1 H	17.07.2009	17.07.2009	90176	WCFXM01
10.09.2009	P-403	400 PC	4 H	20.07.2009	20.07.2009	90178	WCFXM01
10.09.2009	P-403	200 PC	2 H	21.07.2009	21.07.2009	90184	WCFXM01
10.09.2009	P-403	100 PC	1 H	22.07.2009	22.07.2009	90189	WCFXM01
10.09.2009	P-403	100 PC	1 H	23.07.2009	23.07.2009	90192	WCFXM01
10.09.2009	P-403	100 PC	1 H	24.07.2009	24.07.2009	90195	WCFXM01

Figure 29 The example of planned order screen in SAP R/3

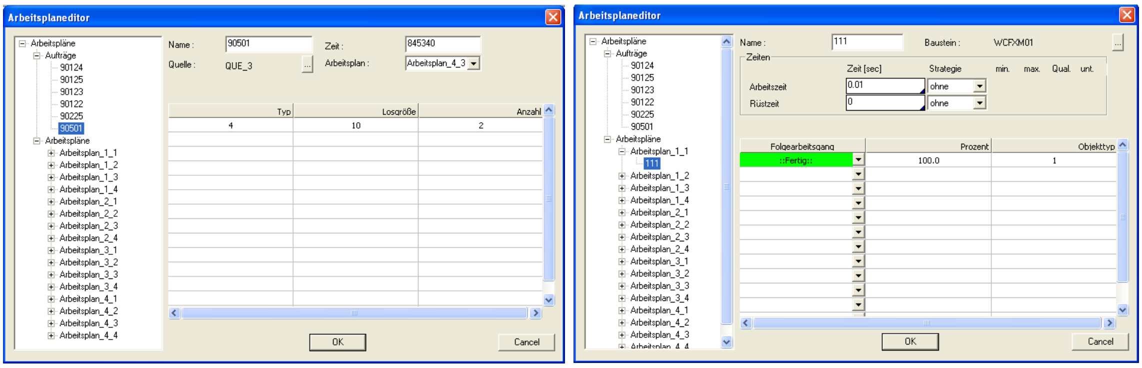
Work Schedules (Dosimis-3 version)

In Dosimis-3, the work schedules to be read by simulator at run can be found in menu <*Modell*><*Arbeitpläne*> for German Version. Figure 30 shows the work plan format which is completely different from planned order of SAP. The work schedule is divided into two screens: order (Aufträge) screen and work plan (Arbeitpläne) screen. The order screen consists of the following information (Figure 30, left).

- *Name* is unique key to identify an order.
- *Time* specifies the load time of order (job).
- *Source* specifies the element name of the source in which the materials are generated at the load time.
- *Object Type* is unique key to identify a material (object)
- *Lot-size* represents production lot-size
- *Quantity* specifies the quantity of material to be manufactured
- *Work plan* specifies the sequence work centers in which the given material will be manufactured. It is similar to routing in term of SAP R/3.

At least, the work plan screen consists of the following information (Figure 30, right).

- *Name* specifies a unique key to identify work plan.
- *Work center* determines the productive resource on which the material is processed.
- *Processing time* specifies the time required to process the given material.
- *Setup time* specifies the time required to set up machine in the given work center.
- *Next operation* determines whether the material needs to be processed further or has been completed.
- *Percentage* specifies the scrap material generated by the corresponding work center.
- *Object Type* specifies a unique key to identify material.



Arbeitsplaneditor

Name : 90501 Zeit : 845340

Quelle : QUE_3 Arbeitsplan : Arbeitsplan_4_3

Typ	Losgröße	Anzahl
4	10	2

Arbeitsplaneditor

Name : 111 Baustein : WCFXM01

Zeiten

	Zeit [sec]	Strategie	min.	max.	Qual.	unt.
Arbeitszeit	0.01	ohne				
Rüstzeit	0	ohne				

Folgearbeitsgang	Prozent	Objekttyp
::Fertig::	100.0	1

Figure 30 The example of order screen (left). The example of work plan screen in Dosimis-3 (right)

Figure 31 shows the data mapping between planned order of SAP R/3 and work plan (Aufträge & Arbeitpläne) in Dosimis-3. Note that one needs to synchronize time format of SAP and Dosimis-3 since SAP R/3 works on daily basis, while Dosimis-3 runs on real-time basis which is commonly in second. In order to enable simulator read the work plans automatically, the imported planned order must be save in the file *Modelname.apl*. For detail explanation about the file format of *Modelname.apl,* please go to Chapter 6.3.2.1.

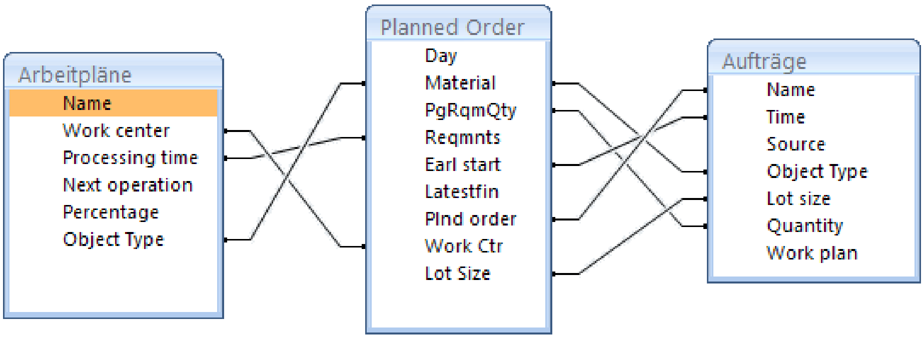


Figure 31 Mapping planned order (SAP) and work plan (Dosimis-3)

5.3.3 Data Exchange between Dosimis-3 and MATLAB

As presented in Table 23, the *C/C++ Engine Functions* enable us to call MATLAB from C program. In this study, the *C/C++ Engine Functions* is used to invoke GA's function in MATLAB from project *EtFunktion* call. To do so, first of all, the following steps must be done to configure the environment of project *EtFunktion.*

- Open workspace *UserLibrary* and then click menu *Tools -> Options*
- Select directories Tab and thus select "Include files" from the drop-down list
- Add "c:\MATLAB701\EXTERN\INCLUDE" to the list
- Select "Library Files" from the drop-down list
- Add "c:\MATLAB701\EXTERN\LIB\WIN32\MICROSOFT\MSVC60"
- Library path vary depending on the language and compiler

Table 24 The C/C++ Engine Functions

Function	Description
engClose	Quit MATLAB engine session
engEvalString	Evaluate expression in string
engGetVariable	Copy variable from engine workspace
engGetVisible	Determine visibility of engine session
engOpen	Start MATLAB engine session
engOpenSingleUse	Start MATLAB engine session for single, non-shared use
engOutputBuffer	Specify buffer for MATLAB output
engPutVariable	Put variables into engine workspace
engSetVisible	Show or hide engine session
mxCreateDoubleMatrix	Create double array matrix

In file *EtFunktionSim.cpp,* the header <*engine.h*> must be inserted. In addition, the following lines must be added just under tag *#include* to add appropriate library files.

```
#pragma comment(lib,"Libmx.lib")
#pragma comment(lib,"libmex.lib")
#pragma comment(lib,"libeng.lib")
```

Now, the MATLAB engine session can be invoked by using function *engOpen*. Once the engine session started, any MATLAB's functions can be called from the C/C++ environment. Note that MATLAB treats all variables as matrix; therefore, the scalar variable of C/ C++ must be converted into array matrix and vice versa. Table 24 presents C/C++ functions used frequently for interfacing MATLAB with external system. The following code (Table 25) is an example how to pass variable value to MATLAB's workspace and vice versa.

Table 25 The C/C++ code for integrating MATLAB

```
Engine *m_pEngine;                      % initiate variable
m_pEngine = engOpen(NULL);              % create MATLAB engine session
if (m_pEngine == NULL)                  % check whether session has been created or not
{
    % 1. pass variable value into workspace
    double c[1][ 2];                                    % variable in c++
    mxArray *mxc;                                       % create mxArray variable
    mxc = mxCreateDoubleMatrix(1, 2 ,mxREAL);           % allocate space of mxArray variable
    memcpy((void*) mxGetPr(mxc), (void *) c, sizeof(double)*1*2);    % copy a value
    engPutVariable(m_pEngine,"x", mxc);                 % pass value into workspace
    engEvalString(m_pEngine, "y=x';");                  % assign transpose of value to variable "y"

    % 2. Retrieve value of y from workspace
    double *cret[2][1];                                 % variable in c++
    mxArray *mret;                                        % create mxArray variable
    mret = engGetVariable(m_pEngine,"y");                 % retrieve value from workspace
    cret = mxGetPr(mret);                                 % copy value to variable "cret"
}
```

5.4 Production Planning – SAP

As shown Figure 23, the development of tentative MPS involves other production planning functions such as Sales & Operation Planning (SOP), Demand Management, and Material Requirement Planning (MRP). To avoid redundant work, this research uses PP-SAP module to carry out those planning functions instead develops own planning logics. This sub-chapter gives brief illustration how to perform SOP, Demand Management and MRP for the given manufacturing system in SAP R/3 (see case illustration in Chapter 3.3). It is assumed that all given materials are manufactured in-house.

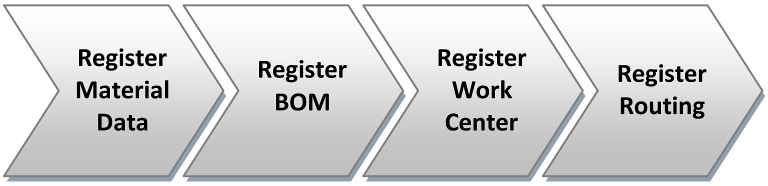


Figure 32 The sequence to maintain master data

5.4.1 Maintain Master Data

The master data is prerequisite of all logistic functions in SAP R/3 including production planning and control (PPC). The planning cannot be carried out until the master data has been entered into SAP. In general, the master data includes material data, work center & capacity, bill of material (BOM) in case of multi-parts material and routing data. Figure 32 depicts sequence of maintaining master data in SAP R/3.

5.4.1.1 Material Data

The material master is very essential data in SAP R/3 since it is used by all components in the SAP logistic system. The material data is required for purchasing, inventory management, invoice verification, sales & distribution and production planning & control (PPC) functions. The new material data can be configured by using transaction code (t-code) MM01. In general, the material data is classified into several screens as follows: basic data, sales, MRP, forecasting, work scheduling, plant data, warehouse, quality management, accounting and costing. For the given production scenario, the material data that must be entered into SAP is presented in Table 26. To simplify the production scenario, the rest materials are assumed to have identical properties as those of P-403. Therefore, the other three materials P-404, P-405 and P-406 respectively can be registered into SAP with the same information as presented in Table 26.

Table 26 The example of material data for P-403

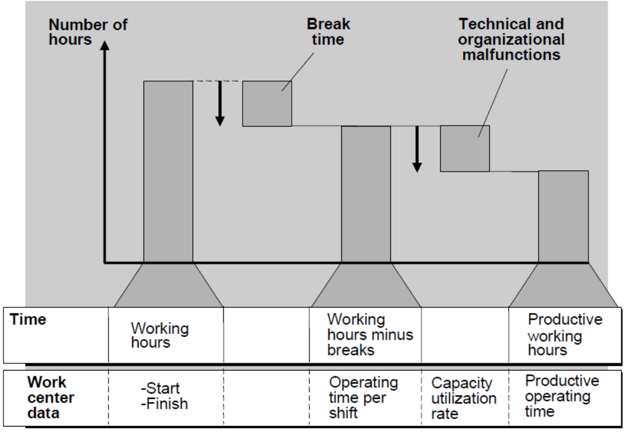
Field	Screen	Value	Description
Material Type	Basic data	FERT	Finished material
Material Number	Basic data	P-403	-
Unit	Basic data	PC	Piece (PC)
Material Name	Basic data	Pump standard	-
MRP Type	MRP	PD	The material is regularly planned according to MRP logic
Lot-Size	MRP	ES	It means lot-for-lot quantity with splitting. The lot-size is determined according to the quota of production line
Rounding Value	MRP	100	Lot-size is rounded up to 100 PC
Procurement Type	MRP	E	In-house production
In-house Production	MRP	1	One days production lead time
Sch. Margin Key	MRP	000	No float before and float after is set
Strategy	MRP	10	Make-to-stock production
Production Version	Work Sch.	0001 # 0002 # 0003 # 0004	The material can be produced in production lines 1 or 2 or 3 or 4.

5.4.1.2 Bill of Material

According to German standard (DIN), a bill of material (BOM) is a complete, formally structured list of the components that make up a product or assembly. BOM incorporating with routing contain essential data for integrated materials management and production controls. BOM of a material can be configured using transaction code CS01. The configuration of BOM is not necessary in this case because the produced materials in the given production scenario are composed of single part instead of multiple parts.

5.4.1.3 Work Center

Work center (WC) is a business object in which an operation is carried out. In SAP R/3 a work center can represent: machine or machine groups, production lines, assembly work centers, or employee/groups of employee. Together with BOMs and routings, work center belong to the most important master data in PPC system. Work centers are used for scheduling, costing, capacity planning and simplifying operation maintenance. The data in work center is grouped thematically in screen or screen groups as follows: basic data, assignment, capacities, scheduling, costing and hierarchy.



Time	Working hours		Working hours minus breaks		Productive working hours
Work center data	-Start -Finish		Operating time per shift	Capacity utilization rate	Productive operating time

Figure 33 The illustration of operating time calculation[17]

To perform specific task, each work center has capacity or called as work center capacity. The work center capacities are expressed in unit time with reference to a day or a shift. In SAP R/3, the work center capacity is differentiated into two terms: available capacity and operating time, which each of them is described as follows.

$$Available\ capacity = Operating\ time\ \times number\ of\ individual\ capacities$$

$$Operating\ time = (working\ time - break\ time) * capacity\ utilization\ rate/100\%$$

Working time is calculated from the start time minus the finish time. The figure 33 illustrates the relationship between those parameters.

[17] Source: SAP IDES Documentation

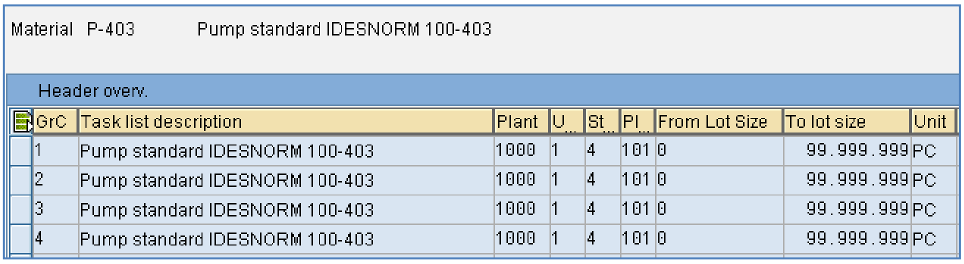
The available capacity is also known as standard available capacity. In SAP R/3, one can configure different available capacity profile along planning horizon if necessary. For example, adding night-shift into second week of planning horizon. For simplification, this scenario assumes that each production line consists of one work center. It implies that the productive version 0001, 0002, 0003, and 0004 is represented by work center WCFXM01, WCFXM02, WCFXM03, and WCFXM04 respectively. The information of work centers can be entered in SAP R/3 by using transaction code CR01. It is assumed that all work-centers are identical and have the same properties as presented in Table 27.

Table 27 The work center data

Field	Screen	Value	Description
Category	Basic Data	0001	Category of work center is Machine
Person responsible	Basic Data	101	A crew with code 101
Location	Basic Data	1	In production area
Suitability	Assignment	04	Mechanic is assigned as operator
Setup type key	Assignment	MB	Setup task is performed by machine operator
Wage group	Assignment	002	Qualification of operator is skilled worker
Setup formula	Capacity	-	No setup required
Processing formula	Capacity	SAP006	The requirement capacity due to processing is calculated according to formula: (machine*operation quantity/Base qty)
Start time	Capacity	08:00:00	Operations start at 8 AM
Finish time	Capacity	17:00:00	Operations in all work centers finish at 17 PM except WCFXM03 which ends at 16 PM.
Break time	Capacity	1	Break time is one hour
Utilization	Capacity	100	Utilization 100%
No of individual	Capacity	1	There only 1 machine in one work center
Active version	Capacity	1	No shift is assigned during planning horizon
Base unit	Capacity	H	Hours
Processing formula	Scheduling	SAP002	The scheduling of this work center is counted based on the formula: (machine*operation quantity/base qty/operation split)

5.4.1.4 Routing

A routing is a description of which operations (process steps) have to be carried out and in which order to produce a material. As well as information about the operation and the order in which they are carried out, a routing also contains details about the work centers at which they are carried out as well as about the required production resources and tools (including jigs and fixtures). In SAP R/3, the routing information is attached to a material. The routing tell us how (where and in which operation sequence) the material is made.



Material P-403 Pump standard IDESNORM 100-403

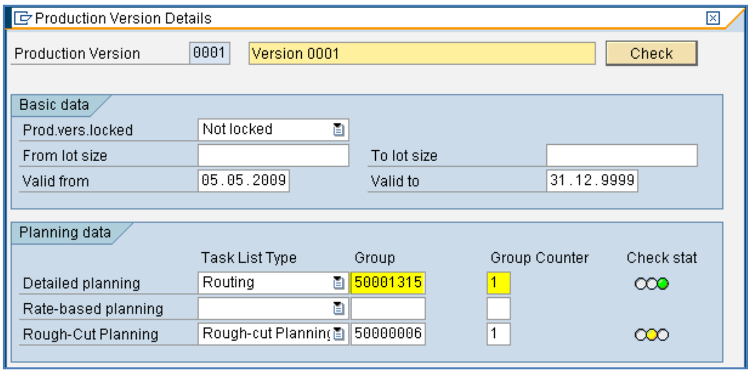
Header overv.

GrC	Task list description	Plant	U...	St...	Pl...	From Lot Size	To lot size	Unit
1	Pump standard IDESNORM 100-403	1000	1	4	101	0	99.999.999	PC
2	Pump standard IDESNORM 100-403	1000	1	4	101	0	99.999.999	PC
3	Pump standard IDESNORM 100-403	1000	1	4	101	0	99.999.999	PC
4	Pump standard IDESNORM 100-403	1000	1	4	101	0	99.999.999	PC

Figure 34 Material P-403 has four alternative routings

The given production scenario (Figure 13) considers that, in this case, each material can be produced in any available production lines (WCFXM01 or WCFXM02 or WCFXM03 or WCFXM04). It implies that an individual material should have four routing alternatives which each of them associates with the production lines. To implement such production strategy, the following procedures must be carried out.

1. For each material, create four alternative routings as shown by Figure 34 (t-code: CA01). Once created, each routing will have group id and group counter id, which can be reviewed on the routing header.
2. Input the *group id* and *group counter id* of routing into the corresponding production version which is defined as the material creation (t-code: MM01). The configuration is shown in Figure 35 below.



Production Version Details

Production Version 0001 Version 0001

Basic data

Prod.vers.locked: Not locked

From lot size: To lot size:

Valid from: 05.05.2009 Valid to: 31.12.9999

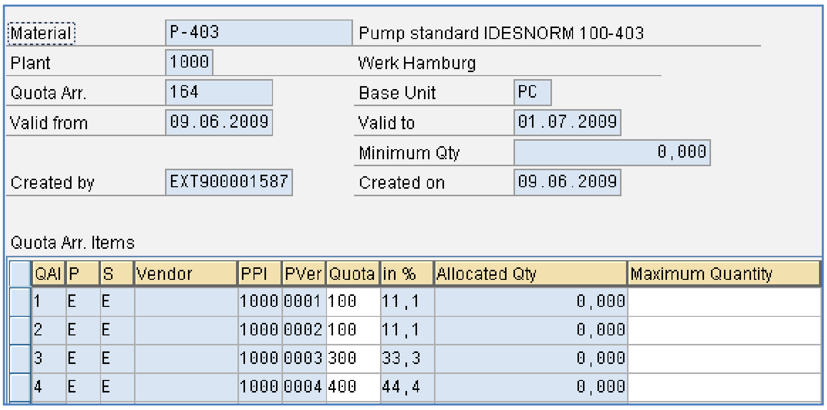
Planning data

	Task List Type	Group	Group Counter	Check stat
Detailed planning	Routing	50001315	1	○○●
Rate-based planning				
Rough-Cut Planning	Rough-cut Planning	50000006	1	○●○

Figure 35 The example of routing for production version 0001

3. The last step is to configure quota arrangement, which distribute the total quantity requirement to the available production lines. The total requirement quantity is split according to quota, which is in this case defined based on MPS solution. The quota of each

production line can be configured using t-code MEQ1. Figure 36 gives the example of quota arrangement for production lines. In this example, 11% of total quantity will be manufactured in production line 1, 11% is manufactured in production line 2, 33% is manufactured in production line 3 and 44% is manufactured is production line 4.



Material	P-403	Pump standard IDESNORM 100-403	
Plant	1000	Werk Hamburg	
Quota Arr.	164	Base Unit	PC
Valid from	09.06.2009	Valid to	01.07.2009
		Minimum Qty	0,000
Created by	EXT900001587	Created on	09.06.2009

Quota Arr. Items

QAI	P	S	Vendor	PPl	PVer	Quota	in %	Allocated Qty	Maximum Quantity
1	E	E		1000	0001	100	11,1	0,000	
2	E	E		1000	0002	100	11,1	0,000	
3	E	E		1000	0003	300	33,3	0,000	
4	E	E		1000	0004	400	44,4	0,000	

Figure 36 The example of quota arrangement

5.4.2 Maintain Production Plan in SOP

Sales and Operation Planning (SOP) is a flexible forecasting and planning tool with which sales, production, and other supply chain targets can be set on the basis of historical, existing, and estimated future data. To estimate the amount of capacities required to meet these targets, rough-cut planning can also be carried out. The outcome of SOP is passed to Demand Management in the form of independent requirement. In turn, this data is fed to PP-MPS and subsequently drives PP-MRP functions. In addition, SOP functions can be interfaced with other analysis tool such as Profitability Analysis (CO-PA), Cost Center Accounting (CO-OM-CCA) and Activity-Based Costing (CO-OM-ABC).

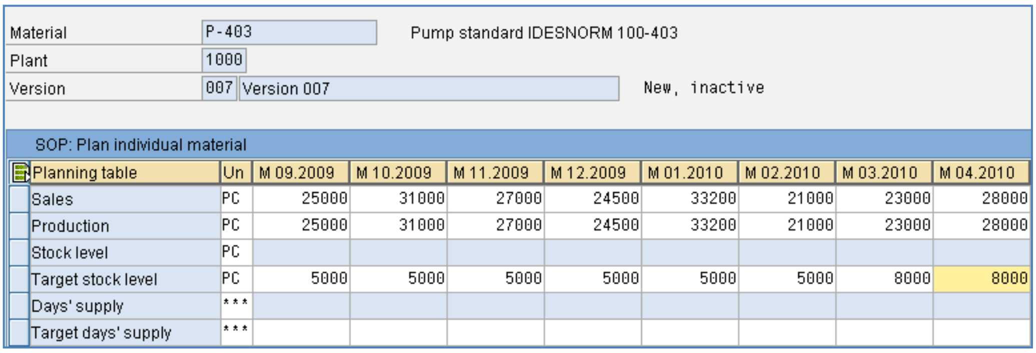
In SAP R/3, the production plan is created in SOP modules by using transaction code MC87. The sales quantity and operation policy such as target stock level and target day's supply are entered in the production plan screen. The system will calculate the quantity to be produced on the basis of production plan strategy (Figure 37). SAP R/3 provides four plan strategies as follows.

1. *Synchronous to sales plan*. It implies that the production quantity is exactly the same as planned sales quantity.
2. *Zero stock level*. This strategy uses any existing stock to cover sales quantities. When this stock has been used up, the production quantities are planned in synchronization with sales quantity.

3. *Target stock level*. This strategy calculates the production quantities needed to achieve target stock levels. The target stock level could be distribution function.

   Then, the system works out the production quantities on the basis of sales quantity and target stock level.

4. *Target day's supply*. This strategy calculates the production quantities needed to achieve target day's supplies. The day's supply for a period is the stock level divided by the average requirements (where the average requirement = sales quantity/number of workdays).



Material P-403 Pump standard IDESNORM 100-403

Plant 1000

Version 007 Version 007 New, inactive

SOP: Plan individual material

Planning table	Un	M 09.2009	M 10.2009	M 11.2009	M 12.2009	M 01.2010	M 02.2010	M 03.2010	M 04.2010
Sales	PC	25000	31000	27000	24500	33200	21000	23000	28000
Production	PC	25000	31000	27000	24500	33200	21000	23000	28000
Stock level	PC								
Target stock level	PC	5000	5000	5000	5000	5000	5000	8000	8000
Days' supply	***								
Target days' supply	***								

Figure 37 The example of production plan with strategy "synchronous to sales" for material P-403

5.4.3 Maintain Demand Management

The production plan created in SOP is transferred to Demand Management as planned independent requirement. Combined with customer requirement, which is created in Sales Order Management, the planned independent requirement is used by Demand Management to create demand program. The demand program contains the total requirement quantities (finished product, tradable assemblies, trading goods, replacement parts) and delivery dates for finished product assemblies. In order to create the demand program, one must define the planning strategy for a product that represents methods of production for planning and manufacturing or procuring a product. Using these strategies, one can decide if production is triggered by sales order (make-to-order), or not triggered by sales order (make-to-stock), or (assembly-to-order). The outcome of demand program is planned independent requirement.

Transaction code MC74 is used to transfer production plan into Demand Management. Table 28 presents the transfer parameters applied for material P-403. Other materials (P-404, P-405, and P-406) have also the same parameter values. The given scenario assumes that the production is triggered only by sales plan. The customer demand will consume only the warehouse stock. Therefore, the selected strategy planning is make-to-stock (MTS).

Table 28 The transfer parameters for material P-403

Field	Value	Description
Material	P-403	It is material identification
Transfer strategy	3	The quantity transferred into Demand Management is exactly equal to the production plan quantity
Start date	01.07.2009	Planning horizon start on 1st July
Finish date	31.07.2009	Planning horizon end on the end of July
Requirement Type	LSF	Planning strategy is make-to-stock (MTS)
Version	001	The planned independent requirement is inactive (001). It means that the output will not be applied directly to the running (active) planning.

5.4.4 Material Requirement Planning (MRP)

The main function of material requirement planning (MRP) is to guarantee material availability, that is, it is used to procure or produce the requirement quantities on time both for internal purposes and for sales and distribution. This process involves the monitoring of stock and, in particular, the automatic creation of procurement proposal (planned order for production or purchase requisition for purchasing). The process flow of MRP in SAP R/3 for in-house production can be described as follows.

1. In order to cover the total requirement defined by MPS, MRP calculates procurement quantities and date as well as plans the corresponding procurement elements. The procurement element in the planning run is the planned order (POR) that can be changed, rescheduled, or deleted at any time.
2. The system also calculates the dependent requirements, that is, the quantity of component required to produce the finished product or the assembly, by exploding the BOM. If a material shortage exists, planned orders are created at every BOM level to cover requirement.
3. The system then converts these planned orders into production order.
4. The progress of order for material produced in-house is controlled by using the production order. The production order contains its own scheduling procedures, capacity planning, and status management. Cost accounting is also carried out via the individual production order.
5. The quantities made available by production are placed stock and are managed by Inventory Management.

Screen for running the MRP process can be activated with transaction code MS31. The parameters must be entered into the screen presented by Table 29. Since this planning is meant for simulation and to avoid the MRP output disturbs the active MRP, the long-term planning module is used instead. Consequently, a planning scenario must be created before the MRP run, which specifies the version of planned independent requirement that is taken into account in this planning. The planning scenario can be configured by transaction code MS32.

Table 29 The MRP running parameters for material P-403

Field	Value	Description
Planning scenario	200	Unique number to specify the version of planned independent requirement
Material	P-403	Unique key to identify material
Processing key	NETCH	The system changes plan for materials that is included in the planning file
Create MRP list	1	MRP list will be created after run
Planning mode	3	Delete and recreate planning data
Scheduling	2	Lead time scheduling and capacity planning will be created
With firm planned order	1	Use setting in planning scenario

Chapter 6

Simulation Model

As described in the production planning process, the set of planned orders (POR), which is derivative of master production schedule, will trigger operation execution as soon as they are converted into production order (PO). One may think that all converted planned orders must be able to be completed on time using the available productive resource since one has estimated the required capacity as MPS development. However, in real world an unpredictable occasion may occur as execution and ultimately make some production orders delayed. That unpredictable event can be machine failure, product decay, labor shortage, or material shortage where all of them are stochastic in nature. That can be also varying processing time or setup time. In real practice, it is very rare that all of those underlying causes of uncertainty are considered properly as MPS creation. On the basis of this drawback, the simulation model of manufacturing system, which attempts to imitate the real situation of manufacturing system including its stochastic situation (machine failure and varying processing time), is built. The generated PORs will be executed in the simulation model to verify whether the planning is still feasible to implement or not if certain level of uncertainty (disturbance) take places. Naturally, the uncertainty events are not able to be predicted 100 percent in advance; however, most literatures agree that it can be approximated by simulation.

6.1 Simulation Procedures

Simulation is a methodology whereby a user attempts to understand a real-world situation (*Klafehn at el., 1996*). In understanding this situation, the user ordinarily has an objective in mind and a problem to solve. As oppose to other methodologies in which optimal, or best, solution are generated, simulation attempts to model a system and observe how changes in some of the system's parameters affect the behavior of system (output) over time. In order to get a meaningful simulation study, a systematic procedure must be carried out in developing simulation model. The following steps summarize standard procedure used by *Ziems at el* (1993) for developing a realiable simulation in Dosimis-3.

1. *Problem Formulation* – At this stage, one must have an overall objective in mind. For example, why is the simulation being pursued? What does one hope to gain from the results of the simulation? In addition, the original system structure including its boundary and process must be defined.
2. *Model Development* – This step includes the development of conceptual model, the designing of work flow of model's operation, the selection of appropriate model elements to represents original system components and the data collection.
3. *Model Implementation* – This stage implements the defined original system into a simulation model. The model parameters are entered, the connection between model elements is defined, and the original operation strategies are transformed into model.
4. *Model Validation* – The model validation aims to verify the model functionality and model input. During validation, one must ensure that the model's outputs are indicative of result generated by the real-world system.
5. *Experimental Testing* – The step shows the ability of simulation model to illustrate the impact of model parameter or input change on the simulation output. The model is regarded as valid if it has the same dynamic behavior as the real system.
6. *Analysis of Output Data* – This step consists of the evaluation of simulation output quality by using statistical analysis, graphical analysis and animation.
7. *Documentation and Presentation* – All activities related to the simulation development, the evaluation of simulation output, conclusion and recommendation are documented at this stage.

6.2 Problem Formulation

As illustration, a simulation model which represents the given production scenario (see Chapter 3.3) is developed in this chapter. Dosimis-3 is selected as environment to build the manufacturing system model since it provides complete material flow element models such as work station, conveyor, transportation, warehouse, etc. In addition, Dosimis-3 gives the visualization of material flow and it's easy to use.

6.2.1 Simulation Purpose

This simulation has two main objectives which are described as follows.

1. To ensure that the DM's targets which are expressed as $EI^*, RNM^*, BSS^*\ and\ OC^*$ can be achieved as the work plan (POR) execution within planning horizon.
2. To verify whether the planned orders (POR) can be completed on time under dynamic production variables such as varying processing time and machine failure. This simulation shows the effect of those underlying causes of uncertainty on the completion of POR.

6.2.2 Manufacturing System Description

A manufacturing system consists of four working area: supply area, production area and packaging area (Figure 38). It manages four production lines (R1, R2, R3, R4) running in parallel, and produce four types of products (P-403, P-404, P-405, P-406). All products can be manufactured in any production lines with different production rate respective of product types. In the supply area, there are four warehouses which are used to stock the material. The quantity of each material respecting of production lot-size to be manufactured in a specified production line has been determined at the beginning of period. The defined quantities are delivered to the corresponding production lines by using internal transportation system (T1).

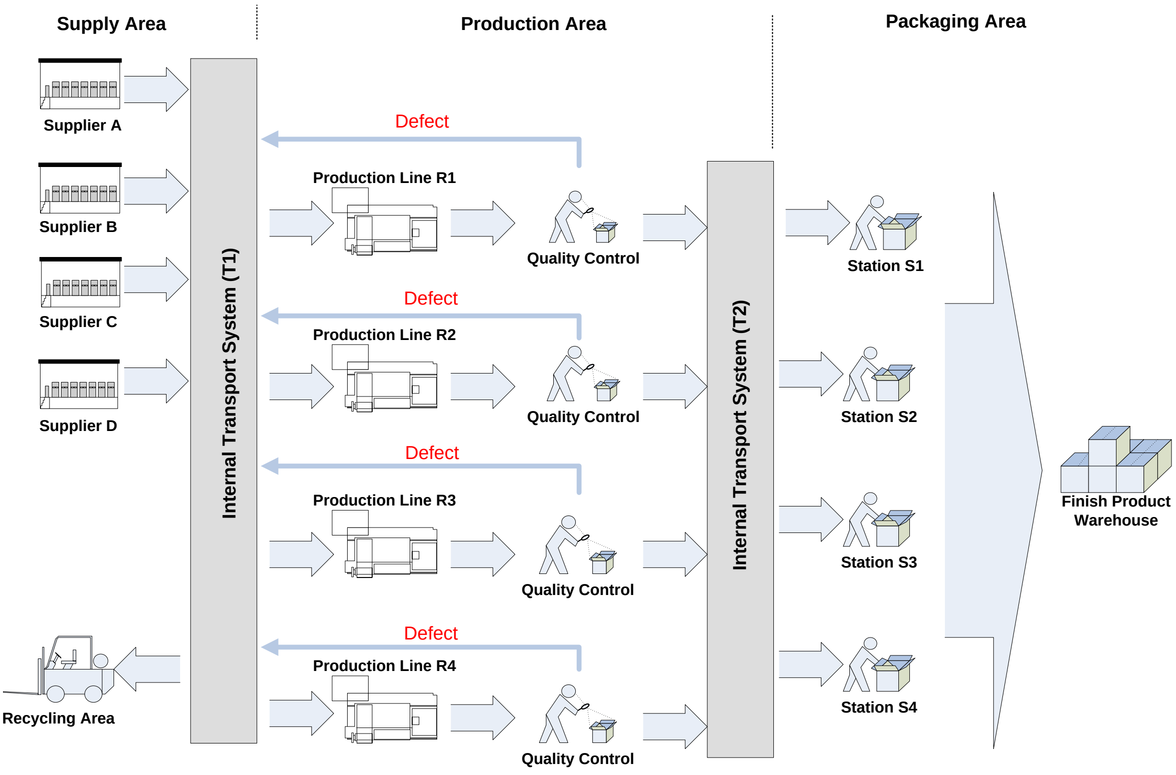


Figure 38 The overview of manufacturing system

The production area has four identical production lines, where each of them is composed of one work-center and one quality control station. Having processed in the work center, the defect products are forwarded to the recycling area, while the final product is carried to the packaging area using internal transportation system (T2). The packaging area consists of four stations S1, S2, S3, S4 which package the final products P-403, P-404, P-405, P-406 respectively. The T2 will automatically identify the products types and deliver them to the right packaging station. From the packaging station, the final product is carried to the final product warehouse.

At the end of planning horizon, a scheduler must generate master production schedule to determine how many and when the quantity must be produced in the manufacturing system for the next planning horizon. The problem is that the MPS was often failed to be completed due to inaccurate presumption. For instance, as MPS creation, the master scheduler assumes that the processing times (production rate) of production lines are constant for all time period. In fact, the value of processing time is random normal distribution with certain standard deviation. Therefore, the scheduler intends to validate the MPS using the simulation model with which the varying production rate can be modeled. In addition, other disturbance such machine failure is also exposed into the manufacturing model in order to evaluate the reliability of MPS.

6.3 Model Development and Implementation

The discussion of model development and implementation is grouped into two sub-sections; selection of model elements and model auxiliaries. The former describes the model structure of manufacturing system while the last one puts more attention on the auxiliary elements supporting the main model. Note that in this simulation all elements, on which one does not concern (not related to simulation objective), are not allowed to lead to material flow delay. Thus, the time and velocity parameters for those elements are set very small and very large, respectively (*Ziems at el,.* 1993).

6.3.1 Selection of Model Elements

The primary problem in this scenario is the varying production rate (processing time) and unpredictable event (machine failure) on production line which are not be taken into account as MPS creation. Therefore, the production area, particularly work station, will be the main focus to model, whereas supply and packaging area are treated as external system. The internal transport system (T1) is excluded in the simulation model since the one assures that there is no material delay due to disturbance on the transport system (T1). If any, one does not want the transportation system problem make the analysis of simulation output bias.

Table 30 The parameter of storage element (LAG_51)

Product	Single Cycle (s)	Dual Cycle (s)	Storage Time (s)	Capacity (unit)
1	0.001	0.002	0.001	6000
2	0.001	0.002	0.001	6000
3	0.001	0.002	0.001	6000
4	0.001	0.002	0.001	6000

On the basis of that consideration, the supply area is modeled as one storage element and four source elements. The supply area can deliver the material into production area on time (not exceed planned required date) at all times. To realize this, the cycle time and storage time of warehouse is set relatively very small compared with the processing time of productive line and the capacity of warehouse is set very large (infinity). Table 30 present the value of parameter mask for the raw material warehouse. These parameter values will ensure that the production area can pull the materials any time as it is required.

Table 31 The elements of simulation model

Area	System Component	Element Type	Element Name
Supply area	Supplier 1, 2, 3, 4 Raw material warehouse	Source Storage	QUE_1, QUE_2, QUE_3, QUE_4 LAG_51
Production area	Incoming conveyor 1, 2, 3, 4 Outgoing conveyor 1, 2, 3, 4 Internal transport system Quality control 1, 2, 3, 4 Work Center 1, 2, 3, 4	Acc. conveyor Acc. conveyor Shuttle Distributor Workstation	SST_19, SST_24, SST_29, SST_34 SST_23, SST_28, SST_33, SST_38 VTW_43 VEL_22, VEL_27, VEL_32, VEL_37 WCFXM01, WCFXM02, WCFXM03, WCFXM04
Packaging & storage area	Packaging station 1, 2, 3, 4 Intermediate warehouse Final product collector Distributors/Customers	Workstation Storage Comb. station Sink	AST_44, AST_45, AST_46, AST_47 LAG_48 ZEL_49 SEN_50
Recycling/scrap area	Decay product collector A, B Recycle station A, B	Comb. station Sink	ZEL_39, ZEL_41 SEN_52, SEN_53
Auxiliaries	Calendar database Calendar control Timer Log output data control 1,2,3,4 Machine failure control 1,2,3,4 Running date	Quick table Decision table Timer Decision table Failure control Animation text	QTB_2 GSZ_3 TIM_1 STA, STB, STC, STD STO_1, STO_2, STO_3, STO_4 date

Thus, the four source elements generate amount of material quantity according to the work plan (POR). Once generated, the products are stocked into "infinite capacity" warehouse. Such that the materials will be not blocked in source elements although large quantity of materials must be produced in the same time. The production area consists of four identical production lines. Each production line is composed of one incoming buffer storage (conveyor), two workstations, one distributor, and one outgoing conveyor.

The distributor element represents quality control station, which carries the scrap materials to recycling area. The length and speed of distributor elements is set as 0.2 m and 100 m/s respectively. The recycling area is simply modeled as sink element. At the end of all production lines, a shuttle element is installed to model internal transport model (T2). This transportation system is used to carry a specified material from the production lines to the corresponding packaging station. Material P-403 is carried to packaging station S2, material P-404 to S2, material P-405 to S3 and material P-406 to S3. To do so, the strategy "destination with" is chosen for this system. It is assumed that there is no block caused by T2, so that all length dimensions is set as 0.2 m and all velocity dimensions are set as 1000 m/s.

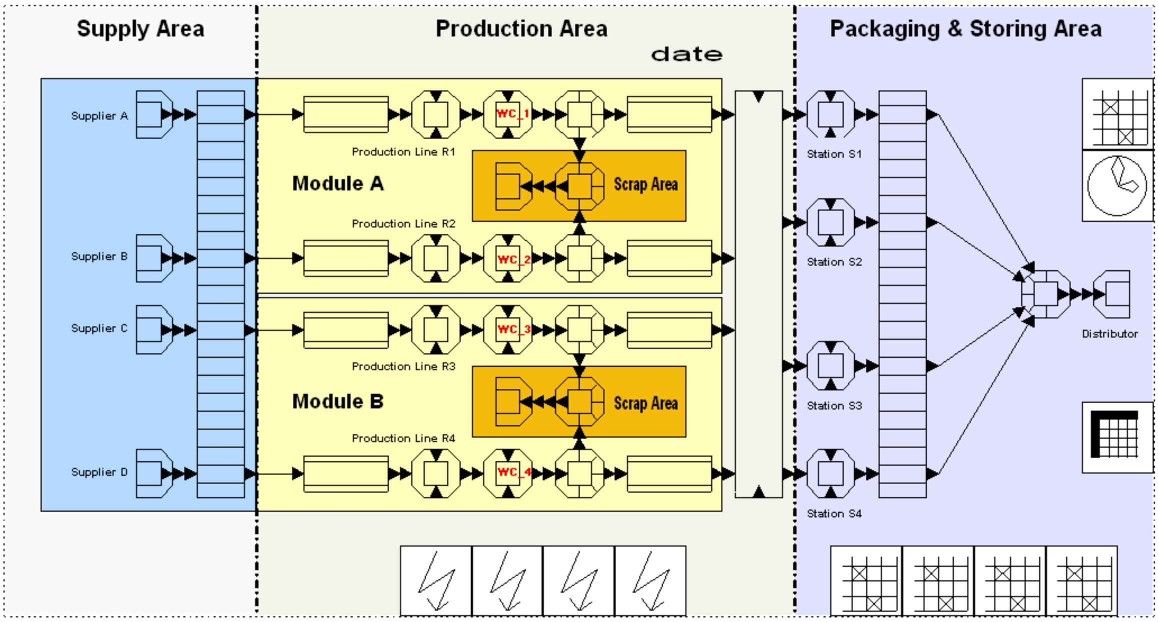


Figure 39 The simulation model of manufacturing system

The packaging area consists of four work stations, a warehouse, a combining station and a sink element. One also expects no block (disturbance) due to any model elements in packaging area. In model, the speed and length of all work stations and combining station are set as 100 m/s and 0.2 m, respectively. The warehouse capacity, cycle time and storage time is 8000 units, 0.01 s, and 0.01 s, respectively. The elements used in this simulation model are summarized in Table 31, whereas Figure 39 depicts complete simulation model for the given manufacturing system. To model uncertainty, the varying processing time and machine failures are applied on the active work stations which are WCFXM01, WCFXM02, WCFXM03 and WCFXM04.

6.3.2 Auxiliary Controls

The auxiliary controls are the parts out of model elements supporting simulation. It encompasses work plan, factory calendar, machine failure control, and log throughput data. The next section explains more detail about the auxiliary elements.

6.3.2.1 Work Schedules

In this simulation, the objects (materials) are processed according to work schedules, which are read from work plan files by the simulator at run time. The work schedules must be saved in a file [*Modelname.apl*] and put in the directory in which the model is located. To make the file readable by the simulator, the file format must be agreed with standard format defined by Dosimis-3. The work schedule file is divided into three blocks as follows.

1. File blocks

   This block may be located once in the file [*Modelname.apl*] and is initiated with the keyword **DATEI**. The following words up to the next blank line are interpreted as files names. The file blocks are indicated by the following syntax.

   DATEI <*file name*> <*blank line*>

2. Job blocks

   The job block contains list of objects which must be created at specified time in the specified source. The job data are specified by the following keywords.

ZEIT <*positive real number*>	: specifies the load time of job
QUE <*element name*>	: the element name of the source in which the objects are created at the load time
TYP <*object type*>	: specifies object type (material number)
LOS <*lot size*>	: specifies production lot size
ANZ <*number of objects*>	: specifies the quantity to be created
APL <*work plan name*>	: specifies the name of work plan

3. Work plan blocks

   The work plan blocks contain the job-independent part of the work plans. The following format indicates the work plan blocks.

   APL <*work plan name*> <*new line*><*working step blocks*>

   Where, the *working step block* is indicated by the following keywords:

AGN <*name*>	: specifies working step name
AST NAME <*valid workstation*>	: specifies work station in which the object should be processed
AZT <*time in s*>	: specifies processing time/piece in the work station
RZT <*time in s*>	: specifies set-up time per piece
AGV <*AG-no*>	: the next working step block if any
PROZ <*probability*>	: the probability of object finish without defect
TYP <*object type*>	: indicates object type after processing

One work plan may contain multiple working step blocks.

The following is an example of the work schedule file whose name is `SAP-AUFTRAG`, which has four jobs and two work plans .

```
DATEI SAP-AUFTRAG
AUF 90124 ZEIT 0 TYP 1 LOS 10 ANZ 29 QUE QUE_1 APL route_1
AUF 90125 ZEIT 0 TYP 1 LOS 10 ANZ 7 QUE QUE_2 APL route_2
AUF 90222 ZEIT 10440 TYP 2 LOS 10 ANZ 50 QUE QUE_1 APL route_2
AUF 90422 ZEIT 25320 TYP 4 LOS 10 ANZ 6 QUE QUE_1 APL route_1

APL route_1 AGN 111 AST NAME WCFXM01 AZT 0.01 AGV 0 PROZ 100 TYP 1
APL route_2 AGN 111 AST NAME WCFXM02 AZT 0.01 AGV 0 PROZ 100 TYP 1
```

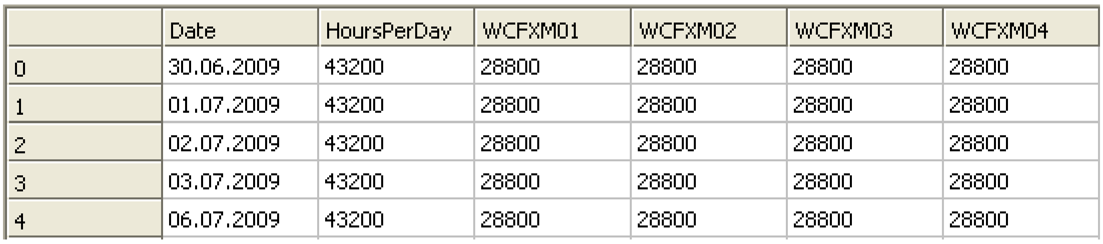
6.3.2.2 Factory Calendar

One of tricky tasks in this simulation is to introduce the factory calendar to the simulation environment. As the real manufacturing system, the simulation model should be able to distinguish between the working day, on which the operation can be carried out, and day off. To implement the factory calendar in the simulation model, it is required at least three control elements; quick table, timer and decision tables.

1. Quick table control (QTB_2)

   The quick table store the important information related to calendar such as the working date, the maximum working hours per day, and the operation hours per day (standard available capacity) for specified work station. Table 32 presents the example of quick table used in this simulation. The table indicates that the planning horizon to be simulated is between 30.06.2009 and 06.07.2009. The fourth (Saturday) and fifth (Sunday) July is excluded since they considered as day off. Second column represents the working hours per day. In this table, the factory can work maximum 12 hours (43200 s) per day. However, the work station WCFXM01, WCFXM02, WCFXM03 and WCFXM04 have standard available capacity as 8 hours (28800 s) per day. It implies that the maximum additional resource capacity (overtime) for each work station is 4 hours per day. Using this approach, one can assign different maximum working hours per day for a day as well as different available capacity for each work station.

Table 32 Quick table content



	Date	HoursPerDay	WCFXM01	WCFXM02	WCFXM03	WCFXM04
0	30.06.2009	43200	28800	28800	28800	28800
1	01.07.2009	43200	28800	28800	28800	28800
2	02.07.2009	43200	28800	28800	28800	28800
3	03.07.2009	43200	28800	28800	28800	28800
4	06.07.2009	43200	28800	28800	28800	28800

2. Timer control (TIM_1)

   Timer control is used to initiate set of actions in the decision table control. The timer control is activated every 12 hours (change of day).

3. Decision table control (GSZ_3)

   The decision table contains set of action (algorithm) that must be executed as day changes. The algorithm is described as the following pseudo-code.

Algorithm: **set of action in GSZ_3**

```
sim_zeit ← simulation time
day ← day variable
If (sim_zeit == (day*43200)) then
    day = day + 1;
    Find date in the current day in QTB_2
    Find available capacity for each work station in the current day
End if
```

If the simulation time is equal to multiple of maximum hours per day, it means that one period has been passed and new period is going to be started. The system increments the day variable with step 1 and then search the corresponding date in the quick table.

6.3.2.3 Machine Failure Control

This control is used to simulate the random machine failure following a certain probabilistic distribution. The scenario of this simulation is to apply failure on active work stations. Four failure controls STO_1, STO_2, STO_3, and STO_4 are set to trigger the random failure in work stations WCFXM01, WCFXM02, WCFXM03, and WCFXM04 respectively. The parameter values of all failure controls are presented in Table 33. To approximate the real situation, it is assumed that each machine has different availability which is 98%, 96%, 97%, 95% for work station WCFXM01, WCFXM02, WCFXM03, and WCFXM04 respectively.

Table 33 The parameter values of failure control

Failure Control	Work Station	Probabilistic Distribution	Equivalent Availability	MTBF (second)		MTTT (second)	
				Mean	Std. Deviation	Mean	Std. Deviation
STO_1	WCFXM01	normal	98%	29400	9800	600	200
STO_2	WCFXM02	normal	96%	18000	6000	750	250
STO_3	WCFXM03	normal	97%	21340	7113	660	220
STO_4	WCFXM04	normal	95%	14250	4750	750	250

6.3.2.4 *Log Files*

To ease in analyzing the simulation output, particularly, for computing performance measures, the system throughput is logged in the separated text file. The files [*FP_A.txt*], [*FP_B.txt*], [*FP_C.txt*] and [*FP_D.txt*] are created to save the throughput data of production line R1, R2, R3 and R4 respectively. The action to record the throughput data is invoked from the decision tables (*STA, STB, STC, and STD*) which are associated with the corresponding work station. Once an object entered into a work station, the system writes the data related to the object in the corresponding files. The recorded data consists of the date in which the object is processed, order number, object type, object quantity, lot size, and the simulation time at which the object is completely processed.

6.4 Model Validation

In order to validate the model, let consider some work plans (POR) that must be completed within the given planning horizon and the given production rate as presented by Table 34 and Table 35 respectively.

Table 34 Work schedules

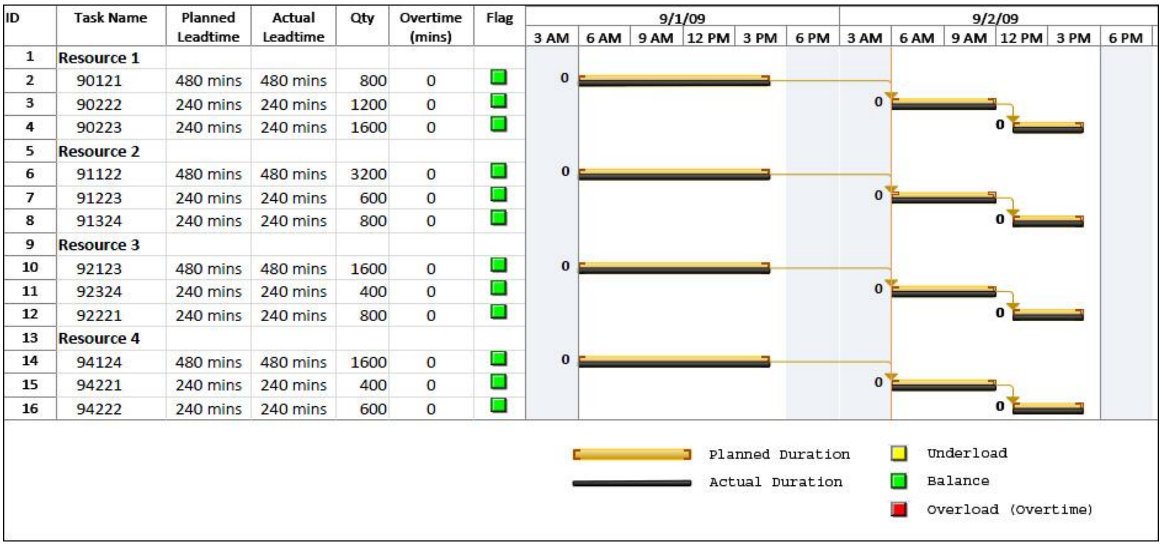
POR	Due date	Rscr.	Prod.	Qty	Lot size
90121	01.09.2009	1	1	800	10
91122	01.09.2009	2	2	1200	10
92123	01.09.2009	3	3	1600	10
94124	01.09.2009	4	4	3200	10
90222	02.09.2009	1	2	600	10
91223	02.09.2009	2	3	800	10
92324	02.09.2009	3	4	1600	10
94221	02.09.2009	4	1	400	10
90223	02.09.2009	1	3	800	10
91324	02.09.2009	2	4	1600	10
92221	02.09.2009	3	1	400	10
94222	02.09.2009	4	2	600	10

Table 35 Production rate

	Rscr. 1	Rscr. 2	Rscr. 3	Rscr. 4
Prod 1	100	100	100	100
Prod 2	150	150	150	150
Prod 3	200	200	200	200
Prod 4	400	400	400	400

Firstly, the stable environment, in which there is no disturbance (without machine failure and constant processing time), is subjected to the simulation model. In such condition, theoretically, one can assume that all production tasks (POR) must be able completed smoothly on time as work schedules. Table 36 presents a Gantt-chart that compares the planned and actual task execution. As it's expected that the actual lead times required for executing each POR are exactly the same as those of planning; the planned and actual quantities are equal and the overtime is also not required (zero). In another word, under ideal environment the model has shown similar behavior to the real system, where the production tasks can be carried out as work plan.

Table 36 Simulation output without disturbance



ID	Task Name	Planned Leadtime	Actual Leadtime	Qty	Overtime (mins)	Flag
1	Resource 1					
2	90121	480 mins	480 mins	800	0	🟩
3	90222	240 mins	240 mins	1200	0	🟩
4	90223	240 mins	240 mins	1600	0	🟩
5	Resource 2					
6	91122	480 mins	480 mins	3200	0	🟩
7	91223	240 mins	240 mins	600	0	🟩
8	91324	240 mins	240 mins	800	0	🟩
9	Resource 3					
10	92123	480 mins	480 mins	1600	0	🟩
11	92324	240 mins	240 mins	400	0	🟩
12	92221	240 mins	240 mins	800	0	🟩
13	Resource 4					
14	94124	480 mins	480 mins	1600	0	🟩
15	94221	240 mins	240 mins	400	0	🟩
16	94222	240 mins	240 mins	600	0	🟩

6.5 Experiment and Output Analysis

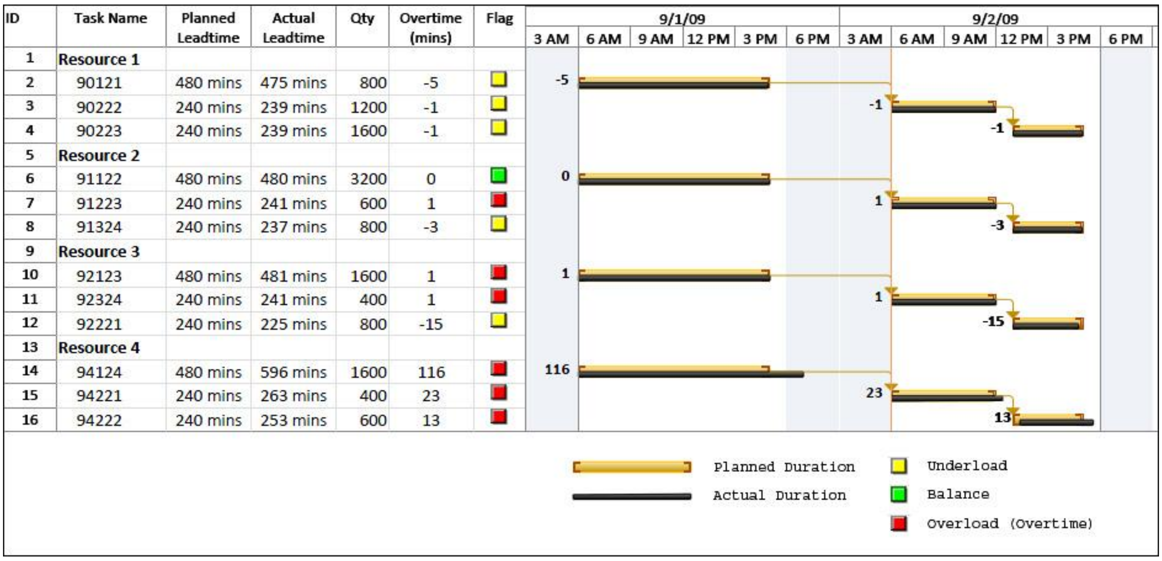
Two scenarios are proposed to evaluate the impact of dynamic production variables on the execution of work plan (POR). First scenario is to run simulation model with varying processing time instead of constant (Table 37). Second scenario is an extension of scenario one, where the work stations running with varying processing time and at the same time the machine failures take places. The machine failure and the varying processing time data for each work station are given in Table 33 and Table 37 respectively.

Table 37 The varying processing time (s/lot size)

Articles	Prod. Line 1		Prod. Line 2		Prod. Line 3		Prod. Line 4	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	360	20	360	60	360	25	360	90
2	240	20	240	60	240	25	240	90
3	180	20	180	60	180	25	180	90
4	90	20	90	60	90	25	90	90

These experiments are meant to test the influence of uncertainty due to varying processing time and machine failures on the work plan (POR) execution. It is imperative to note that there is no backlog permitted. This means that all orders must be finished on the same date as it is started with whatever cost. The additional resource capacity (overtime) can be employed if the available resources are overloaded.

Table 38 Simulation output of scenario 1 (with varying processing time)



ID	Task Name	Planned Leadtime	Actual Leadtime	Qty	Overtime (mins)	Flag
1	**Resource 1**					
2	90121	480 mins	475 mins	800	-5	🟨
3	90222	240 mins	239 mins	1200	-1	🟨
4	90223	240 mins	239 mins	1600	-1	🟨
5	**Resource 2**					
6	91122	480 mins	480 mins	3200	0	🟩
7	91223	240 mins	241 mins	600	1	🟥
8	91324	240 mins	237 mins	800	-3	🟨
9	**Resource 3**					
10	92123	480 mins	481 mins	1600	1	🟥
11	92324	240 mins	241 mins	400	1	🟥
12	92221	240 mins	225 mins	800	-15	🟨
13	**Resource 4**					
14	94124	480 mins	596 mins	1600	116	🟥
15	94221	240 mins	263 mins	400	23	🟥
16	94222	240 mins	253 mins	600	13	🟥

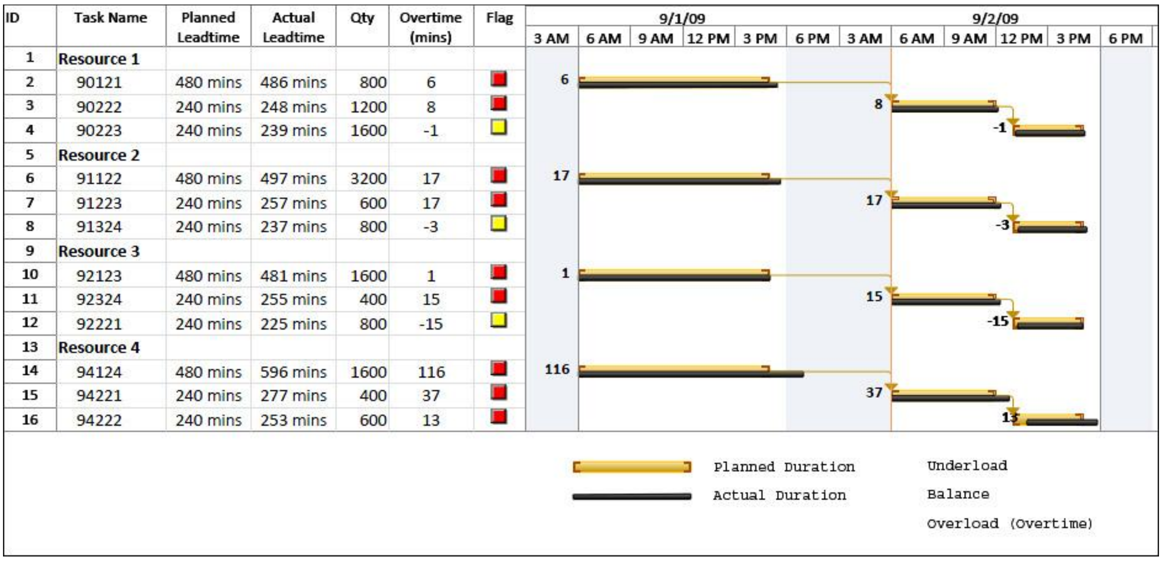
🟨 Underload
🟩 Balance
🟥 Overload (Overtime)

The simulation outputs of scenarios 1 are summarized in Table 38. Column *Overtime* indicates the discrepancy between planned and actual execution time. The minus value (-) means that a task can be completed earlier than planning (Under-load), whereas the positive value means that additional time (overtime) is necessary to execute the corresponding tasks completely. From the table, one can see clearly that the varying processing time gets some tasks finish earlier and some of them require longer time than that in work plan. For instance, at resource 3, task (POR) 9221 can be completed only in 225 minutes (or 15 minutes earlier), while at resource 4, to finish task 94124 it is required additional time as much as 2 hours.

This fact indicates that the discrepancy between planned and actual lead time increase relatively proportional to the fluctuation of processing time. In this context, production line (resource 4) has the highest discrepancy because of its high deviation.

The simulation result of scenario 2 is presented in Table 39. By adding stochastic disturbance (machine failure) during simulation, the actual execution time differs significantly from that in work plan. One can see that almost all of the productions tasks require longer execution time than the planned time which varies from 1 to 116 minutes. Moreover, the amount of overtime needed to complete each task also increases considerably.

Table 39 Simulation output of scenario 2 (with varying processing time and machine failure)



ID	Task Name	Planned Leadtime	Actual Leadtime	Qty	Overtime (mins)	Flag
1	Resource 1					
2	90121	480 mins	486 mins	800	6	🟥
3	90222	240 mins	248 mins	1200	8	🟥
4	90223	240 mins	239 mins	1600	-1	🟨
5	Resource 2					
6	91122	480 mins	497 mins	3200	17	🟥
7	91223	240 mins	257 mins	600	17	🟥
8	91324	240 mins	237 mins	800	-3	🟨
9	Resource 3					
10	92123	480 mins	481 mins	1600	1	🟥
11	92324	240 mins	255 mins	400	15	🟥
12	92221	240 mins	225 mins	800	-15	🟨
13	Resource 4					
14	94124	480 mins	596 mins	1600	116	🟥
15	94221	240 mins	277 mins	400	37	🟥
16	94222	240 mins	253 mins	600	13	🟥

Quantitatively, comparing with the scenario one, the total lead times required by Resource 1 (WCFXM01), Resource 2 (WCFXM02), Resource 3 (WCFXM03) and Resource 4 (WCFXM04) increase 2%, 3.4%, 1.5% and 1.2% respectively. This fact underlines the current assumption that the higher uncertainty in manufacturing system is, the more difficult an accurate production planning is made. This conclusion implies that in volatile environment one should prepare backup plan for unpredictable worse case. In term of machine reliability (failure), one may install backup machine or store more spare-parts, whereas in term of varying processing time, one may set more reasonable lead time or develop flexible production planning in which the adjustment can be applied anytime.

Chapter 7

MPS Implementation

In the previous chapters, almost steps of the closed-loop MPS has been discussed systematically. It is starting from maintaining SOP and demand management data in SAP, generating a tentative MPS using fuzzy multi-objective linear programming (FMOLP), running MRP to transform MPS into the detail planned orders and at the end simulating the execution of planned order using discrete event simulation (DES). Now, one comes to the last step that is re-optimization of the tentative MPS.

After simulation running, one may figure out that the tentative MPS is invalid and unrealistic to implement due to some reasons, for example, too much volatile production variables. In such situation, the adjustment must be carried out to make the tentative MPS feasible. To do so, the load leveling technique is employed to adjust the quantity and date configuration of tentative MPS. This chapter will discuss detail about the implementation of load leveling technique by using fuzzy multi-objective linear programming (FMOLP) approach. In addition, this chapter gives also illustration how to implement the complete closed-loop method using numerical example.

7.1 Symptoms of Master Scheduling Problems

According to *Sheikh* (2003) and *Proud* (1999), the symptoms of mismanaged master production shedule can be identified as follows.

1. Excessive unplanned overtime
2. Extensively front-loaded capacity plan. The capacity plan should be neither overloaded nor extensively front-loaded. The load should be reasonably level over the entire planning horizon.
3. Excessively short lead-times and frequent need for expediting

4. Excessive past due shop orders and purchase orders
5. Poor on-time delivery performance
6. Increase component (mostly, work-in-process) inventory

Since the manufacturing system assumes that there is no backlog allowed within a planning horizon, the most appropriate parameter to judge the feasibility of proposed MPS is the number of unplanned overtime. In this context, the tentative MPS is considered as valid and realistic if the actual overtime is equal or approximate to the planned overtime.

7.2 The Tentative MPS Optimization

As long as the proposed MPS is considered as infeasible, the system will re-optimize the tentative MPS until it turn into valid and realistic. In principle, the re-optimization adopts the load leveling concept, in which the load capacity of over-utilized resource is reduced and shifted to the under-utilized resource. To determine how many quantity of material to be shifted, it is used fuzzy multi-objective linear programming (FMOLP). The formulation of FMOLP will be discussed more detail on the next section.

7.2.1 Load Leveling Technique Using FMOLP

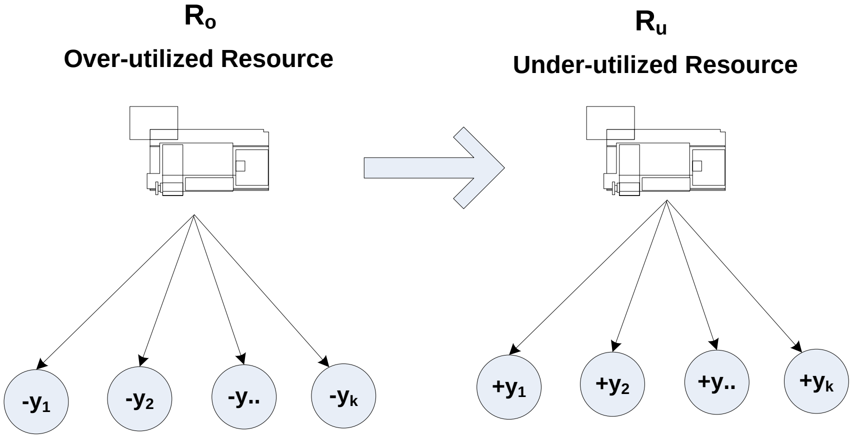


Figure 40 The illustration of load leveling problem

The principle idea of the load leveling technique is to level some quantities of material to be produced from the over-utilized resources to the under-utilized resources in the same period or previous period. This task becomes sophisticated as multiple resources, multiple products, and multiple periods are considered. The questions need to be addressed will be how many quantities and which products should be shifted, from/to which resources, and from/to which period. Inappropriate decision may cost poor performance of other criteria. For instance, moving too much quantity to the previous period (producing product early in advance) can lead to extensively front-loaded capacity or high inventory level. In addition, shifting some quantities to the less realiable resources may increase the number of defect products. The task becomes more complex as the conflicting multi-objectives are involved in the load leveling processes such as minimizing overall overtime and minimizing the potential decay products. Fortunately, an advanced optimization technique such as FMOLP can help us to solve such complicated problem.

To describe the FMOLP formulation for load leveling problem, let consider two resources: over-utilized resources (R_o) and under-utilized resources (R_u). Both resources has capability to produce material $k = 1,2,3,\ldots,K$. The problem is to determine how many quantities of material 1 (y_1) should be shifted from R_o to R_u , how many quantities of material 2 (y_2) should be shifted from R_o to R_u and so on, in a way that the overall overtime can be minimized and the excessive potential defect product can be avoided. Figure 40 gives illustration about the load leveling problem. The linear programming for the load leveling problem can be formulized as follows.

Find y_k $(k = 1,2,\ldots,K)$ to satisfy:

$$Min\ RT(y) = AC_o - \sum_{k=1}^{K} \frac{y_k}{U_{ko}}$$

$$Min\ DI(y) = \sum_{k=1}^{K} s_{ku} \times y_k$$

$$s.to.:$$

$$g(y) = \sum_{k=1}^{K} \frac{y_k}{U_{ku}} \le RC_u$$

$$y_k \le x_{ko}$$

$$k = 1,2,\ldots,K$$

$$y_k \ge 0 \quad integer\ and\ multiple\ of\ lot\ size \qquad (7.1)$$

Where,

RT Used load capacity at resource R_o after leveling

DI Quantity of potential defect product manufactured at resource R_u

y_k Quantity of item k (multiple of lot size) to be shifted from R_o to R_u

x_{ko} Planned quantity of item k to be manufactured at resource R_o

AC_o Actual load capacity at resource R_o (hrs)

RC_u Available load capacity at resource R_u (hrs)

U_{ko} Average of production rate at resource R_o to manufacture product k (unit/hrs)

U_{ku} Average of production rate at resource R_u to manufacture product k (unit/hrs)

s_{ku} Defect rate of product k at resource R_u

g This constraint ensures that the R_u does not turn into overloaded after leveling

Following the standard procedure described in Chapter 3.2, the FMOLP for lead leveling problem can be obtained through these steps.

Step 1:

If decision maker has fuzzy goal such that the objective functions $\widetilde{RT}(y), \widetilde{DI}(y)$ should be substantially more than or less than or equal to some aspiration level RT^*, DI^* respectively then the FMOLP for the load leveling problem can be formulated as follows.

Find $y_k(k = 1,2,\dots,K)$ *to satisfy:*

$$\widetilde{RT} = AC_o - \sum_{k=1}^{K} \frac{y_k}{U_{ko}} \lesssim RT^*$$

$$\widetilde{DI} = \sum_{k=1}^{K} s_{ku} \times y_k \lesssim DI^*$$

$$g(y) = \sum_{k=1}^{K} \frac{y_k}{U_{ku}} \leq RC_u$$

$$k = 1,2,\dots,K$$

$$y_k \geq 0\,, integer\ and\ multiple\ of\ lot\ size \qquad (7.2)$$

Step 2:

The lower and upper bound of aspiration level for objective used capacity and defect quantity are expressed as RT^0, RT^1, DI^0, DI^1 respectively. Those are obtained by solving the multi-objective problem as single objective using, each time, only one objective. The upper bound values RT^0 and DI^1 are reached if all available capacities at resource R_u are occupied, whereas RT^1 and DI^0 are met if no planned quantities are shifted from R_o to R_u.

Step 3:

Let assume that $Z_i^0\ and\ Z_i^1$ denotes the value of the objective function $Z_i(y)$ such that the degree of membership function is 0 or 1 respectively. Thus, the degree of membership function for objective used load capacity and defect product can be expressed as follows (see Figure 43).

$$\mu_{RT}(y) = \begin{cases} 1 & RT(y) \leq RT^0 \\ \dfrac{RT^1 - RT(y)}{RT^1 - RT^0} & RT^0 < RT(y) < RT^1 \\ 0 & RT(y) \geq RT^1 \end{cases} \tag{7.3}$$

$$\mu_{DI}(y) = \begin{cases} 1 & DI(y) \leq DI^0 \\ \dfrac{DI^1 - DI(y)}{DI^1 - DI^0} & DI^0 < DI(y) < DI^1 \\ 0 & DI(y) \geq DI^1 \end{cases} \tag{7.4}$$

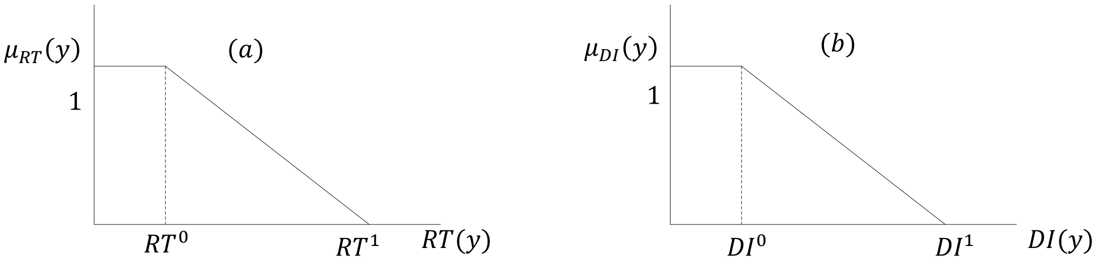


Figure 41(a) Non-increasing membership function for minimizing used load capacity (b) Non-increasing membership function for minimizing defect product

Step 4:

Let assume $w_i (i = 1,2)$ are the weighting coefficients that present the relative importance between the fuzzy objectives. Using the weighted additive model (Equation 2.32), the crisp linear programming equivalent to the FMOLP can be obtained as follows.

$$Maximize\ w_1\lambda_{RT}+w_2\lambda_{DI}$$

$$subject\ to.:$$

$$\lambda_{RT} \leq \mu_{RT}(y)$$

$$\lambda_{DI} \leq \mu_{DI}(y)$$

$$g(y) = \sum_{k=1}^{K} \frac{y_k}{U_{ku}} \leq RC_u$$

$$y_k \leq x_{ko}\ ;\ k = 1,2,..,K$$

$$y_k \geq 0\ , integer\ and\ multiple\ of\ lot\ size \tag{7.5}$$

Step 5:

Solving the crisp linear programming to obtain the optimum solution y^* uses DE algorithm.

7.2.2 Numerical Example

The following numerical example gives illustration how to level the proposed MPS using FMOLP method. Let considers two productive resources (under- and over-utilized resources, R_o and R_u respectively) with the load quantity, the varying processing time and the percentage of defect products as presented in Table 39. Both productive resources manufacture four types of products with standard lot size 10 units for all products. Each resource has available capacity 8 hours (480 minutes) per period and the availability of resources R_o , R_u is 95% and 98% respectively. Due to varying processing time and failure, after simulation run, it is found that the capacity required in resources R_o and R_u is 529 and 391 min respectively.

It implies that the overtime in over-utilized resource (R_o) is 49 min and the unused capacity in under-utilized resource (R_u) is 89 min. The task it to determine how many quantities and which products should be shifted from R_o to R_u such that overall overtime can be reduced while maintaining appropriate performance of other criteria. For the given case, the lower and upper bound of aspiration level for each objective can be computed as follows.

$$RT^0 = AC_o - RC_u = 529 - (480 - 391) = 440\,min$$

$$RT^1 = AC_o = 529\,min$$

$$DI^0 = 0\,SKU$$

$$DI^1 = 46\,SKU \quad \text{(Solved by ignoring objective used load capacity, } RT\text{)} \qquad (7.6)$$

Table 40 The production parameters of under- and over-utilized resources

	Resource	Product 1	Product 2	Product 3	Product 4
Load (unit)	R_o	280	650	20	380
	R_u	260	450	150	60
Processing time (s/lot size)[18]	R_o	360(20)	240(20)	180(20)	90(20)
	R_u	360(90)	240(90)	180(90)	90(90)
Percentage defect (%)	R_o	3	5	3	4
	R_u	4	4	2	2

The non-increasing membership function $\mu_{RT}(x)$ and $\mu_{DI}(x)$ can be expressed as follows.

$$\mu_{RT}(x) = \frac{529 - \{((y_1 \times 36) + (y_2 \times 24) + (y_3 \times 18) + (y_4 \times 9))/60\}}{529 - 440} \qquad (7.7)$$

[18] 360(20) means average: 360 and standard deviation: 20

$$\mu_{DI}(x) = \frac{\{(y_1 \times 0.04) + (y_2 \times 0.04) + (y_3 \times 0.02) + (y_4 \times 0.02)\}}{46 - 0} \qquad (7.10)$$

Assuming that $w_1 = 0.7\ and\ w_2 = 0.3$, thus the crisp single objective linear programming for the load leveling problem can be formulized as follows.

$Maximize\ \ 0.7\lambda_{RT} + 0.3\lambda_{DI}$

$subject\ to.$:

$\lambda_{RT} \le \mu_{RT}(y)$

$\lambda_{DI} \le \mu_{DI}(y)$

$g(y) = \{(y_1 \times 36) + (y_2 \times 24) + (y_3 \times 18) + (y_4 \times 9)\}/60 \le 89$

$y_1 \le 280$

$y_2 \le 650$

$y_3 \le 20$

$y_4 \le 380$

$y_k \ge 0\,, k = 1,2,3,4$ (7.11)

The differential evolution algorithm (see Chapter 4.5) is used to solve this linear programming. The optimal solution for the above formulation is obtained as follows.

$y_1 = 0,\ y_2 = 100,\ y_3 = 20,\ y_4 = 20$

$RT = 480,\ DI = 5$

The load capacity of over-utilized resource (R_o) is deducted as 49 min so that the current requirement capacity for R_o becomes 480 min. It means that the overtime is no longer required. Due to this adjustment, the load capacity in under-utilized resource (R_u) increases up to 440 min. However, the load capacity of resource R_u does not still exceed the standard available capacity. Now, one can see that this technique can improve the resource utilization considerably.

7.2.3 Optimization Scenario

In the previous section, the problem to determine quantities and types of products need to be shifted has been addressed. This chapter discusses how to choose over-utilized resource (R_o) and under-utilized resource (R_u), such that the leveling can be carried out effectively for all planning horizon and ultimately the better solution can be obtained. In term of this selection, the optimization is grouped into four levels as follows.

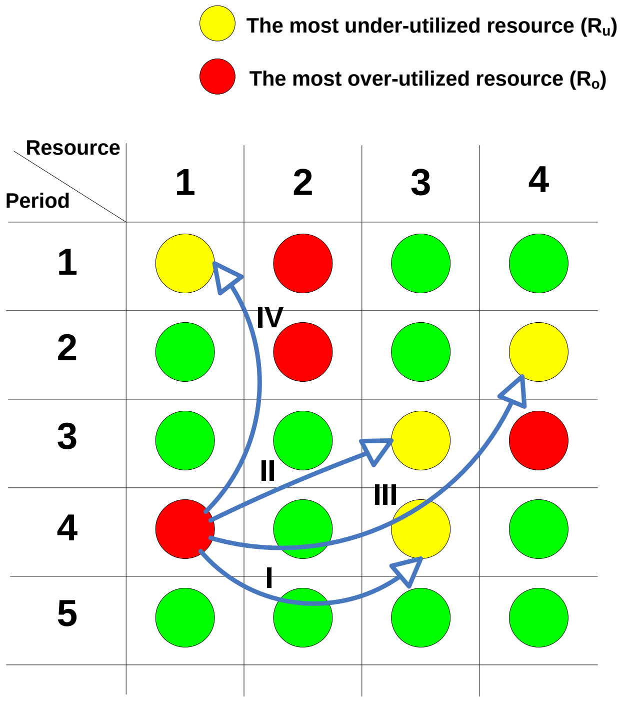

The most under-utilized resource (Ru)
The most over-utilized resource (Ro)
Resource
Period
1
2
3
4
1
2
3
4
5
IV
II
III
I

Figure 42 The illustration of optimization level

1. *Optimization Level I*

   Both of R_o and R_u are selected from the resource pool in the same period (p). R_o is the most over-utilized resource at planning period p, while R_u is one whose amount of load capacity is smallest. For example, in the planning period p, the resource pool consists of resources R_1, R_2, R_3, R_4 whose load capacities are 360, 520, 485, 340 minutes respectively. In this example, R_2 will be R_o and R_4 will be R_u. Since the production load is shifted to another resource in the same period, this optimization level does not lead to poor performance of other criteria such as inventory level and customer satisfaction.

2. *Optimization Level II*

   Unlike the previous one, this optimization shifts some load capacities from the current period p to the previous period $p-1$. The over-utilized resource (R_o) is selected from period p and under-utilized resource (R_u) is picked up from planning period $p-1$. The optimization level II may affect other criteria performance, particularly, inventory level.

3. *Optimization Level III*

   This optimization shifts some load capacities in the current period p to the previous period $p-2$. The over-utilized resource (R_o) is selected from period p and under-utilized resource (R_{u}) is picked up from planning period $p-2$. This optimization level potentially increases inventory level and may also affect other criteria.

4. *Optimization Level IV*

   This optimization shifts some load capacities in the current period p to the previous period $p-3$. The over-utilized resource R_o is selected from period p and under-utilized resource R_u is chosen up from planning period $p-3$. Note that this type of optimization level may increase inventory level considerably and affects other criteria.

The optimization level I is considered as ideal adjustment because it does not influence other criteria performance. To provide clear picture, Figure 42 visualize the selection of over- and under-utilized resource for each optimization level. In the MPS development, the optimization will be carried out ascending from level I, II, III and IV. An optimization level can be executed if its lower level has been employed. It means that the execution of optimization level II will include optimization level I; the execution of optimization level III must include optimization level I and II; the execution of optimization level IV must include optimization level I, II and III.

7.2.4 Code Implementation

In principle, the proposed load leveling technique encompasses three main steps: select appropriate over-utilized resource (R_o) and under-utilized resource (R_u) , compute the load capacity to be leveled, and update the load capacity on each resource. Although it seems simple, this process can be irritating for a scheduler, particular, if it includes long planning horizon, large resources and hundred or thousand products. Therefore, a code which is carried out those three steps is developed to speed up the load leveling process.

```
Algorithm: **Load leveling technique**
Input: a tentative MPS solution, optimization level (s)

$s \in \{0, 1, 2, 3\}$
$y_k$ ← [ ]
**For** $p$ ← 1 to SizeOf(MPS)
    Select $R_o$, which is resource with maximum load capacity in planning period $p$
    Select $R_u$, which is resource with minimum load capacity in planning period $p - s$
    Acquire the production data of resource $R_o$ and $R_u$
    Compute $y_k$
    **If** ($y_k != []$) **Then**
        Update the tentative MPS accordingly **End If**
End For
**Return**: feasible MPS
```

Inputs of the load leveling process are the tentative MPS and the optimization level. For each period of MPS, the resources R_o and R_u will be selected according to the entered optimization level. The value of optimization level will be 0, 1, 2 and 3 which represents optimization level I, II, III and IV respectively. Once resources R_o and R_u are determined, the production data of corresponding resources such as processing time, actual overtime, standard available capacity, and so on, are acquired. Based on the data, the leveled quantities (y_k) are computed. Thus, the tentative MPS is modified accordingly.

7.3 Standard Operating Procedure of MPS Creation

In term of good manufacturing practice, the standard operating procedure is defined as a written document/instruction detailing all steps and activities of a process or procedure. Unlike the business process, which encompasses activities of all involving actors, the standard operating procedure lists only activities that should be carried out by manpower (human). This sub-chapter summarizes all activities that must be performed to generate valid and realistic master production schedule (MPS) by using the proposed closed-loop methodology. The following steps will be standard operating procedure for scheduler to create MPS (Figure 43).

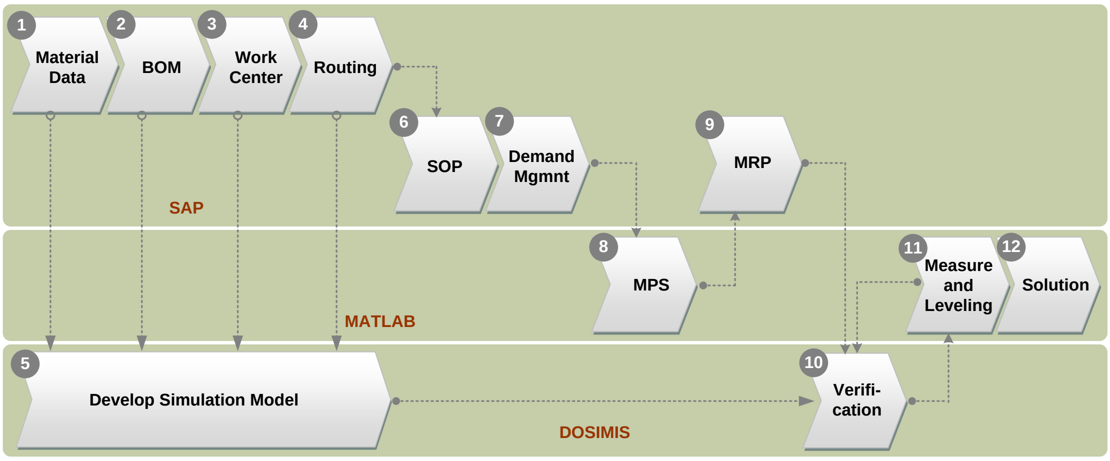


Figure 43 Standard operating procedure of MPS creation

P1. Register material data in SAP. The materials include part, sub-assembly, assembly and final products. All data related to the material such as basic information, planning and scheduling, inventory and sales must be entered. See chapter 5.4.1.1 for detail description.

P2. Bill of material (BOM) needs to be defined if the material is made of multiple parts or sub-/assembly. This data is essential for material requirement planning (MRP) process.

P3. Register work centers, in which the material will be processed, and the corresponding available capacity. The work center can be machine, labor, or group of machine/labor. See chapter 5.4.1.3 for detail explanation.

P4. Maintain the routing data in SAP. The routing determines in which productive resource and sequence the corresponding material is manufactured (See Chapter 5.4.1.4).

P5. Develop simulation model of the real manufacturing system in DOSIMIS. The master production data in previous steps (material, BOM, work center and routing) will be the source data of model. See chapter 6 for detail description.

P6. Maintain Sales and Operation Planning (SOP) in SAP. In this step, the medium-/long-term production plan will be created. See chapter 5.4.2 for detail explanation.

P7. Maintain demand management in SAP. All planned independent requirement will be organized in this step and subsequently is fed to MPS (See Chapter 5.4.3).

P8. Generate the tentative MPS using FMOLP and genetic algorithm. The generated MPS must satisfy the DM's goals, which are represented by four objectives (*EI, RNM, BSS, and OC*). This topic has been discussed comprehensively in chapter 3 and chapter 4. The algorithm of MPS creation is implemented in the computation software, MATLAB.

P9. The tentative MPS, which is outcome of MPS generator, is exported to SAP for driving material requirement planning process. In this step, scheduler simply runs the MRP function in SAP (See chapter 5.4.4).

P10. Verify the feasibility of tentative MPS. This verification is carried out by simulating the execution of planned orders (POR), which are essentially the representation of master production schedule in the lower level planning. The POR delay can be interpreted that the tentative MPS is infeasible.

P11. Analyze the outcome of simulation and compute the performance measures. In this case, the four objectives (*EI, RNM, BSS, and OC*) are used as performance indicator. The MPS is considered as valid and realistic if the performance measure approximates the DM's targets defined in step 8. The adjustment (load leveling) of the proposed MPS is carried out in case the DM's targets are not met.

P12. Repeat the step 10 and 11 until the DM's targets are satisfied.

7.4 MPS Creation

To evaluate performance of the proposed methodology, an example of MPS creation is presented. Using the above standard procedure, the process of MPS creation can be described as follows.

1. Assuming that step P1 – P7 has been carried out (see the corresponding chapters) and the input parameters of MPS are obtained as presented in Table 41.

Table 41 The input parameters of MPS

Parameters	Notation	Value
Number of products	K	4 (P-403, P-404, P-405, P-406)
Number of production lines	R	4 (WCFXM01, WCFXM02, WCFXM03, WCFXM04)
Planning horizon	P	20 days
Standard lot size	BS_{kp}	10 unit for all products and all planning periods
Average of production rate	UR_{rp}	Table 35
Initial inventory	OH_k	Zero for all products
Safety stock level	SS_{kp}	200 unit for all products and all planning periods
Gross requirement	GR_{kp}	Table 42
Available capacity	AC_{rp}	8 hours/period for each resource
Maximum overall overtime	OC_{tot}	4 hours during planning horizon
Allowed maximum overtime	OL_{max}	4 hours/ period for each resource
Aspiration Levels	EI	$Z_1^0 = 50,\ Z_1^1 = 5250$
	RNM	$Z_2^0 = 0,\ Z_2^1 = 3200$
	BSS	$Z_3^0 = 10,\ Z_3^1 = 2000$
	OC	$a = b = \ Z_4^1 = 4$
Relative importance	w_i	$w_1 = 3, w_2 = 3,\ w_3 = 3, w_4 = 1$
Assumption	-	Make-to-stock (MTS) & no backlog
Production layout	-	Figure 13 & Figure 38
Percentage of defect product	-	0% for all resources

Table 42 The gross requirement

Day	Due Date	P-403	P-404	P-405	P-406	Day	Due Date	P-403	P-404	P-405	P-406
1	01.07.2009	1300	1250	1390	580	11	15.07.2009	1400	1000	1610	680
2	02.07.2009	930	1360	1360	530	12	16.07.2009	340	1200	1460	690
3	03.07.2009	1100	1660	1450	1650	13	17.07.2009	420	320	1310	960
4	06.07.2009	720	1490	1420	840	14	20.07.2009	760	1440	1440	500
5	07.07.2009	740	590	1270	230	15	21.07.2009	880	1300	1070	280
6	08.07.2009	1030	1520	1710	210	16	22.07.2009	940	1030	1670	560
7	09.07.2009	970	1570	1540	1710	17	23.07.2009	1190	1010	1270	1360
8	10.07.2009	1500	1500	1700	1810	18	24.07.2009	720	840	1320	2040
9	13.07.2009	850	1760	1360	1660	19	27.07.2009	720	1240	1710	920
10	14.07.2009	1610	1380	1240	140	20	28.07.2009	580	1180	1530	320

2. The following step (P8) is to generate the tentative MPS using fuzzy multi-objective linear programming (FMOLP). In this stage, the decision makers should create a MPS which presents their goal that is expressed in the performance measures (EI, RNM, BSS, OC). The EI denotes for *Ending Inventory*, RNM stands for *Requirement Not Met* which represents the customer service level, BSS means *Below Safety Stock* which indicates the inventory target level, and OC represents *Over-capacity (Overtime)* or the additional resource capacity required to complete the MPS. Based on the given input parameters, the best MPS are obtained at the overall degree of satisfaction $(\mu_D = 0.8943)$ with the achievement level of each objective as follows.

$$EI^*\ (Z_1) = 1201\ \ ; RNM^*\ (Z_2) = 160\ \ \ ; BSS^*\ (Z_3) = 172\ \ \ ; OC^*\ (Z_4) =\ 240\ min$$

These objectives are considered as the expected performance measures or target that the decision makers want to achieve. The planned requirement capacities on each productive resource along planning periods that is consumed by the proposed MPS (x^*) are visualized by Figure 44. The graph shows that the total requirement capacities are not distributed equally among the available resources. Some resources are over-loaded, some of them have been fully occupied, and the rest are partly occupied (under-utilized).

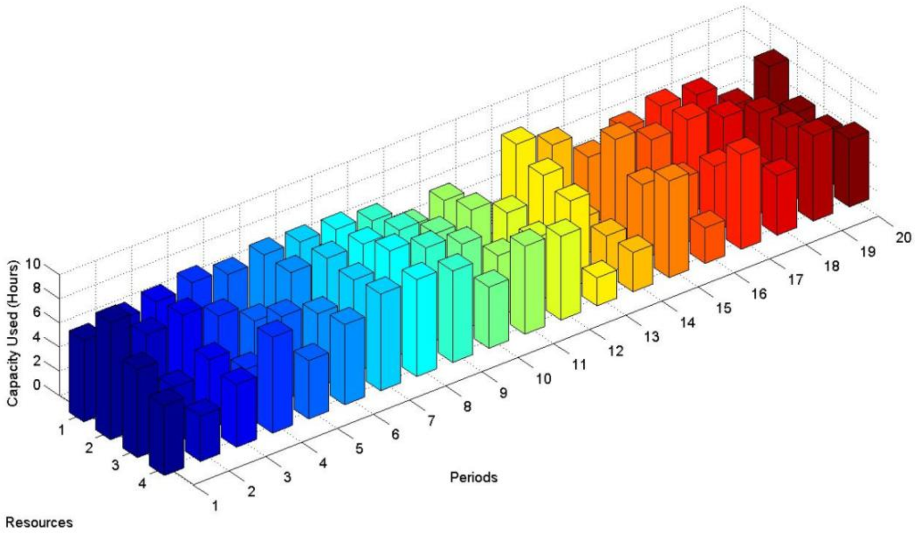
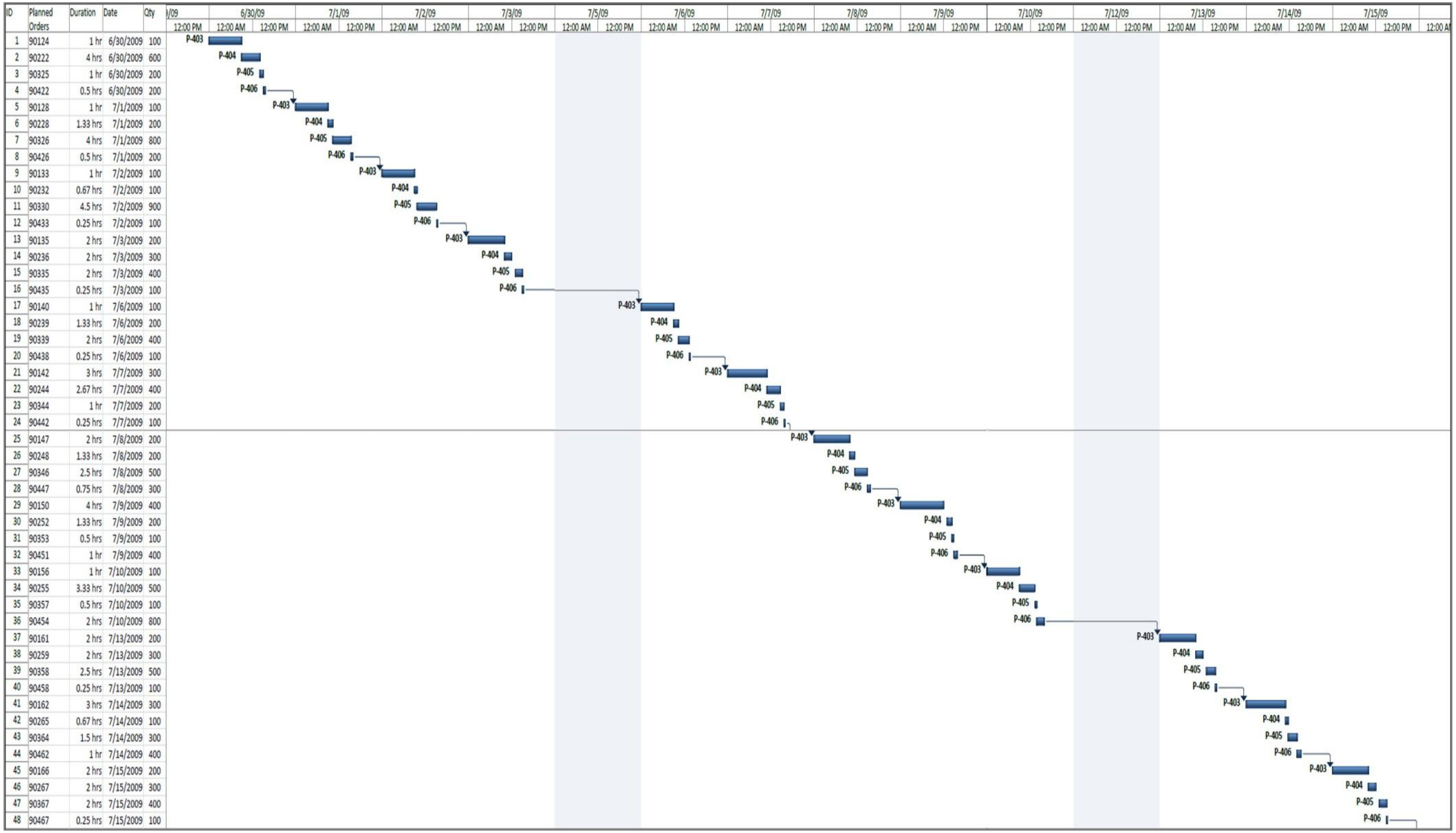


Figure 44 The profile of planned requirement capacity

3. The next step (P9) is to export the MPS solution (x^*) into SAP and run MRP based on the tentative MPS. The outcome of MRP is list of planned orders (POR) which must be worked in the existing productive resources (WCFXM01, WCFXM02, WCFXM03, and WCFXM04). The following are partial example of production schedule assigned to WCFXM01 (Table 43).

Table 43 Gantt-Chart of Resource 1 (WCFXM01)



ID	Planned Orders	Duration	Date	Qty
1	90124	1 hr	6/30/2009	100
2	90222	4 hrs	6/30/2009	600
3	90325	1 hr	6/30/2009	200
4	90422	0.5 hrs	6/30/2009	200
5	90128	1 hr	7/1/2009	100
6	90228	1.33 hrs	7/1/2009	200
7	90326	4 hrs	7/1/2009	800
8	90426	0.5 hrs	7/1/2009	200
9	90133	1 hr	7/2/2009	100
10	90232	0.67 hrs	7/2/2009	100
11	90330	4.5 hrs	7/2/2009	900
12	90433	0.25 hrs	7/2/2009	100
13	90135	2 hrs	7/3/2009	200
14	90236	2 hrs	7/3/2009	300
15	90335	2 hrs	7/3/2009	400
16	90435	0.25 hrs	7/3/2009	100
17	90140	1 hr	7/6/2009	100
18	90239	1.33 hrs	7/6/2009	200
19	90339	2 hrs	7/6/2009	400
20	90438	0.25 hrs	7/6/2009	100
21	90142	3 hrs	7/7/2009	300
22	90244	2.67 hrs	7/7/2009	400
23	90344	1 hr	7/7/2009	200
24	90442	0.25 hrs	7/7/2009	100
25	90147	2 hrs	7/8/2009	200
26	90248	1.33 hrs	7/8/2009	200
27	90346	2.5 hrs	7/8/2009	500
28	90447	0.75 hrs	7/8/2009	300
29	90150	4 hrs	7/9/2009	400
30	90252	1.33 hrs	7/9/2009	200
31	90353	0.5 hrs	7/9/2009	100
32	90451	1 hr	7/9/2009	400
33	90156	1 hr	7/10/2009	100
34	90255	3.33 hrs	7/10/2009	500
35	90357	0.5 hrs	7/10/2009	100
36	90454	2 hrs	7/10/2009	800
37	90161	2 hrs	7/13/2009	200
38	90259	2 hrs	7/13/2009	300
39	90358	2.5 hrs	7/13/2009	500
40	90458	0.25 hrs	7/13/2009	100
41	90162	3 hrs	7/14/2009	300
42	90265	0.67 hrs	7/14/2009	100
43	90364	1.5 hrs	7/14/2009	300
44	90462	1 hr	7/14/2009	400
45	90166	2 hrs	7/15/2009	200
46	90267	2 hrs	7/15/2009	300
47	90367	2 hrs	7/15/2009	400
48	90467	0.25 hrs	7/15/2009	100

4. The subsequent step (P10) is to export the planned orders and simulate the execution of PORs. As the development of tentative MPS, the corresponding processing times (production rates) are assumed deterministic and constant all the time. In particular case, this assumption may lead to inappropriate decision, especially, in manufacturing system whose the processing time is varying considerably. The simulation attempts to approach closer to the dynamic behavior of real world by employing the stochastic processing time instead of constants ones. Let assume that the times needed to process products at certain productive resource are random normal distribution with mean and standard deviation as presented in Table 37. In addition, the simulation model also attempts to verify the reliability of MPS against disturbance. For this purpose, the random machine failures are applied on each productive resource (Table 33).
5. The next step (P11) is to compute the objective function (performance measure) of MPS after simulation running. As a result, the performance measures are obtained as follows.

$$EI_0 = 1201 \ , RNM_0 = 160 \ \ , BSS_0 = 172 \ \ , OC_0 = 740 \, min$$

The result shows that the excessive unplanned overtimes are required to complete the proposed MPS. According to *Proud* (1999) and *Sheikh* (2003), this symptoms denotes that the proposed MPS get potentially problem. Since the performance measures do not satisfy the DM's targets, particularly, the overtime objective, the tentative MPS must be re-optimized. To do so, the *Optimization Level One* is carried out for first MPS adjustment. As a result, the system suggested to level some load capacities as tabulated in Table 44.

Table 43 The MPS adjustment according to optimization level I

Origin		Destination		Quantities to be leveled (units)			
Period	**Resource**	**Period**	**Resource**	**P-403**	**P-404**	**P-405**	**P-406**
1	WCFXM02	1	WCFXM04	10	150	160	0
1	WCFXM02	1	WCFXM01	0	20	30	0
3	WCFXM02	3	WCFXM04	10	20	10	0
4	WCFXM04	4	WCFXM03	20	10	10	0
4	WCFXM01	4	WCFXM03	0	0	10	0
6	WCFXM02	6	WCFXM03	0	10	10	0
7	WCFXM04	7	WCFXM03	20	0	10	0
7	WCFXM02	7	WCFXM01	0	10	0	20
8	WCFXM03	8	WCFXM01	10	0	0	0
9	WCFXM02	9	WCFXM01	40	20	0	20
9	WCFXM02	9	WCFXM04	0	20	10	20
13	WCFXM01	13	WCFXM04	40	130	10	0
13	WCFXM02	13	WCFXM04	20	10	10	0
15	WCFXM02	15	WCFXM01	60	10	0	20

15	WCFXM04	15	WCFXM03	50	0	10	10
17	WCFXM04	17	WCFXM03	50	0	0	20
17	WCFXM02	17	WCFXM03	10	0	0	0
19	WCFXM04	19	WCFXM01	20	10	10	0

6. After the first MPS adjustment is conducted, the previous step (P10) to run simulation is repeated. Note that the same production variables (processing time and machine failure) are applied on every simulation.
7. The step (P11) is carried out again and the performance measures are obtained as below.

$$EI_1 = 1201 \ , RNM_1 = 160 \, , BSS_1 = 172 \ \ , OC_1 = 304 \, min$$

One can see that the overtime objective can be reduced considerably from 740 to 304 minutes; however, this is higher than the DM's targets. It means that the further optimization level II is required and as a result, the following adjustments are suggested to do (Table 45).

Table 44 The MPS adjustment according to optimization level II

Origin		Destination		Quantities to be leveled (units)			
Period	Resource	Period	Resource	P-403	P-404	P-405	P-406
3	WCFXM02	2	WCFXM04	0	0	20	0
7	WCFXM04	6	WCFXM04	0	10	20	0
7	WCFXM02	6	WCFXM04	10	0	0	0
8	WCFXM03	7	WCFXM04	150	10	0	0
8	WCFXM02	7	WCFXM04	60	0	20	0
9	WCFXM03	8	WCFXM04	20	20	30	0
13	WCFXM02	12	WCFXM01	10	10	0	20
19	WCFXM04	18	WCFXM01	10	40	10	0

8. After the second MPS adjustment is carried out, the simulation is run again (P10).
9. Thus, the performance measures are computed (P11) and the result is obtained as follows.

$$EI_2 = 1215 \ , RNM_2 = 160 \, , BSS_2 = 171 \ \ , OC_2 = 266 \, min$$

The overtime objective can be decreased, but as consequence, the inventory objective rise slightly. This is reasonable because in optimization level II, some quantities (load) are moved to the previous period. Now, all performance measures are close to the DM's targets. At this stage, the choices to stop or continue for further optimization level depend on the decision makers. Note that the further optimization level (III & IV) may help to reduce overtime significantly, however, they may lead to the poor performance of other criteria, particularly, the inventory objective.

7.5 MPS Evaluation

In the previous section, the re-optimization of tentative MPS has been stopped at the second level because the decision makers have satisfied with the obtained outcome. To provide better overview about the implication of optimization levels, the further optimization level (III & IV) also carried out to solve the above problem. Table 46 summarizes the outcome of MPS adjustment associating with the optimization level.

Table 45 The performance improvement according to the optimization level

No.	Simulation	Adjustment	Remark	EI	RNM	BSS	OC (min)
1	No	No	DM’s target	1201	160	172	240
2	Yes	No	Before adjustment	1201	160	172	740
3	Yes	Yes	Optimization Level I	1201	160	172	304
4	Yes	Yes	Optimization Level II	1215	160	171	266
5	Yes	Yes	Optimization Level III	1261	160	169	88
6	Yes	Yes	Optimization Level IV	1248	160	177	8

Due to varying processing time and unpredictable event such as machine failure, the proposed MPS is unrealistic to implement or DM’s targets cannot be met. This is indicated by the excessive unplanned overtime (740 min). Therefore, re-optimization of the proposed MPS is necessary. As described in the previous sub-chapter, the re-optimization is carried out in several levels.

The *optimization level I* is able to reduce the unplanned overtime almost half without interfering other criteria. However, this achievement is not enough to satisfy DM’s targets. The *optimization level II*, which shift some load capacities to the previous period, can drop successfully the unplanned overtime close to the planned overtime. The *optimization level III* reduces overtime considerably but in the same time, it make inventory level rise. The rest of optimization level indicates the same result, where the overtime can be forced down but it bring implication to the increasing of inventory level. In the context of MPS development, this again underlines that the inventory level and overtime are conflicting objectives; therefore, the appropriate trade-off must be performed to ensure the optimum solution.

This simulation model considers a volatile manufacturing system which is indicated by frequent breakdown (machine failure) that occurs 1 – 2 times a day. In such uncertain environment, relying only on the reactive action during execution phase may lead to the poor resource allocation and causes much unnecessary delay. Therefore, a realistic and optimum planning, which takes into account the prediction of breakdown in its development’s phase, is inevitable.

In this context, one can see that the prediction of uncertainty plays important role in MPS creation using this methodology. Fail to acquire accurately information about the dynamic behavior of manufacturing system may make the planning output even getting worse.

In spite of the circumstance, it is proved that the proposed methodology is able successfully to improve the MPS quality. However, it is acknowledged that the given model may be only a simplified reality of manufacturing system. More sophisticated model should be developed to examine further the robustness of this methodology. The underlying cause of uncertainty can also be extent; not only involving random breakdown and stochastic processing time but also other unpredictable occasions due to limited buffer space, labor shortage, material shortage, etc. In addition, the applicability of this methodology in various environments with different grade of uncertainty may be worthwhile to investigate as well.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

Uncertainty is one of root causes of the production planning inaccuracy. This study has proved that the underlying causes of uncertainty such as varying processing time, machine failure, etc have considerable implication on the master production schedule (MPS). Ignoring them may lead to incorrect decision and ultimately make poor production and delivery performance. Through simulation approach, this study has been successfully able to estimate the effect of uncertainty and subsequently attempts to diminish its effect on the proposed MPS. Concerning to the development of valid and realistic MPS, this research has brought us to the valuable conclusions as follows.

1. The fuzzy multi-objective linear programming (FMOLP) can be surely employed to solve the MPS problem. Moreover, with certain parameter configurations, the FMOLP is able to yield better achievement level of objectives than non-fuzzy solution. The FMOLP approach is recommended, particularly, for a case where the objective functions are imprecise and can only be stated within certain aspiration level.

2. The FMOLP has ability to determine intelligently how much, when, and where the additional capacities (overtimes) are required such that the inventory can be reduced without affecting customer service level. This study has also shown that the overtime profile fluctuates in parallel with the customer demand variation. This highlights that the FMOLP is also suitable for handling the MPS creation with high varying customer demand.

3. In term of MPS development, the inventory objective (EI) contradicts with customer satisfaction (RNM) and inventory target objective (BSS). The increasing of achievement level of inventory objective is balanced with the poor performance of customer satisfaction and inventory target objectives (Figure 16).

4. The genetic algorithm can solve efficiently the crisp single objective model equivalent to the FMOLP. This research reinforces the Soares's finding (2008) that the GA can be applied to real industrial master planning problems.

5. Concerning to solving the MPS problem, the genetic algorithm is more efficient in term of processing time than the differential evolution (DE) algorithm. However, in term of overall degree achievement, the differential evolution is more reliable than genetic algorithm.

6. Through iteration process, the closed-loop MPS can bridge the drawback of approach proposed by *Kochhar* (1998), *Knowledge-based system approach*, which does not take into account unpredictable event (uncertainty) in manufacturing system (execution level). The similar framework may also be employed to improve other production planning functions.

7. The proposed information system, which is integration of three systems (Dosimis-3, SAP R/3 and MATLAB), enables us to realize the closed-loop planning concept. SAP eases us to implement production planning logic such as SOP, demand management and MRP logic. MATLAB helps us to apply intelligence optimization technique and analyze easily the statistic data, while Dosimis-3 enable us to simulate the execution of planned orders. However, the IT expert skill is required to integrate those systems in professional and reliable manner if this methodology wants be implemented for real industrial case.

8. This study reveals that Dosimis-3 can be potentially used to enhance the functionality of Production Planning and Control of SAP because both of them have the similar mechanism for operation execution. The operation in SAP is triggered by a production order (PO), while the operation in Dosimis-3 can be driven by a work plan. If the PO and work plan can be mapped properly, then the production planning generated by SAP can be verified before the plan is released to real manufacturing system.

9. Using discrete event simulation, it can be proved that the underlying causes of uncertainty such as varying processing time and machine failure have considerable implication for the production lead time. In general, the uncertainty make the actual lead time be longer than the planned lead time. The higher the degree of uncertainty is, the larger discrepancy is found between the actual and planned lead time.

10. Through the simulation of POR execution, this study also reveals that the underlying cause of uncertainty should be taken into account as MPS development. Ignoring the uncertainty lead to the unrealistic MPS which is indicated by excessive unplanned overtime.

11. The load leveling technique using fuzzy multi-objective linear programming (FMOLP) seems working effectively to determine the quantity of material should be shifted from over-utilized resource to under-utilized resource. In addition, the approach for selection of under- and over-utilized resources, which is divided into four optimization levels, seems quite reliable to reduce the unplanned overtime without obstructing too much other criteria.

12. Using the propose load leveling technique, it is found that moving some amount of load quantities to previous period can keep the tentative MPS be valid and realistic under uncertain environment. It implies that if the manufacturing environment is quite volatile, it is suggested to produce material in advance as long as the capacity is available. However, one must note that producing materials in earlier period than they are consumed will increase inventory cost.

13. Last but not least, in general the closed-loop MPS, which incorporates fuzzy multi-objective linear programming (FMOLP) and simulation approach, is able definitely to develop valid and realistic MPS under uncertain environment. However, it is admitted that this methodology may be not suitable for stable environment in which uncertainty events are insignificant and ignorable.

8.2 Future Work

In principle, the proposed methodology can successfully improve the feasibility of master planning. However, the author realizes that the further improvement is necessity to make this concept applicable for complex industrial practice. Therefore, the author underlines some points that may be worthwhile for future research.

1. Other manufacturing restrictions such as setup time, product sequencing, production priorities and backlog should be included in the MPS mathematical model. Moreover, the model can be also extent by considering multiple products with different importance.

2. Further study may consider objective functions with actual cost, instead of quantity units, like production cost, inventory cost, inventory cost, and losses for not meeting demand.

3. Fuzzy customer demand due to forecasting error and imprecise production rate should be also expressed in the FMOLP model using fuzzy coefficient of objective functions. For this kind of model, either interactive fuzzy programming or probabilistic linear programming can be employed to solve.

4. The MPS problem for more complex production scenario, which represents the real manufacturing case, may be attempted to be solved using the proposed methodology.

5. The simulation model should be improved by including more frequently real issues in a manufacturing system such as bottle neck, shift-time, and work-in-process. In addition, more underlying causes of uncertainty such as labor & material shortage, random defect product, varying queue time and setup time may be modeled as well.

6. It may be worthwhile to try other GA configurations in order to get the most optimum solution in efficient manner. The implementation of differential evolution algorithm in production planning also needs to be considered as it shows better reliability in spite of less efficiency.

7. The data exchange between Dosimis-3 and SAP should be automated through SAP-BAPI (Business Application Programming Interface) so that the simulation model structure can be build automatically based on the master data in SAP. In addition, the planned order data in SAP and the work plan data in Dosimis-3 should be mapped more detail and properly.

References

[1] Amid, A., Ghodsypour, S. H., and O'Brien, C. (2005). Fuzzy multiobjective linear model for supplier selection in a supply chain. *International Journal Production Economics*, 104, 394-407.

[2] Arosio, M., and Sianesi, A. (1993). A heuristic algorithm for master production schedule generation with finite capacity and sequence dependent setup. *International Journal of Production Research*, Vol. 31 No.3, pp. 531-553.

[3] Aytug, H., Khouja, M., and Vergara, E. (2003). Use of genetic algorithms to solve production and operations management problems: a review. *International Journal Production Research*.

[4] Bäck, T. (1996). *Evolutionary algorithms in theory and practice.* New York: Oxford Univ. Press.

[5] Bellman, R. E., and Zadeh, L. A. (1970). Decision-making in a fuzzy environment. *Management Science*, Vol. 17 No. 4.

[6] Berry, W. L., Vollmann, T. E., and Whybark, D. C. (1987). *Master Production Scheduling: principle and practice.* Newbury, New Hampshire: APICS.

[7] Brown, L. (2000). *Integration Models: Templates for Business Transformation.* USA: Sams.

[8] Buck-Emden, R., and Galimow, J. (1996). *Die Client/Server-Technologie des SAP-Systems R/3.* Addison-Wesley.

[9] Chambers, L. (1995). *Practical handbook of genetic algorithms.* Boca Raton: CRC Press.

[10] Chen, I. J. (2001). Planning for ERP system: analysis and future trend. *Business Process Management Journal*, Vol. 7, No. 5, pp. 374-386.

[11] Chen, S. H. (1985). Ranking fuzzy number with maximizing set and minimizing set. *Fuzzy sets and systems*, 17 pp. 113-129.

[12] Chu, S. C. (1995). A mathematical programming approach towards optimized master production scheduling. *International Journal of Production Economics*, Vol. 38 No. 2-3, pp. 269-279.

[13] Chung, S. H., and Lee, T. R. (1989). -. *Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications* (pp. pp.127-32). Amsterdam: Elsevier Science.

[14] De Jong, K. A. (1975). *An analysis of the behaviour of a class of genetic adaptive systems.* Phd Thesis, University of Michigan, Michigan.

[15] Freedendall, L. D., and Lea, B. R. (1997). Improving the product mix heuristic in the theory of constraints. *International Journal of Production Research*, Vol. 35 No. 6, pp. 1535 - 1544.

[16] Garey, M., and Johnson, D. (1979). *Computer, complexity and intractability. A guide to theory of NP-Completeness.* Freeman, San Franscisco, USA.

[17] Gonzalez, J. J., and Reeves, G. R. (1983). Master production scheduling: a multiple-objective linear programming approach. *International Journal of Production Research*, Vol. 21 No. 4, pp. 553-562.

[18] Gray, C. D., and Landvater, D. V. (1989). MRP II Standard System. *Oliver Wright Limited Publications, Essex Junction, VT* .

[19] Grefenstette, J. J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics* , SMC-16(1): 122-128.

[20] Gumaer, R. (1996). Beyond ERP and MRP II. *IIE Solution* , Vol. 29 No.9, September, pp.32-6.

[21] Heizer, J. H., and Render, B. (2006). *Operations Management.* Upper Saddle River, New York: Pearson Prentice Hall.

[22] Higgins, P., and Browne, J. (1992). Master production scheduling: a concurrent planning approach. *Production Planning & Control*, Vol. 3, no.1, 2-18.

[23] Hill, J. A., Berry, W. L., and Leong, G. K. (2000). Master production scheduling in capacitated sequence-dependent process industries. *International Journal of Production Research*, Vol. 38 No. 18, pp.4743-4761.

[24] Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor, MI: The university of Michigan Press.

[25] Hopp, W. J., Spearman, M. L. (2008). *Factory Physics.* New York: 3rd Edition, McGraw Hill.

[26] Kahraman, C. (2008). *Fuzzy Multi-Criteria Decision Making. Theory and Application with Recent Development.* New York, USA: Springer.

[27] Keung, K. W., Ip, W. H., and Chan, C. Y. (2001). An enhanced MPS solution for MPS using GAs. *Integrated Manufacturing System*, Vol. 12 No. 5, pp. 351-359.

[28] Kickert, W. J. (1978). *Fuzzy theories on decision-making.* Leiden: Nijhoff Social Sciences Division.

[29] King, B. E., and Benton, W. C. (1988). Master production scheduling, customer services and manufacturing flexibility in an assembly-to-order environment. *International Journal of Production Research*, Vol. 26 No. 6, pp. 1015-1036.

[30] Klafehn, K., Weinroth, J., and Boronico, J. (1996). *Computer simulation in operation management.* Westport: Greenwood Publishing Group, Inc.

[31] Knolmayer, G., Mertens, P., and Zeier, A. (2002). *Supply Chain Management Based on SAP Systems.* Heidelberg: Springer-Verlag.

[32] Kochhar, A. K., Ma, X., and Khan, M. N. (1998). Knowledge-based systems approach to the development of accurate and realistic master production schedules. *Journal of Engineering Manufacture*, Vol. 212 pp.453-60.

[33] Koh, S. L., and Simpson, M. (2005). Change and uncertainty in SME manufacturing environment using ERP. *Journal of Manufacturing Technology Management*, Vol. 16 No.6, pp.629-653.

[34] Koh, S. L., Gunasekaran, A., and Saad, S. M. (2005). Tacking uncertainty in ERP-controlled manucaturing environment: a knowledge management approach. *International Journal Advanced Manufacturing Technology*, DOI 10.1007/s00170-005-0251-2.

[35] Koh, S. L., Simpson, M., and Lin, Y. (2006). Uncertainty and contingency plans in ERP-controlled manufacturing environments. *Journal of Enterprise Information Management*, Vol. 19 No. 6, pp. 625-645.

[36] Kumar, M., Vrat, P., and Shankar, R. (2005). A fuzzy programming approach for vendor selection problem in a supply chain. *International Journal of Production Economics*, 101 pp. 273 - 285.

[37] Lai, Y.-J., and Hwang, C.-L. (1992). *Fuzzy mathematical programming.* Berlin Heidelberg: Springer-Verlag.

[38] Lai, Y.-J., and Hwang, C.-L. (1994). *Fuzzy multiple objective decision making. methods and application.* Berlin Heidelberg: Springer-Verlag.

[39] Lee, Y. Y. (1990). *Fuzzy set theory approach to aggregrate production planning and inventory control.* Dept. of I.E, Kansas State University: Ph.D. Dissertation.

[40] Lin, N.-P., and Krajewski, L. (1992). A model for master production scheduling in uncertain environments. *Decision Science*, Vol. 23 No. 4, pp. 839-861.

[41] Liu, J., and Lampinen, J. A. (2005). A Fuzzy Adaptive Differential Evolution Algorithm. *Soft Computing - A Fusion of Foundations, Methodologies and Applications* , 9, pp.448 - 462 .

[42] MathWorks. (2004). *Matlab Documentation.* MathWorks Inc.

[43] McClelland, M. K. (1987). Order promising and Master Production Schedule. *Decision Science*, Vol. 19 No. 4, pp. 858-879.

[44] Michalewicz, Z. (1996). *Genetic algorithms + data structure = evolution programs.* Berlin: Springer.

[45] Moon, Y. B., and Phatak, D. (2005). Enhancing ERP system's functionality with discrete event simulation. *Industrial Management and Data System*, Vol. 105 No. 9, pp.1206-1224.

[46] Moser, G. (1999). *SAP R/3 Interfacing using BAPIs.* Wiesbaden: 1st edition, Gabler.

[47] Murray, M. (2006). *SAP MM - Functionality and Technical Configuration.* Bonn, Germany: SAP PRESS.

[48] Orlicky, J., and Plossl, G. W. (1994). *Orlicky's Material Requirement Planning.* New York: 2nd Edition, McGraw-Hill.

[49] Plossl, G. W., and Lankford, R. L. (1984). The direction of U.S. manufacturing. *Production and Inventory Management Review*, Vol. 4 No. 10, pp. 74-88.

[50] Plossl, G. W., and Mather, H. F. (1983). *The master production schedule: Management's handle on the business.* Atlanta: Revised edition, George Plossl Educational Services .

[51] Proud, J. F. (1999). *Master Scheduling - 2rd Edition.* John-Wile-Sons Inc.

[52] Ptak, C. A., and Schragenheim, E. (1999). *ERP.* Florida, USA: CRC Press LLC.

[53] Roy, T. K., and Maiti, M. (1998). Multi-objective inventory models of deteriorating items with some constraints in a fuzzy environment. *Computer and Operation Research*, Vol. 25, No.12, pp. 1085-1095.

[54] Sakawa, M. (1993). *Fuzzy sets and interactive multiobjective optimization.* New York: Plenum Press.

[55] SAP. (2001). SAP IDES Documentation. Release 4.6C. Walldorf, Baden Württemberg, Germany.

[56] Sari, F. J. (1997). Master production schedule. In J. H. Greene, *Production & Inventory Control Handbook* (p. chap. 12). 3rd Edition, APICS.

[57] Schaffer, J. D., Caruna, R. A., Eshelman, L. J., and Das, R. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. *Proceeding of the 3rd International Conference on Genetic Algorithms and Their Applications* (pp. 51-60). San Mateo, CA: Morgan Kaufmann Publishers.

[58] SDZ. (2003, February). DOSIMIS-3 for MS-WINDOWS. Dortmund, NRW, Germany: Version 3.2, SDZ GmbH.

[59] Sheikh, K. (2003). *Manufacturing Resource Planning (MRP II).* New York, USA: McGraw-Hill.

[60] Sivanandam, S. N., and Deepa, S. N. (2008). *Introduction to genetic algorithms.* Heidelberg: Springer-Verlag.

[61] Slack, N. S. (2001). *Operation Management.* Prentice Hall – 3rd.

[62] Snabe, J. H., Rosenberg, A., Möller, C., and Scavillo, M. (2009). *Business Process Management - The SAP Roadmap.* Boston: 1st edition, Galileo Press Inc.

[63] Soares, M. M., and Vieira, G. E. (2008). A New multi-objective optimization method for master production scheduling problems based on genetic algorithm. *International Journal of Advanced Manufacturing Technology*, DOI 10.1007/s00170-008-1481-x.

[64] Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control.* John Wiley & Sons, Inc.

[65] Storn, R. M., and Price, K. V. (1997). Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11, pp.341 - 359.

[66] Tallon, W. J. (1988). A comparative analysis of master production scheduling techniques for assemble-to-order products. *Decision Science*, Vol. 20 No. 3, pp. 492-506.

[67] Tiwari, R. N., Dharmahr, S., and Rao, J. R. (1987). Fuzzy goal programming-an additive model. *Fuzzy Sets and Systems*, 24, pp. 27-34.

[68] Tsang, E. P. (1995). Scheduling techniques: a comparative study. *BT technology journal*, Vol. 13, no 1 (127 p.) (44 ref.), pp. 16-28.

[69] Vieira, G. E. (2004). A practical view of the compexity in developing master production schedules: fundementals, examples, and implementation. In J. W. Herrmann, *Handbook of Production Scheduling* (pp. pp. 149-176). Maryland, USA: Springer.

[70] Vieira, G. E., and Favaretto, F. (2006). A new and practical heuristic for master production scheduling creation. *International Journal of Production Research*, Vol. 44 No. 18-19,pp. 3607-3625.

[71] Vieira, G. E., and Ribas, C. P. (2004). A new multi-objective optimization method for master production scheduling problems using simulated annealing. *International Journal of Production Research*, Vol.42, No. 21, 4609-4622.

[72] Vieira, G. E., Favaretto, F., and Ribas, P. C. (2003). Comparing genetic algorithms and simulated annealing in master production scheduling problems. *Proceeding of 17th International Conference on Production Research.* Blacksburg, Virginia, USA.

[73] Vollmann, T. E., Berry, W. L., and Whybark, D. C. (1997). *Manufacturing planning and control system.* New York: 4rd Edition, McGraw-Hill.

[74] White, S. A. (2004, May 3). *http://www.bpmn.org/Documents/BPMN V1-0 May 3 2004.pdf.* Retrieved September Wednesday, 2009, from http://www.bpmn.org.

[75] Wu, Y., Liu, M., and Wu, C. (2002). A genetic algorithm for optimizing the MPS of a processing-assembly production line with identical machines. *Proceedings of the First International Conference on Machine Learning and Cybernetics.* Beijing.

[76] Yusuf, Y., and Little, D. (1998). An empirical investigation of enterprise-wide integration of MRP II. *International Journal of Operation & Production Management*, Vol. 18 No. 1, pp. 66-86.

[77] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, Vol 8, pp. 338-353.

[78] Ziems, D., Neumann, G., Georgiev, M., Tolujev, J., and Sujev, W. (1993). *Materialflußsimulation mit Dosimis-3.* Magdeburg.

[79] Zimmermann, H. J. (1976). Desciption and optimization of fuzzy system. *International Journal of General Systems*, pp. 209-215.

[80] Zimmermann, H. J. (1978). Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems*, 1, pp. 45-55.

[81] Zimmermann, H. J. (1991). *Fuzzy set theory and its applications.* Dordrecht: Kluwer Academic Publisher Group.

[82] Zimmermann, H. J., and Zysno, P. (1985). Quantifying vagueness in decision model. *European Journal of Operational Research*, Vol. 22, pp. 148-158.