2011

CHANGE-READY MPC SYSTEMS AND PROGRESSIVE MODELING: VISION, PRINCIPLES, AND APPLICATIONS

Mohamed Ismail
University of Windsor

CHANGE-READY MPC SYSTEMS AND PROGRESSIVE MODELING: VISION, PRINCIPLES, AND APPLICATIONS

by

Mohamed Abdel-Wahab Mangoud M. Ismail

A Dissertation
Submitted to the Faculty of Graduate Studies
through Industrial and Manufacturing Systems Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada
2011

DECLARATION OF PREVIOUS PUBLICATIONS

This dissertation includes one original paper that has been previously published in a peer-reviewed journal. In addition, a book chapter that was a very early framework of change ready MPC systems is also reported here.

Thesis Chapter	Publication Title/Full Citation	Publication Status
Chapter 4	Ismail, M. A. and H. ElMaraghy (2009). "Progressive modeling-An enabler of dynamic changes in production planning." CIRP Annals - Manufacturing Technology **58**(1): 407-412.	Published
--	Ismail, M. A. and H. A. ElMaraghy (2009). Component Oriented Design of Change-Ready MPC Systems. Changeable and Reconfigurable Manufacturing Systems. H. A. ElMaraghy, Springer-Verlag 213-226.	Published

I certify that the above material describes the work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I do not need to obtain a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my dissertation, including any final revisions, as approved by my dissertation committee and the Graduate Studies office, and that this dissertation has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

The last couple of decades have witnessed a level of fast-paced development of new ideas, products, manufacturing technologies, manufacturing practices, customer expectations, knowledge transition, and civilization movements, as it has never before. In today's manufacturing world, change became an intrinsic characteristic that is addressed everywhere. How to deal with change, how to manage it, how to bind to it, how to steer it, and how to create a value out of it, were the key drivers that brought this research to existence. Change-Ready Manufacturing Planning and Control (CMPC) systems are presented as the first answer. CMPC characteristics, change drivers, and some principles of Component-Based Software Engineering (CBSE) are interwoven to present a blueprint of a new framework and mind-set in the manufacturing planning and control field, CMPC systems.

In order to step further and make the internals of CMPC systems/components change-ready, an enabling modeling approach was needed. Progressive Modeling (PM), a forward-looking multi-disciplinary modeling approach, is developed in order to modernize the modeling process of today's complex industrial problems and create pragmatic solutions for them. It is designed to be pragmatic, highly sophisticated, and revolves around many seminal principles that either innovated or imported from many disciplines: Systems Analysis and Design, Software Engineering, Advanced Optimization Algorisms, Business Concepts, Manufacturing Strategies, Operations Management, and others. Problems are systemized, analyzed, componentized; their logic and their solution approaches are redefined to make them progressive (ready to change, adapt, and develop further). Many innovations have been developed in order to enrich the modeling process and make it a well-assorted toolkit able to address today's tougher, larger, and more complex industrial problems. PM brings so many novel gadgets in its

toolbox: function templates, advanced notation, cascaded mathematical models, mathematical statements, society of decision structures, couplers—just to name a few.

In this research, PM has been applied to three different applications: a couple of variants of Aggregate Production Planning (APP) Problem and the novel Reconfiguration and Operations Planning (ROP) problem. The latest is pioneering in both the Reconfigurable Manufacturing and the Operations Management fields. All the developed models, algorithms, and results reveal that the new analytical and computational power gained by PM development and demonstrate its ability to create a new generation of unmatched large scale and scope system problems and their integrated solutions. PM has the potential to be instrumental toolkit in the development of Reconfigurable Manufacturing Systems. In terms of other potential applications domain, PM is about to spark a new paradigm in addressing large-scale system problems of many engineering and scientific fields in a highly pragmatic way without losing the scientific rigor.

DEDICATION

TO THE MARTYRS OF THE JAN 25TH EGYPTIAN REVOLUTION AND EVERYONE PARTICIPATED: TO THOSE WHO LET US SMELL FREEDOM, TRUST IN JUSTICE, AND HOPE FOR A BETTER TOMORROW.

TO THE SPIRIT OF MY MOTHER (1946-2009)

ACKNOWLEDGEMENTS

First, foremost, and throughout my life, I would like to thank Almighty Allah as all words should fail to express my eternal gratitude for all the blessings that I couldn't count, for energy, and for guidance without which that research couldn't exist.

I would like to thank Dr. Hoda ElMaraghy for her support, guidance, and feedback throughout my PhD program. I would like to express my gratitude to Dr. Firtz Rieger, Dr. Mike Wang, and Dr. Gouqing Zhang for their suggestions, comments, and overall involvement. I would like to thank also Dr. Waguih ElMaraghy for his involvement and suggestions. I am also grateful for Dr. Abdul-Fattah Asfour for his support during my PhD program.

I would like to express my sincere gratitude to Dr. Ashraf Nassef who simply had an instrumental role in many endeavours of my life. Special recognition goes to my preparatory and secondary school teachers Nabil Abdul-Elmohsen, Dr. Abdel Wahed Yossef and Ra'oof.

My utmost gratitude, love, and appreciation go to my father and my mother, may God blesses her soul, for their sacrifices and care throughout my life. Lest to forget all my brothers and sisters who overwhelmed me with their love and unflagging support. I would like to extend my gratitude to my father in law, may God bless his soul too, and my mother in law for their continuous support and best wishes. I would like to express my very sincere appreciation and gratitude to my wife Ola who has shared with me every moment in this tough and hard experience and to my little smiles Judi and Yahya. Many thanks go to all my friends for their help and support throughout my life especially my friend ElSayed Sobhi, may God bless his soul.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MPC	Manufacturing Planning and Control
CBSE	Component Based Software Engineering
CMPC	Change-Ready Manufacturing Planning and Control
PM	Progressive Modeling
APP	Aggregate Production Planning
MMAPP	Multi-Objective Multi-Product Aggregate Production Planning
LDR	Linear Decision Rule
SDR	Search Decision Rule
ROP	Reconfiguration and Operations Planning
EMO	Evolutionary Multi-objective Optimization
SPEA	Strength Pareto Optimization Algorithm
MOGA	Multi-Objective Genetic Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
ERP	Enterprise Resource Planning
SE	Software Engineering
OOAD	Object-Oriented Analysis and Design
LP	Linear Programming
NLP	Non Linear Programming
OR	Operations Research
GP	Goal Programming
IP	Integer Programming
MIP	Mixed Integer Programming
RMS	Reconfigurable Manufacturing Systems
DML	Dedicated Manufacturing Lines
FMS	Flexible Manufacturing System

NOMENCLATURE

T	number of planning periods
C_r	regular time production cost per unit in period t $(\$/\text{unit})$
C_o	overtime production cost per unit in period t $(\$/\text{unit})$
C_s	subcontracting cost per unit in period t $(\$/\text{unit})$
C_h	inventory holding cost perunit in period t $(\$/\text{unit})$
C_b	backorder cost per unit in period $t(\$/\text{unit})$
C_F	firing cost per worker in period t $(\$/\text{worker})$
C_H	hiring cost per worker in period t $(\$/\text{worker})$
R_t	regular time production volume in period t (units)
O_t	overtime production volume in period t (units)
S_t	subcontracted volume in period t (units)
$S_{\max}$	maximum subcontracted volume in period (units)
$W_{\max}$	maximum allowed workforce level (worker)
W_t	work force level in period t (man-day)
Δw	preferedIncrementalworkforcechange
W_0	initial workforce level (worker)
$I_{\max}$	maximum inventory level in period t (units)
B_{max}	maximum backorder level in period t (units)
I_o	initialinventory (units)
B_o	intialbackorders(units)
δ	worker's productivity (unit/hr)
β	number of regular hours per worker in a planning period (hrs)
γ	number of overtime hours per worker in a planning period (hrs)

P_t total product supply in period t (units)

W_t workforce level in period t (workers)

N number of products, product families, or product groups

D_{it} demand of product i in period t (units)

R_{it} regular time production volume of product i in period t (units)

O_{it} overtime production volume of product i in period t (units)

S_{it} subcontracted volume of product i in period t (units)

I_{it} inventory level of product i at the end of period t (units)

B_{it} backorder level of product i at the end of period t (units)

H_t workers hired in period t (man-day)

F_t workers fired in period t (man-day)

W_{it} work force proportion allocated to product i in period t (man-day)

t_{li} labor time to produce one unit of product i (*hours*)

$t_{\sigma i}$ setup time of product i (*hours*)

C_{mi} *material* cost per unit of product i (\$ / uint)

C_{hi} holding cost per unit of product i (\$/uint)

$\mathrm{C}_{\sigma i}$ set-up cost for product i in period t (\$/uint)

C_{si} subcontracting cost of product i (\$/uint)

C_{bi} backorder cost of product i (\$ / unit)

p_i price per unit of product i (\$/unit)

ψ_{i} Set-up decision variable of product i

$W_{t_{\max}}$ maximum work force available in period t (man-day)

ROP Tuplized Nomenclature[1]

Numbers and IDS

p	Product ID, P101, P102, P103 etc.
m	Module ID, M1000, 2000, 3000 etc.
c_k^m	Configuration k loaded to module m ID, C1101,C2101, C3101 etc.
N_b	Number of planning buckets

Product Demand: Mix and Volumes

$D_p[t]$	Product p demand during a planning bucket t
N_p	Number of demanded products
P	a set of all product IDs that a manufacturing firm can make or supply to its markets i.e. product mix, $p \in P$
$\Pr_p$	Price of product p

Product Supply: Parameters

PS_p	Product Supply Tuple
PS_p^{C-mtr}	Material cost of product p (\$/unit)
PS_p^{C-hld}	Holding cost of product p (\$/unit)
$PS_p^{C-sbcntrc}$	Subcontracting cost of product p (\$/unit)

[1] The Tuplized Nomenclature is one of the advancements brought by Progressive Modeling to define the novel "Mathematical Statements." The interested reader is advised to read chapter 7 first.

$PS_p^{C-bkord}$ Backorder cost of product p (\$/unit)

Product Make Parameters

$PM_{p[c_k^m]}$ Configuration dependent products make tuple. It encompasses all product manufacturing dependent data.

- $PM_{p[c_k^m]}^{t-srp}$ Set up/Ramp-up time of product p loaded to configuration c_k^m
- $PM_{p[c_k^m]}^{t-unld}$ Unloading time of product p unloaded from configuration c_k^m
- $PM_{p[c_k^m]}^{C-srp}$ Setup cost of product p loaded to configuration c_k^m
- $PM_{p[c_k^m]}^{C-unld}$ Unloading cost of product p unloaded from configuration c_k^m
- $PM_{p[c_k^m]}^{t-cycle}$ Cycle time of product p loaded to configuration c_k^m
- $PM_{p[c_k^m]}^{thrpt}$ Throughput of product p loaded to configuration c_k^m
- $PM_{p[c_k^m]}^{VC}$ Variable cost of product p loaded to configuration c_k^m (Fixed costs are already included in set/ramp ups and unloading costs)

Product Make Plan Variables

$PM_p[t]$ Product Make Tuple

- $P_{[c_k^m]}^{()}[t]$ Product ordered set that should be made by configuration c_k^m during time bucket t
- $\beta_{p[c_k^m]}^{t-srp}[t]$ Product p setup/ramp up binary variable: equals 1 when a product is loaded to a configuration c_k^m during time bucket t
- $\beta_{p[c_k^m]}^{unld}[t]$ Product p unloading binary variable: equals 1 when a product is unloaded from a configuration c_k^m during time bucket t
- $\beta_{p[c_k^m]}^{prd}[t]$ Product manufacturing binary variable: equals 1 when a product is

manufactured by configuration c_k^m during time bucket *t*

$PMB_{p\left[c_k^m\right]}^{t-srp}[t]$	Set/ramp up time of product p made by configuration c_k^m during time bucket *t*
$PMB_{p\left[c_k^m\right]}^{t-unld}[t]$	Unloading time of product p if it has been made by configuration c_k^m during time bucket *t*
$PMB_{p\left[c_k^m\right]}^{R}[t]$	Regular quantity of product p made by configuration c_k^m during time bucket *t*
$PMB_{p\left[c_k^m\right]}^{O}[t]$	Overtime quantity of product p made by configuration c_k^m during time bucket *t*
$PMB_{p\left[c_k^m\right]}^{O\max}[t]$	Maximum overtime quantity of product p made by configuration c_k^m during time bucket *t*
$PMB_{p\left[c_k^m\right]}^{t-R}[t]$	Regular time allocated to produce product *p* during a planning bucket *t* when it is loaded to configuration c_k^m
$PMB_{p\left[c_k^m\right]}^{t-O}[t]$	Over time allocated to produce product *p* during a planning bucket *t* when it is loaded to configuration c_k^m

Product Supply: Mix and Volumes

$PSB_p[t]$	Product supply bucket: a tuple of product supply mix and volumes during a certain bucket *t*
$PSB_p^R[t]$	Total regular time supply (volume) of product p during time bucket *t*
$PSB_p^O[t]$	Total overtime supply (volume) of product *p* during time bucket *t*
$PSB_p^S[t]$	Total subcontracted quantity (if subcontracting is available) of product *p* during time bucket *t*
$PSB_p^I[t]$	Total Inventory quantity of product *p* during time bucket *t*

$PSB_p^B[t]$	Total backordering quantity of product p during time bucket t
$PSB_p^I[0]$	Initial Inventory quantity of product p prior to the current planning session
$PSB_p^B[0]$	Initial backordering quantity of product p prior to the current planning session

Module Workforce Plan

$W_m[t]$	Module Workforce bucket: a tuple of (Workforce level and their dependent hiring and firing values)
$W_m[t]$	Workforce level at module m during period t
$H_m[t]$	Workers hired at module m during period t
$F_m[t]$	Workforce fired at module m during period t

Configuration Parameters

$G_{\left[c_k^m\right]}$	Configuration tuple that represents all the data related to a certain configuration k loaded to a certain module m
$G_{\left[c_k^m\right]}^{W}$	Work force level attached to configuration c_k^m
$G_{\left[c_q^m\right]\left[c_k^m\right]}^{t-Rcn}$	Reconfiguration time from configuration q to configuration k
$G_{\left[c_q^m\right]\left[c_k^m\right]}^{VC-Rcn}$	Reconfiguration variable cost from configuration q to configuration k
$G_{\left[c_q^m\right]\left[c_k^m\right]}^{FC-Rcn}$	Reconfiguration fixed cost from configuration q to configuration k
$P_{\left[c_q^m\right]}$	Configuration product set that can be made while configuration c_k^m is loaded

Module related Data

W_m	Module work force tuple
C_F^m	Work force firing cost of a worker has a skill-set standard needed by module *m*
C_H^m	Work force Hiring cost of a worker has a skill-set standard needed by module *m*
C_W^m	Regular work force hourly rate ($/hr)
C_o^m	Overtime hourly rate ($/hr)
CP_m	Configuration path of module m: N_b-tuple of configurations c_k^m indexed by bucket order $(1,2,3,\ldots,N_b)$
$\beta_{CP^m}^{rcnfg}[t]$	Reconfiguration binary variable of configuration $CP_m[t]$
$\tau_{CP^m}^{rcnfg}[t]$	Reconfiguration time consumed to load configuration $CP_m[t]$ during time bucket *t*
$C_{CP^m}^{rcnfg}[t]$	Reconfiguration cost of module m during time bucket *t*
$C_{CP^m}^{srp}$	Setup and ramp costs of configuration path *m*
$C_{CP^m}^{unld}$	Unloading costs of configuration path *m*
$C_{CP^m}^{prd}$	Operation costs of configuration path *m*
$C_{CP^m}^{rcnfg}[t]$	Reconfiguration cost of module *m* during time bucket *t*
CMap	Configuration Map: Nm-tuple of configuration paths (CP_m s)

System Work Regulations

SWR	System Working Regulation Tuple (working days/month, hrs/shift,

shift/day etc.)

$SWR^{WD}[t]$	Working Nb-Tuple: a sequence of period-work days, indexed by bucket number or order (1,2,..)
$SWR^{WH}[t]$	Working Nb-Tuple: a sequence of period-work days, indexed by bucket number or order (1,2,..)
$SWR^{OT}[t]$	Maximum overtime hours allowed per day: Nb-tuple indexed by bucket number or order
$SWR^{h/s}$	Number of working hours per shift
$SWR^{s/d}$	Number shifts per day

System Initial and Final States

B_{p0}	Outstanding back orders of product p at start of planning horizon (units)
I_0	Inventory of product p at start of planning horizon (units)
W_0	Workforce at the at the start of the planning horizon (man-day)
A_{kp}	Pre-planning system state constants, *k=1,2,…* of products *p*
$\mathfrak{M}_{kp}$	End-of-Planning System desired constants, *k=1,2,..* state of product *p*
$_{[^m C_k]}[0]$	Initial Configuration k loaded to module *m* (i.e. bucket 0 configuration)
$_{[^m C_k]p}[0]$	last product *p* loaded to configuration *k* loaded to module *m* (i.e. bucket 0 last product)

Chapter 1 INTRODUCTION

1.1 Introduction

1.1.1 Manufacturing Paradigms

Over the years, the manufacturing environment has been changing relentlessly with market conditions and customer requirements. Manufacturers have kept inventing, developing, and modernizing products, manufacturing process, manufacturing technology, and business process. They have been striving very hard to promote their competitive edge and trying to hit excellence in every aspect in order to prosper and sometimes in order to just survive. Early manufacturing paradigms were both very primitive in technology and very lethargic to market needs. Multitudes of one-of-a-kind products were available and the craftsmanship was the key enabler of those old paradigms. At the beginning of the twentieth century, mass markets started to triumph and the economics of scale was the chief driving force. Productivity and large volumes were needed to feed the starving mass markets. Product cost was the main customer key driver. Stocking huge amounts of products at manufacturing facilities and wholesalers was the key solution to stay responsive. Dedicated production lines were the manufacturing technology at that time. This manufacturing paradigm was referred to as the mass production era. After the World War II, The manufacturing process was re-innovated by the advent of Just in Time Manufacturing (JIT). The philosophy behind is the elimination of waste. JIT, founded at Toyota manufacturing plants, represents one of the most famous contributions of the Japanese manufacturers to the industrial world. The goal is to make equipment, resources, and labour available in the right amounts and at the right time. Several enablers are needed a priori to establish a successful JIT system such as integrating and optimizing every step of manufacturing process, producing a quality product, reducing manufacturing costs, producing on demand, developing manufacturing flexibility, and keeping commitments and links between customers and suppliers. Failing to have any of these enablers undermines severely JIT success (Hutchins 1999). In the late eighties, the term "Lean Manufacturing" was coined. According to Roose (cited in Groover (2000)) there are four principles underlie lean production: 1) minimize waste 2) perfect first time quality 3) flexible production lines 4) continuous improvement. Extended on the lean manufacturing principles, agile manufacturing was founded in the early nineties. Agility can be simply defined as the ability of a firm to thrive in a competitive environment characterized by continuous change and, sometimes, unanticipated change. Similar to lean manufacturing, agile manufacturing is based on four principles: 1) organize to master change 2) leverage the impact of people and information 3) cooperate to enhance competitiveness 4) enrich the customer (Groover 2000).

Manufacturing paradigms define the overall direction of manufacturing enterprises and how they formulate their core competencies. Manufacturing technology and manufacturing planning and control systems play critical roles in shifting the direction of manufacturing firms towards one or more of these paradigms. Change-Ready Manufacturing Planning and Control (CMPC) systems and Progressive Modeling (PM) presented by this research embrace the best practices of these paradigms and capitalize on mixing and matching some of them in order to define many distinguished competitive formula that should bring many manufacturing activities to an optimized tandem.

1.1.2 Manufacturing Technology

Manufacturing technology has developed over the years from general-purpose machines and equipment to more specific ones with built-in or pre-set characteristics. Product volume/variety spectrum plays a key role in describing the appropriate technology. General-purpose machines are used in job-shop manufacturing where products are of high variety and very low volumes. Driven by economics of scale, dedicated manufacturing lines serve the other extreme where products are produced in massive volumes and very low variety. In order to address the mid-volume and mid-variety production zones, Flexible Manufacturing Systems (FMS) were developed. Flexible Manufacturing Systems (FMSs) are designed with anticipated product variations and built-in flexibility a priori to achieve what is known as economics of scope (ElMaraghy 2005). FMS suffer from being capital intensive and sometimes possesses underutilized flexibility. In an initiative to overcome these shortcomings and to introduce a better agile manufacturing technology, the Reconfigurable Manufacturing Systems concept was introduced in the late nineties (Koren, Heisel et al. 1999; Mehrabi, Ulsoy et al. 2000; Koren 2003). Instead of built-in flexibility provided by FMS, RMS promises a customized flexibility on demand. RMS aims to achieve either convertible or scalable capacity or both. In an RMS, machines, machine modules, equipment, etc. can be added, interchanged, upgraded, and removed as needed and when needed. Proponents of RMS believe that the emerging technology can offer a cheaper solution, at least in the long run, compared to FMSs as it can increase the life and utility of manufacturing systems (ElMaraghy 2005). In order to be readily reconfigurable, manufacturing systems must possess certain key characteristics: i) Modularity of component design, ii) Integrability for both ready integration and future introduction of new technology, iii) Convertibility to allow quick changeovers between products and quick system adaptability for future products, iv) Diagnosability to identify quickly the sources of quality and reliability problems, v) Customization to match designed system capability and flexibility to applications, and vi) Scalability to incrementally change capacity rapidly and economically.

Reconfigurable Manufacturing Systems (RMS) and their intrinsic nature of being in a continuous state of change was the first catalyst that spurred the CMPC systems project. The early objective was to develop an MPC system that is able to catch the pace of the underlying changeable manufacturing system and the manufacturing process. RMS, which suffer from the lack of existence in a full-fledged format, make thinking in terms of modeling RMS operations management and other problems a tough task. There is a need to develop a new modeling approach that is able to build flexible yet robust models that can master the activities of the next generation manufacturing technology that made change propagates to almost everything surrounding: system, product, process, workforce, and even the value offered in terms of competitive products and market postures.

1.1.3 Manufacturing Planning and Control Systems

Whatever the manufacturing paradigm embraced and the manufacturing technology utilized, the eternal challenges for all manufactures stay the same: manufacturers need to control the types and quantities of the materials they purchase. They need to plan which products to produce and in what quantities. They need to ensure that they are able to meet current and future customer expectations whether they are low cost, well differentiated, or customized products. Making inappropriate decisions under certain circumstances or related to a certain strategic area not only make the company lose money but also undermine its core competency. The competency built in these systems represents a corner stone in defining excellence and competitiveness in any modern manufacturing enterprise. Change-Ready Manufacturing Planning and Control (CMPC) systems aim to accomplish manufacturing practice excellence by capitalizing on orchestrating system levers, embracing a balanced set of best industrial practices, and developing highly esoteric models using the novel Progressive Modeling approach.

Figure 1-1 shows the widely accepted model of MPC system presented by Berry and Hall (1992) and reported in Vollman et al (2005). The model divides the MPC activities into three stages that are differentiated by their hierarchy: front end, engine, and back end or time frames: long, medium, and short terms. The front end establishes the overall company direction. Demand management coordinates all the activities of the business and lay some restrictions and requirements on system resources. Sales and operations planning balance the marketing plans with the available production resources. The Master Production Schedule (MPS) is the disaggregated version of sales and operations plan. Resource planning determines the capacity necessary to produce the required products now and in the future. It provides the basis necessary for matching manufacturing plans and capacity. The engine encompasses detailed material and capacity planning. The Material Requirement Planning (MRP) explodes the period-by-period plans for all component parts and raw materials required to produce all the products in the MPS. This material plan is utilized in the detailed capacity planning systems to compute labour or machine capacity required to manufacture parts. The back end depicts the MPC execution system. The system configuration depends on the products manufactured and production process employed.

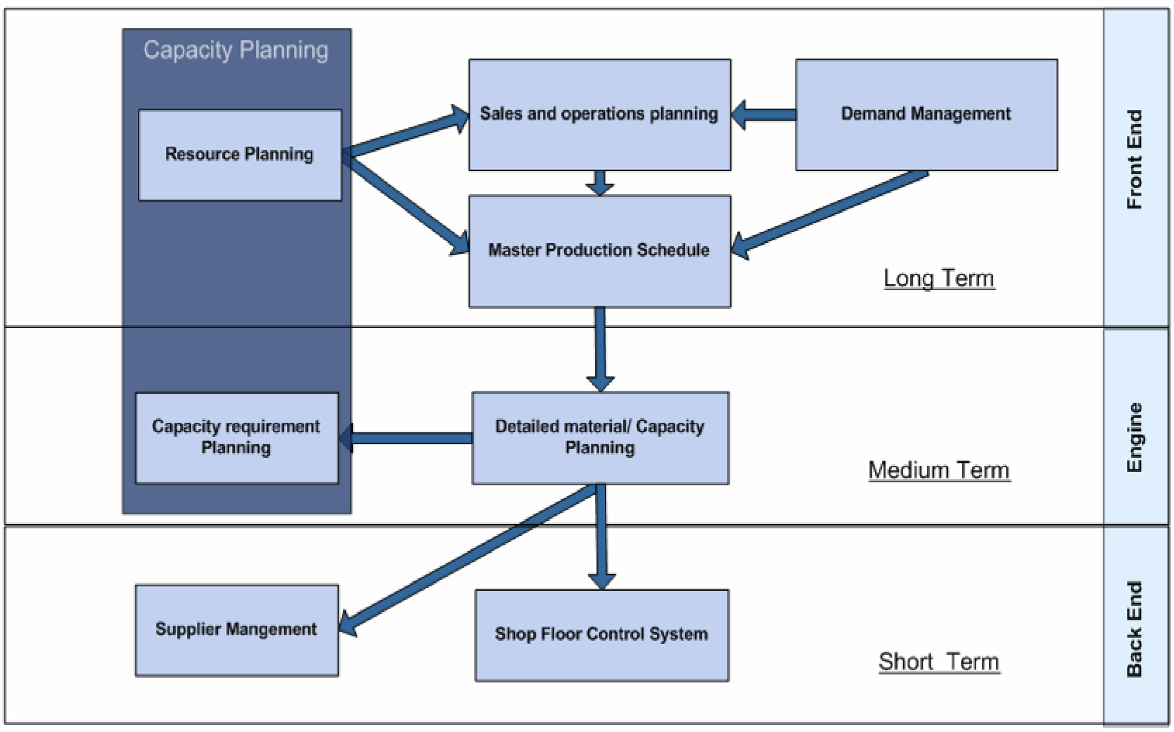


Figure 1-1: Manufacturing planning and control (Vollman, Berry et al. 2005)

An important activity that is not shown in the previous figure is the measurement, follow-up, and control of actual results. If the actual results differ from original plans, appropriate actions must be made to bring results back to plan. These measurements and control are part of all three phases of MPC system. MPC systems as an area of research are beyond the scope of any individual work. This research focuses on just presenting the new mindset of change-ready manufacturing planning and control systems and the new modeling approach, Progressive Modeling. All the work presented by this research can be applied to the MPC frameworks as a whole and many of its individual components or problems. The area of operations planning and control is chosen as a source of applications to test the new foundations and innovations developed by both CMPC systems and PM. The classical Aggregate Production Planning problem will be presented in two variants. In addition, the Reconfiguration and Operations Planning problem that is defined for the first in the RMS literature will be presented.

1.1.4 MPC in Practice: From Reorder Point Systems to Enterprise Resource Planning (ERP)

In their early days, MPC systems consisted of a group of plant foremen who were responsible for scheduling production, ordering materials, and shipping products to their surrounding markets (Rondeau and Litteral 2001). The simplicity of manufacturing process allowed relatively low-skilled workers to manage the whole process.

As the manufacturing technology evolved towards a highly specialized one, reorder point system of production and inventory control gradually prevailed and replaced the older foremen-based systems. Early reorder point systems were manual but they turned to be automated with the advent of commercial mainframe computers in the late 1950s and early 1960s. Later in the mid-1960s, material requirement planning systems started to evolve and replace reorder point systems. MRP systems offered a forward-looking demand based approach for planning the manufacturing of products and the ordering of inventory. They overcame the high variability of inventory levels experienced by reorder point systems through smoothness and effective management. They provided, also, a basic set of computerized production reporting tools used to evaluate the viability of master production schedule against projected materials demand. In the mid of 1970s, Manufacturing Resource Planning (MRPII) started to replace gradually MRP systems as a manufacturing control system of choice. MRPII added the capacity requirement planning (CRP) capability to MRP to create an integrated or closed loop MPC system (Sock Hwa and Snyder 2000). The overall production capabilities were calculated accurately for the first time taking into consideration both materials and capacity requirements constraints. Utilizing the new shop floor control production capabilities production scheduling and monitoring the execution of production plans were much easier. By the 1990s and with the increasing level of global competition, changing markets, and developing technologies, manufacturers all over the world were forced to reinvent their products, services, their organizational structure, and operational control (Sock Hwa and Snyder 2000). ERP system enabled these firms to meet the global directive of continuous improvement of the supply chain process through flexible, customer-driven information management. Most of already existing ERP systems still uses the basic model of MRPII systems and encompasses human resources, decision support applications, and some other specialized configurations. ERP packages encapsulate best business practices (Hiebeler, Kelly et al. 1998) which can guide a manufacturing firm from early stages of product engineering to the last stages of product implementations. ERP adoption takes from few months for firms accepting all settings to sometimes several years for firms need major modifications. Although most ERP systems have many business practice processes embedded in their repositories, not all of them are necessarily best in a certain class of applications or for a specific firm.

ERP systems and their MPC components suffer from being very generic solutions and count a lot on heuristics rather than pragmatic models in executing manufacturing planning and control activities. Industry reports many failure stories about large-scale business solutions implementations in many enterprises. When the Aggregate Production Planning problem was chosen to be the application of the new foundations presented by CMPC and PM, unfortunately, many models of the literature studied failed to have a real application. That was identified as an academic-industrial gap that was addressed by Progressive Modeling. It is now part of PM mission is to create logical models that work and could be implemented in the industry. PM, compared to ready-made packages, allows manufacturer to embed their core competency in their models, which should be a novel competitive edge by itself.

1.2 Motivation

As already reported earlier, developing an MPC system for Reconfigurable Manufacturing Systems was the first catalyst that spurred this research. The new technology counts a lot on building a new modular manufacturing system that should be highly responsive to its market changes. RMS posed a new challenge for maintaining changing and evolving environment over time. In this new environment, the system, associated manufacturing processes, products, workforce could change in order to meet market demand. The Change-Ready manufacturing planning and control systems are proposed to define the new MPC systems that could serve such changeable environment. Change drivers, CMPC characteristics, and Component Based Software Engineering (CBSE) are presented. Now that a new framework of coarse-grained components has been presented, the next step is to select one of its components, study its internals, and show how the ingrained logic can be made change ready. Progressive Modeling, a novel multidisciplinary forward-looking modeling approach, is presented so order to address the MPC problems in changeable environments. Another objective of PM is to create the logic that lessens the gap between the idealistics of some MPC academic literature and the pragmatics of the industrial world. The last orientation was catalyzed by the lack of applicability of many models presented in the MPC literature.

1.3 Research Projects

Throughout this research, three major projects are conducted. The second and the third are interwoven. Except for PM process and some early innovations related to it, most of the advancements came into the way while working on the applications under study and maintaining the RMS challenges and PM vision in mind. Every new idea, gadget, or piece of logic presented is developed to be both generic and applicable to many problems not only those developed in this research.

1. Change Ready MPC Systems: a new vision of how to make manufacturing planning and control systems ready for change that may be created via development or mitigated as a threat originated either from within or from the surrounding environment.

2. Progressive Modeling: PM is an innovative multidisciplinary modeling approach that has been developed to better model industrial problems in a practical and modern way without losing the scientific rigor.

3. MPC Applications: in order to illustrate the principles presented by PM and to illustrate its new potential three applications are presented:

I. **Aggregate Production Planning problem (APP):** PM brings a new system perspective to many industrial problems; the APP problem is redefined from that perspective. A new mathematical model that represents the new PM concept of PM function templates is presented. The objectives are tied to best manufacturing practices.

II. **Multi-objective Multi-Product Aggregate Production Planning Problem with Setup Decisions (MMAPP):** MMAPP is supposed to be a tougher problem than its ancestor APP. While demonstrating MMAPP, system envelop constraints, constraint satisfaction algorithm, couplers, and turning product plans into state machines are presented as some novel PM gadgets. In addition, a new MMAPP formulation, mathematical model, and novel solution algorithm are also demonstrated.

III. **Reconfiguration and Operations Planning Problem (ROP):** The Reconfiguration and operation planning problem define many related principles and foundations of operations management in an RMS environment. The ROP problem is defined for the first time in the RMS literature. Advanced nomenclature, mathematical statements, and structured search space are some PM large-scale problem modeling gadgets presented to serve ROP modeling and solution algorithm.

1.4 Research Approach

1.4.1 Change-Ready MPC Systems

A new set of system characteristics are proposed to envision how a change-ready MPC system should behave and interact to its environment. The suggested system consists of a different set of loosely coupled interacting components. Change Drivers are identified to guide some foundations upon which CMPC system model can be identified. Based on the advances of Component-Based Software Engineering (CBSE), and Object-Oriented Analysis and Design (OOAD), a conceptual framework of Change-Ready Manufacturing Planning and Control (CMPC) system is presented. Some components of MPC system functions are discussed form CMPC perspective.

1.4.2 Progressive Modeling I and APP

Progressive Modeling started as a process to formalize the problem analysis, modeling, and solution in a much modernized and synergistic way. To date, PM has passed by three main phases of development. In the first phase, the process is presented and a progressive mathematical model of the APP problem is developed; in addition, a new solution approach is presented. Progressive models are ready to change, adapt, and develop further. The aggregate production problem itself works as an illustrating application. In this phase, innovations are limited to the analytics and math models.

1.4.3 Progressive Modeling II and MMAPP

In this phase, a better-revised and more generic version of PM process is presented. System envelop constraints are introduced for the first time and many innovations related to the solution algorithms are presented. The Multi-Objective Multi-Product aggregate production planning with setup decisions is utilized as a case problem.

1.4.4 Reconfiguration and Operations Planning Problem

The Reconfiguration and Operation Planning problem is presented as an RMS-application of CMPC and PM principles in an RMS environment. The problem definition, scope, size, and implications are unprecedented in the RMS literature. A great body of this dissertation is dedicated for this problem alone. Advanced notations, cascaded mathematical models, mathematical statements, structured search space, and society of decision structures are some innovations brought by PM to address such large-scale problems.

1.5 Dissertation Outline

The dissertation that documents this research is divided into three major parts. In part 1, introduction and literature review are discussed. This part shows the research framework and gaps to be addressed. In part 2, the main concepts developed, Change-Ready MPC systems and Progressive Modeling are introduced. In part 3, the focus is shifted towards the Reconfiguration and Operations Planning (ROP) problem of reconfigurable manufacturing systems. The following is an outline of the dissertation chapters:

Chapter 1 introduces the whole dissertation and describes the main motivation behind this research, objectives, research projects, and dissertation outline.

Chapter 2 discusses the most related literature that serves the common purpose of this research. MPC frameworks, Object Oriented MPC systems, Reconfigurable Manufacturing Systems, Aggregate Production Planning problem, and Evolutionary Multi-Objective Optimization (EMO) are considered some areas of research that are directly related to this research. Additional and more specific literature may be added wherever necessary as an integrated part of the remaining chapters.

Chapter 3 discusses and introduces the Change-Ready MPC systems, Change drivers, CMPC characteristics, CMPC frameworks, and some samples of how CMPC components should interact and behave in a change ready fashion.

Chapter 4 introduces Progressive Modeling and its first governing philosophy "Propagation of balance." The new modeling approach is illustrated using the Aggregate Production Planning as a case problem. This chapter ends by showing the implications of the new methodology as dynamic, flexible, forward-looking modeling approach.

Chapter 5 introduces the academic-industrial gap from MPC perspective. An updated version of PM process is presented. The focus is shifted towards modernizing solution algorithms to make them progressive. The Multi-Objective Multi-Product Aggregate Production Planning (MAAPP) with set up decisions is the case problem of this chapter.

Chapter 6 introduces the reconfiguration and operations planning problem. The new manufacturing amorphous process presentenced by RMS is introduced first; a data model is developed to serve the ROP problem definition. This chapter lays the foundations needed by the next three chapters. This chapter coins a new problem definition in the RMS literature, the Reconfiguration and Operations Planning Problem.

Chapter 7 presents the mathematical statement of the ROP problem. The ROP problem unleashes the analytical, logical, and computational power brought by PM. Advanced notation, deployed nomenclature, hierarchical binaries, and cascaded mathematical models are some innovations that increased the capabilities of PM to define a problem like the ROP. The chapter concludes by consolidating all the mathematical models presented in one mathematical statement.

Chapter 8 presents the solution approach of the ROP problem. A society of decision structures is presented first to define the ROP entities defined in the search space. Configuration maps represent a condensed capsule of many operations decisions in an RMS environment. Accordingly, dependent/semi-independent decisions are identified: product supply plans, inventory/backorders, subcontracting plans if applicable.

Couplers, a PM concept presented earlier in chapter 5, are used to define independent decisions. Many operators are presented to tweak the search space in order to find better alternatives. This chapter ends by the master algorithm that wires everything together and manages the system optimization process.

Chapter 9 presents a case study of the ROP problem to test its logic and principles developed in the earlier three chapters. The case study shows to what extent PM could be an enabler in embodying an almost real reconfigurable environment that can be analyzed and developed. The results demonstrate that all the planning activates can be done in tandem. The ROP defines a new potential for PM in developing a pragmatic logic that governs systems not just problems.

Chapter 10 summarizes the dissertation, illuminates major contributions, and sets the direction for PM development.

Chapter 2 LITERATURE REVIEW

2.1 Introduction

In this chapter, a review of the literature that generally related to this research is presented. MPC frameworks and Object Oriented MPC systems have direct relations to software aspects and technologies recommended. Since the aggregate production-planning problem is the case problem utilized to demonstrate many principles brought by Progressive Modeling (PM), the literature related is reviewed in brief and some shortcomings are highlighted. All the PM applications presented in this research have an embedded Evolutionary Multi-objective Optimization (EMO) algorithm as a part of their solution algorithms. Some popular EMO algorithms are also presented. Remarks and comments conclude every section (sub-section) to clarify some directions developed by this research.

2.2 Integrated MPC frameworks

Since the late eighties, the development of MPC frameworks and architectures has attracted the attention of many researchers. Monfared and Yang (2007) affirmed that the global competition and the need for improved responsiveness, particularly in low-volume, high-variety manufacturing industries, necessitate further integration and automation in planning, scheduling and control functions. They argued that in order to achieve automation, some concepts and techniques from operations research, control theory, and computer science should be integrated, enriched, and unified to provide a platform for automation. They proposed a new framework for the automation and integration of planning, scheduling, and control functions. A fully automated flow shop production system was presented to illustrate the applicability of the new framework.

Wu (2000; 2001) argued that a conceptual manufacturing framework is essential in order to develop a manufacturing science. He proposed a framework called Manufacturing System Management (MSM) that consists of three main modules: Manufacturing Strategy Analysis (MSA), Manufacturing System Design (MSD), and Manufacturing Operations Management (MOM). The top-level functional MSA and its implications on MSD and MOM are highly emphasized.

In the context of process industry, which is assumed to be less complex than other discrete manufacturing environments, Shobrys and White (2000) recommended that the current MPC functions should work together in an automated and integrated fashion. Furthermore, they affirmed that all MPC functions could gain a better support provided by advances in data capturing and conditioning, sophisticated analytical techniques, and high-performance computing environments. Nevertheless, they confirmed that maintaining consistency among the decisions continues to be difficult with real economic consequences despite the high-speed communications that can transfer information and data almost without limits.

Artiba and Aghezzaf (1997) developed an architecture of a multi-model-based system for production planning and scheduling. The developed system integrates expert systems, discrete event simulation, optimization algorithms, and heuristics to support decision-making for complex production planning and scheduling problems. Once the aggregate plan has been produced, the scheduling level is then tackled. Some of their multi-model functionalities are employed using different models (MILP, heuristics, rules etc.). The object-oriented approach is used for data modelling, and the loop is repeated until the final results are satisfactory or a fixed number of iterations are reached. The tools used are C++, SLAMII, and Microsoft Excel.

Devedzic and Radovic (1999) developed a framework for building Intelligent Manufacturing Systems (IMSs). The framework developed is composed of software components and uses different advanced techniques such as expert systems, fuzzy logic, and neural networks. Depending on the application, the number, the kind, and the complexity of the intelligent components of an IMS framework can vary widely from one system to another, and the components themselves can be combined in many ways.

Shan et al (2001) introduced an integrated approach for manufacturing systems design which is able to develop and test different alternative using an integrated system called Simulation-Based Decision Support System (SBDSS). SBDSS mainly consists of two subsystems: object library modeller and simulation engine with its manager. Using SBDSS, decision makers can evaluate alternatives in manufacturing and production such as an annual production plan under certain circumstances through scenario simulations. The flexibility of the system was illustrated using application cases.

From the literature considered above, it is clear that the design of an integrated MPC framework is still in its early stages of development. Using the state of the art of software development and AI tools are really appreciated in this regard. However, a deeper and shifted focus towards the modeling level and a linkage with the best business practices are desperately needed. A formalized methodology to build these frameworks is also missing. Change Ready MPC systems and Progressive Modeling presented by this research address these issues. The simple block diagram approach used to define those MPC frameworks will be replaced using Component Based Software Engineering (CBSE). Utilizing CBSE will have many implications on frameworks evolvability, robustness, and efficiency.

2.3 Object Oriented MPC Models

The Object Oriented Analysis and Design (OOAD) principles are commonly used as an approach to master the complexity of building MPC frameworks. In OOAD, an object-oriented system is composed of a group objects that collaborate together in order to define the required system behaviour. Metaxiotis et al (2001) presented an adaptable object-oriented model for production planning and management system. The software developed is a decision support tool that incorporates dynamically the operational requirements, characteristics, and constraints for all particular production shop floor activities on which it may be applied. They argued that this could be achieved through both the adaptability of the modelling approach and the modularity and the manageability of the developed platforms. Nevertheless, the suggested model suffers from being a data oriented model and cannot capture the behaviour and the characteristics of the underlying production system.

Zhang et al (1999) discussed the object-oriented modeling for cell control systems. They defined a manufacturing entity object (MEO), Figure 2-1, as a reusable building block of an agile manufacturing cell (AMC). Each MEO object is composed of two parts: a shell for interfacing with other manufacturing entity objects and a core for executing the required processes. Conceptually, MEO can be broken down into information manufacturing entity object (IMEO) and a control manufacturing entity object (CMEO). The distinctive features of the proposed MEO are reusability, shell-core structure, and directivity. Design and implementation of manufacturing entity objects are guided by the structure and the behaviour of real manufacturing entities.

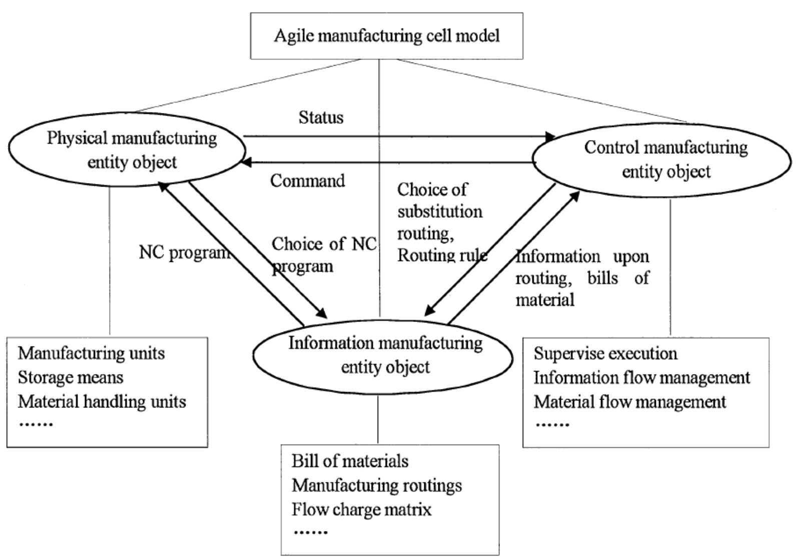


Figure 2-1: Manufacturing entity objects of an agile manufacturing cell (Zhang and Zhang 1999)

Wache (1998) applied Object-oriented Modeling (OOM) in planning and implementation of flexible automated material flow systems. OOM is decomposed into three main components: object-oriented analysis, object-oriented design, and object-oriented programming. The OOM is endorsed as an approach that provides an effective way to master complexity and minimize modeling errors.

Brandimarte et al (2000) devised a high-quality general purpose scheduler which is able to cope with the technological peculiarities of different production environments. A detailed schedule could be prone to disruptions due to the uncertainty affecting the shop floor. Both a modular approach to devise and assemble local schedulers and a way to link predictive and real time scheduling were introduced. In order to cope with both requirements, a scheduling architecture inspired by the well-known shifting bottleneck method was proposed. The modularity of the architecture is illustrated through an object-oriented conceptual model based on the Unified Modeling Language (UML). The resulting architecture is, in some sense, a generalization of the MRP order scheduling mechanism. Unlike MRP, the architecture explicitly deals with capacity constraints and it is not strictly hierarchical.

Chang et al (1990) described an object-oriented system for manufacturing planning and control of a job-shop. Job-shop entities such as cells, machines, jobs, parts, and schedules are modeled as objects. The architecture of the system encapsulates knowledge-based systems and uses computer simulation in its implementation. The developed system considers only the allocation and scheduling of jobs to machines. Tai and Boucher (2002) introduced an architecture for scheduling and controlling a manufacturing system using distributed objects. A cell object that encapsulates data and methods for scheduling and controlling the cell resources is introduced. New jobs entering the manufacturing system are allocated to cells based on schedules computed in real time by these distributed cell objects.

Tsai and Sato (2004) suggested an Agile Production Planning and Control System (APPCS) using UML. In that model, Parts, Bill of Materials (BOMs), Operations, Work-centers, Resources, Shifts, Demand, and Supply are defined as classes with associations among them. Job and Link classes that capture the hierarchical structure of jobs are also presented. The uncertainty caused by customers who might make a change in their orders or by suppliers who might change their promised items was addressed. The proposed system is verified via instantiation and simulation.

Liao et al (2001) developed an integrated MRP system with a job shop simulator that responds quickly to changing requirements and has the capability of integrating heterogeneous manufacturing facilities. The UML is extensively used through the entire procedure. Enterprise Java Beans specification in addition to several emerging technologies such as XML and CORBA are presented. In terms of being responsive to customer, they showed that their system has a distinguished performance.

Pels (2006) proposed a data model called PDML (Product Data Modeling Language) as a tool to define classification hierarchies. Unified Modelling Language (UML) static structures are used as a foundational data modelling language. Capitalizing on UML static structure semantics (classification, aggregation, and generalization), PDML promises more natural models of product families and their complex product structures. The management of product data is supposed to be more generic, easier to understand, and less error prone.

The literature of object oriented MPC systems does not use the object-oriented principles to their fullest potential. Most of the review concentrate on the how to capture the semantics of manufacturing objects and illustrate the existing static relations. The application spectrum of MPC related problems is very limited in both variety and frequency. Most implementations are either limited to very few MPC specific problems such as scheduling or just conceptual frameworks.

2.4 MPC and RMS

Reconfigurable Manufacturing Systems (RMS) represent a new class of manufacturing systems which aims at combining the high throughput of dedicated manufacturing lines (DML) and the flexibility of flexible manufacturing systems (FMS) (Koren, Heisel et al. 1999). This could be achieved by fast scaling of system capacity and functionality in response to new circumstances (Mehrabi, Ulsoy et al. 2000). Setchi and Lagos (2004) defined "Reconfigurability" as the ability to repeatedly change and rearrange the components of a system in a cost effective way. In order to cope with the turbulent and uncertain market demand, The RMS technology capitalizes on continuous adaptation of manufacturing systems. The flexibility offered by Flexible Manufacturing Systems (FMSs) allows manufacturing a variety of products in the same systems; nevertheless, that comes at a price of acquiring a highly capital intensive system. In order to remain competitive under unpredictable and rapid changing market conditions, RMS promises flexibility coupled with responsiveness and cost efficiency. It also provides high reliability, scalability, and ability for easy software/hardware upgrades (Mehrabi, Ulsoy et al. 2000). An RMS is basically a mix of CNC machines, dedicated machines, and reconfigurable machine tools (RMTs) (Landers, Min et al. 2001). RMTs are modular machines that have flexible structures that allow changes of its modules via a group of well-equipped reconfigurable controllers integrated in an open-architecture manner. An RMS can be easily reconfigured at a system level, e.g. changing a configuration layout; machine level, e.g. adding a new spindle; and control level, e.g. integrating a new software module (Koren, Heisel et al. 1999). RMS is defined as a manufacturing system designed at the outset for rapid changes in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in response to sudden changes in market or in regulatory requirements (Koren, Heisel et al. 1999).

Amongst a number of manufacturing support systems, the manufacturing planning and control (MPC) systems are recognized as one of the pivotal infrastructures that firmly supports the organization's manufacturing to align with its higher level market strategy (Wacker and Hanson 1997). Thus, the emergence of RMS requires a new prototype or architecture of MPC systems that can address the changeable nature of manufacturing system and its surrounding environment. RMS with its changeable underlying structure was the first catalyst to spark the change-ready MPC systems project. Many innovations brought by PM were greatly inspired by the lack of having a full-fledged model of such systems. More on the RMS will be reported in chapter 6, when the Reconfiguration and Operations Planning is introduced.

2.5 Aggregate Production Planning

Since the late 1950s, Aggregate Production Planning (APP) has drawn the attention of operations manages, operations researchers, and management scientists. A plethora of research papers, surveys, and textbook chapters have been devoted to analyze and solve this problem. Since the aggregate production planning problem is used as a typical case problem more than once throughout this research, the most recognized problem models and solution approaches are introduced in brief in this section.

Buffa and Tabubert (1972) identified three pure strategies, out of which any APP strategy is considered a combination of some or all of them:

- Adjust the production rate through over-time/under-time.
- Adjust the workforce through hiring and firing.
- Maintain constant production levels by absorbing fluctuation of demand through inventory/backlogging or allowing lost sales.
- Additionally, subcontracting could be allowed.

According to the aforementioned list, Nam and Longendran (1992) identified two quality measures of an APP technique: a) The more adaptable the technique to all of these strategies listed, the more robust it is. b) The more limiting the data assumptions to implement these techniques have been, generally, the more apt the technique to provide an exact mathematical answer for the APP planner.

According to Nam and Logendran (1992) classification, APP techniques may be classified into two main categories: Optimal methods and Near optimal ones. Over the years, these techniques evolved from the very simple mathematical techniques, to today's models with sophisticated multiple objectives and advanced search heuristics. The remaining part of this section demonstrates some of these techniques.

2.5.1 APP Modeling and Solution Approaches

2.5.1.1 Linear models

Linear programming (LP) models try to identify the optimum production rate and workforce levels by minimizing the associated costs over the entire planning horizon. Silver (1972) summarized the basic assumptions underlying the LP APP models as follows:

1. Demand is deterministic
2. Production costs in any given planning period are strictly linear or piecewise linear
3. Costs incurred as a result of any changes to production rates in any given period are also linear or piece-wise linear
4. Inventory can be limited over the entire planning horizon
5. A single production facility serves a single market
6. back orders may or may not be allowed

Some examples of general LP formulations can be found in (Charnes, Cooper et al. 1955; Bowman 1956; Klein 1961; Fetter 1962; Laurent 1976; Meij 1980; Singhal 1989). Bowman (1956) suggested that fluctuations of sales can be accommodated by fluctuations in either production or in inventory or by some combinations of the two. The problem is formulated as a standard form of the transportation problem to make use of the powerful and efficient approaches used to solve these models. Production and inventory represent the source side while product demand represents the destination side. Many researchers followed this paradigm (Bishop 1957; Manne 1957; Akinc and Roodman 1986; Singhal 1989). The main pitfall of LP models is that linearity and deterministic demand undermine severely the applicability of these models. Piece-wise linearity has a very limited effect in alleviating the problem. Progressive Modeling in its philosophical part eliminates the linearity assumption in general; all the technical aspects also facilitate the non-linearity in all related models.

2.5.1.2 Linear Decision Rule (LDR)

Linear decision rule was suggested by Holt et al (1955). Unlike LP models, the cost function is quadratic, the demand is tacitly non-deterministic, all products are aggregated into a unique common product, and payroll costs are related to workforce and production rates. Moreover, and similar to LP models, production rates are proportional to workforce levels. Thus, setting the workforce and production rates determine the inventory levels and consequently inventory and shortage costs. Since the demand is stochastic, the incurred cost figure is the expected cost. As a result of differentiating that cost function, a two piecewise linear rules are determined to evaluate the workforce and production levels. Since the early model of LDR, several extensions have appeared. These models can be set apart based on considering them as single product (Holt, Modigliani et al. 1955; Holt, Modigliani et al. 1956; Khoshnevis and Wolfe 1983; Khoshnevis and Wolfe 1983) or multi-product (Bergstrom and Smith 1970; Chang and Jones 1970; Damon and Schramm 1972; Ebert 1976), and whether backlog is allowed (Holt, Modigliani et al. 1956; Ebert 1976).

Migrating from linear assumptions and assuming non-deterministic demand is considered a good step forward to create a more practical APP models. Nevertheless, a very cumbersome effort is needed to evaluate many constants and to formulate the cost function. Singhal and Adlakha (1989) found out that it is very hard to approximate the true costs of an industrial firm into a quadratic function. Additionally, no production and inventory constraints are allowed (Schild 1959). The linear decision rule was an outcome of the basic assumption of having a quadratic cost objective function. As already reported, PM is prepared for any non-linear model; the utilization of advanced optimization algorithms and CBSE enabled PM models to decouple the intricacies of the mathematical models from the solution algorithms. A highly novel, sophisticated, and synergistic models will be presented by PM and will be illustrated in the upcoming chapters.

2.5.1.3 Simulation Models

Simulation models gain a remarkable power when a complex cost structure is required and when other traditional models fall short in addressing complex problems (Lee and Khumawala 1974; Armacost, Penlesky et al. 1990; Zhang and Zhang 1999). Vergin (1966) used simulation to select parameters for APP decision rules. The simulation process starts with the current production plan followed by a change triggered by tweaking the workforce levels, overtime, production, inventory, etc. the solution process terminates when a local minimum of the objective function is obtained. Simulation models can be adjusted to many production circumstances; however, the computational effort is expensive and the quality of the obtained solutions is not as good as near optimal techniques. In this research, the advanced yet pragmatic logic brought by Progressive Modeling is designed to capture the logic that governs highly complex industrial problems. PM has the potential to be an alternative of simulation models. As would be revealed later in the second half of the dissertation, the ROP as a case problem shows how PM could replace simulation in many engineering applications.

2.5.1.4 Lot Size models

Lot size models address the production planning problem in the context of batch processing manufacturing environments. The lot size decision is an outcome of the trade-offs between lost productivity from frequent set-ups and short runs and higher inventory costs arising from longer production runs. Manne (1958) developed a model in which produced items compete for limited capacity under changing demand requirements. Linear programming (Manne 1958), dynamic programming (Beckmann 1961; Kao 1979), mixed integer programming (Newson 1975; Newson 1975) are the most prominent approaches to handle this problem. Most developed models consider multi-product environments; however, a few of them just allow backordering. Exact methods are limited to small size problems; generated solutions might produce shortages of some production items and sometimes capacity constraints are violated. Search decision rules are proposed to generate near optimal solutions for larger problems. Both MMAPP models and ROP consider lot sizing as an intrinsic part of production planning models developed. It is seminal to that research to make the lot sizing a seamless process during the planning activities. PM creates a network of logical capsules or mini-models that connects many decisions in a manufacturing environment in order to create what is defined by this research as the optimized tandem.

2.5.1.5 Goal Programming (GP)

In order to catch up with the business environment and attract the attention of industrial mangers who like to deal with a variety of objectives and goals, goal programming models were proposed (Leung, Yue et al. 2003; Leung and Chan 2009). All managerial objectives are incorporated as constraints in the suggested models and the objective is to minimize the linear sum of the goals deviations. Models varies by the number of and the type of goals considered such as production costs, workforce and inventory levels, marketing costs, etc. (Nam and Logendran 1992). the first implementation of GP to APP models was suggested by Lee and Moore (1974). The scope of GP models is much wider than other traditional cost objective models. Goals need to be identified and prioritized a priori. Some of GP models are linear based models (Lockett and Muhlemann 1978; Rakes, Franz et al. 1984), while others are HMMS/nonlinear based (Goodman 1974; Welam 1976). Linear GP models suffer from the same deficiencies of LP models mentioned before. However, the great advantage of the GP approach is its ability to promote the manufacturing platform performance. The concept of having goals of the system state variables is presented in this research as a part of system envelop constraints and all the problems considered are addressed from a multiple-objective perspective.

2.5.1.6 Search Decision Rule (SDR)

Search decision rule approach was suggested in order to overcome the limitations of linear decision rules (LDR) and linear programming models (LP). Understanding the sophisticated mathematics in many other techniques represented a real difficulty (Nam and Logendran 1992). In SDR, a computer simulation model of the system is developed and a response surface is searched using standard search techniques to obtain a near optimal solution (Taubert 1968; Goodman 1974). Search algorithms include, for example, applying Hooke and Jeeves search algorithm (Hooke and Jeeves 1961), combining search and branch-and-bound (Taubert 1968), and solving non-linear APP by sectioning search methods (Goodman 1973). SDR introduces a better flexibility in the modeling process; however, its capability to produce a good solution is limited by computer capacities and the complexity of developed models and their solution algorithms. PM brings more advanced and faster optimization algorithms that will be presented later to find better balanced set of compromising solutions.

2.5.1.7 Production Switching Heuristic (PSH)

The main premise underlying production switching heuristic (PSH) is that mangers favour one large change in work force versus a series of smaller and more frequent changes. The production and workforce are limited to a few discrete levels(Mellichamp and Love 1978; Oliff and Leong 1987; Barman and Burch 1989; Hwang and Cha 1995). Mellichamp and Love (1978) classified these levels to just three levels (high, normal, and low) all over the planning horizon and defined production and workforce decisions accordingly. The objective is to minimize any given cost function via an already chosen search procedure. PSH produces less frequent production and workforce schedules and better objective values can be obtained with a higher number of levels but this comes with an increased computational complexities.

The algorithms utilized in that research use the advancements of system envelop constraints, constraint satisfaction algorithms, and many others to create a highly powerful search algorithms that can generate multitudes of good candidates out of them the best is chosen according to a pre-specified selection criteria.

2.5.1.8 Other Approaches

Wang and Fang (2001) introduced fuzzy linear programming (FLP) method for solving the aggregate production planning (APP) problem with multiple objectives where the product price, unit cost to subcontract, work force level, production capacity and market demands are fuzzy variables. An interactive solution procedure was developed to provide a compromising solution. The proposed procedure allows a decision maker to model a problem according to the current information.

Phruksaphanrat et al (2006) commented on the formulation of aggregate production planning problem. Conventionally, a revenue function, a cost function, and a profit function are selected to be the objective function for the APP problems. They highlighted also that even though there are a lot of research work done on formulations of APP problems, there has been no investigation, which formulation is the most appropriate for APP problems. They argued that manufacturers should evaluate their performance by throughput.

Techawiboonwong and Yeneradee (2003) presented an aggregate production planning mathematical model for multiple product types where the system workers can be transferred among different production lines. The model was formulated in a spreadsheet format and a spreadsheet-solver technique was used as a tool to solve the model. They argue that an optimal aggregate production plan should provide the information on managing the available production capacity together with the useful workforce transfer plan. They showed that the total cost is significantly reduced when the workers are allowed to transfer among the production lines.

Ganesh and Punniyamoorthy (2005) formulated a general problem of continuous-time aggregate production planning for a given total number of changes in production rate throughout the studied planning horizon. They proposed a solution algorithm for the problem of continuous-time production planning using local search methods. Genetic algorithms (GA) and simulated annealing (SA) and hybrid genetic algorithms-simulated annealing (GA-SA) were compared for their performance. The results showed that the hybrid algorithm performs better.

Masud and Hwang (1980) analyzed and presented the APP from the multiple criteria decision making perspective. They used goal programming (GP), the step method (STEM), and sequential multiple objective problem solving (SEMOPS) to solve the APP problem considering the objectives of maximizing profit, minimizing changes in workforce level, minimizing inventory investment, and minimizing back-orders. Baykasoglu (2001) extended Masud and Hwang's model by allowing subcontracting and studying set-up decisions. A tabu search algorithm was developed to solve the pre-emptive goal programming model. A Multiple Objective Aggregate Production Planning Software (MOAPPS 1.0) was developed in order to compare Masud and Hwang's model with the extended model. The Multiple Objective Multiproduct Aggregate Production Planning (MMAPP) case problem studied in chapter 5 of this research share some basics of APP problem definition with both Masud and Hwang's and Baykasoglu's models.

2.5.2 APP shortfalls

APP techniques suffer from the lack of acceptance among practitioners in industry. Managers complain that they cannot readily comprehend the complexity of the analyses associated with these models (Gaver 1961; Galbraith 1969). Throughout recent decades, the number of proposed models and approaches has exploded tremendously, which has exacerbated the problem even. Existing techniques do not reflect the APP process in the real world since they are treated as a top-down constraint, while managers often regard it as a bottom-up approach (Silver 1967; Buffa and Taubert 1972). Another complains is the difficulty of aggregating several products into product families or product groups which necessitate some kind of homogeneity. Aggregating system resources—machines and personnel—suffer from the same problem. Machines may differ in their types and their process capabilities. Some workers are more valuable than others and they do not have equal opportunities when they are hired or fired. Data availability represent another obstacle: sometimes, it is not enough; and sometimes, it does not conform to the assumptions of linear and quadratic forms in various models (Groff and Muth 1972).

This long list of APP modeling shortfalls, which have also been identified and highlighted by so many earlier and later researchers, inspired and spurred the Progressive Modeling project. Creating and developing a new modeling approach that can create models that work was of utmost importance. An academic-industrial gap has been identified since then. The aforementioned gap and the immaturity of RMS are the key drivers of PM development throughout that research.

2.6 Evolutionary Multi-objective Optimization Algorithms

All the problems addressed in this research are analyzed and solved from a multi-objective perspective, which is a well-supported principle by change-ready MPC systems. The Evolutionary Multi-objective Optimization (EMO) Algorithms are selected as a well-established sorting and evaluation algorithms in the multi-objective space. EMO were developed in the early nineties based on combining the ideas of Pareto dominance. EMO algorithms differ by evaluating individuals' fitness or ranking population individuals, choosing and maintaining the elite among them, which is also known as elitism, and maintaining diversification during the search process. Early popular algorithms may include Multi-Objective Genetic Algorithm (MOGA) (Fonseca and Fleming 1993), Non-dominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb 1993), and SPEA Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler and Thiele 1999).

In MOGA (Multi-Objective Genetic Algorithm), the rank of each individual is based on the number of individuals by which an individual is dominated. The distribution of individuals over the Pareto front is performed by a fitness sharing procedure.

Srinivas and Deb (1993) introduced the first version of NSGA (Non-dominated Sorting Genetic Algorithm) in which the rank of each individual is based on the rank of the front it belongs. The distribution of individuals over the Pareto region is performed by a fitness sharing procedure.

In 1999, Zitzler and Thiele suggested SPEA (Strength Pareto Evolutionary Algorithm). The algorithm stores the best solutions found, Pareto front, in an external auxiliary population called archive. The rank of each individual is based on its strength factor. A clustering method (average linkage method) based on objective space is implemented to preserve the diversity of front members and avoid the use of any parameter such as the fitness sharing factor. SPEA incorporated elitism does not need any sharing parameter to be set and uses a fast non-dominated sorting algorithm, which makes it faster than many other algorithms.

Zitzler et al (2001) identified some weaknesses of SPEA and developed SPEA2 to overcome some of SPEA problems. Similar to SPEA2, NSGA-II was proposed by Deb (Deb, Agrawal et al. 2000; Deb, Pratap et al. 2002) to alleviate the difficulties associated with NSGA. Both NSGAII and SPEA2 became de facto standards of EMO algorithms and the most prominent to date. SPEA2 is implemented and encapsulated in the Optimizer component and shared all the problems presented by this research. SPEA2 algorithm is described in appendix A.

2.7 Summary

In this chapter, a review of the literature that serves the common purpose of this research was presented. MPC frameworks, reconfigurable manufacturing systems, aggregate production planning, and evolutionary multi-objective optimization algorithms are the main topics discussed. Other related review will be reported in other chapters whenever necessary. The next chapter introduces the change-ready manufacturing planning and control systems.

Chapter 3 CHANGE-READY MPC SYSTEMS

3.1 Introduction

Manufacturing Planning and Control (MPC) systems play a pivotal role in supporting business strategy and improving business performance. Lower production costs, better productivity, and distinguished customer service are some of the key values sought by any manufacturing firm. MPC acts as the manufacturing hub that links different system components, i.e. engineering activities, quality management, inventory status, sales etc. Some typical activities might include: aligning capacity with market needs; planning for on time raw materials delivery; maximize capital equipment utilization; maintaining appropriate inventories; scheduling production activities; tracking materials, people, customer orders, and other system resources; communicating with customers and suppliers on specific issues and long-term relationships; meeting customer requirements; responding when things go wrong and unexpected problems arise; and providing information for other functions on the physical and financial implications of the manufacturing activities (Vollman, Berry et al. 2005). The MPC system's design varies depending on the distinctive needs of manufacturing firms and different manufacturing processes. The system should evolve to meet changing requirements in the market, technology, products, and manufacturing processes. In order to prosper in today's global market, manufacturing planning and control systems should support the strategies and tactics pursued by successful manufacturing firms. The harmony between strategic, tactical, operational initiatives, and markets is fundamental (Olhager and Wikner 2000). Competitive priorities such as quality, delivery speed and reliability, price, and flexibility are vital for satisfying targeted markets supported by MPC systems. Berry and Hill (1992) presented a basic model that links the MPC system to its markets. The model has a great acceptance among scholars and practitioners. In that framework, there are links and choices at three levels of MPC system: At the master scheduling level, these choices are reduced to make-to-order, assemble-to-order (ATO), or make-to-stock (MTS). At the material planning level, the choices are twofold: rate-based and time-based. At the shop floor control level, the choices are either push or pull. Firms with high-volume standardized products in general would choose MTS, rate-based, and pull, whereas firms with many low-volume, customized products would choose MTO, time-phased, and push (Olhager 2003). In ATO environments, both are applicable to different sections of the plant. Grubbstrom and Olhager (1997) discussed it further to include market-related and product/process factors. The market-related is concerned with product related information: demand uncertainty and irregularity, product life cycle, commercial lead-time, and the market requirements heterogeneity within a company. The product/process factor comprises product and process complexity, number of production stages, degree of convergence, diversity of products per department, average utilization, etc. in a similar approach to the basic Berry and Hill model yet with additional a process complexity dimension, Bhattacharya and Coleman (1994) present another framework that addresses this link. The manufacturing process dimension is limited to discrete manufacturing ranging from highly complex job shop or batch type to low complexity flow shop and large batch processing. The strongest link between market requirements and manufacturing strategy concerns the process choice, which supports a firm's competitive priorities.

There are many market requirements, product characteristics, and the process choices that necessitate the MPC system to be inherently changeable. In stable manufacturing environments, rare to exist nowadays, mangers and practitioners can count on their intuition and experience to find appropriate solutions for the problems they might encounter. New challenges posed by today's global and unstable manufacturing competition urge a better MPC systems design and their governing philosophies. The new proposition, Change-ready MPC systems, set the foundation and characteristics to develop dynamic and system-oriented solutions to today's current MPC problems. Even though CMPC was suggested as an MPC evolvable model that should accompany reconfigurable manufacturing technology (Ismail and ElMaraghy 2009), the new CMPC proposition is still applicable to any manufacturing environment which is considered a major contribution of this research.

This chapter is organized as follows: change drivers in a manufacturing environment are presented. In this study, market, product, process, information technology, and industry of context are discussed first. In order to be resilient to these change drivers, a proposition of characteristics that define how an MPC component or functional unit can be described as change-ready is presented. CMPC frameworks and their tight relations to Component Based Software Engineering (CBSE) are further elaborated and some core components of the suggested CMPC system are briefly introduced. CMPC in a sense is both a design framework and a governing philosophy for what will be discussed in later chapters especially Progressive Modeling.

3.2 Change Drivers of MPC systems

3.2.1 Market

Market is the first class key driver of change for any manufacturing firm. Products features, process choice, and personnel involvement in the added-value process determine to what extent a product or a company can position themselves among the competition. In order to create a competitive edge, manufacturing firms allocates their resources and technical skills to maximize their profits and return on investments. With today’s global competition, the pressure to make the value creation process very dynamic became necessary. Manufacturing planning and control cannot stay stand still: the more responsive an MPC system to its market the better a competitive edge can be created. An MPC that remains unchanged for a long time may be inappropriate for market needs and eventually undermine its company competitive edge. In industry, it is a reactive practice to replace a functional component or a whole system through periodic review and evaluation of existing systems. Change-Ready MPC systems focus on making this characteristic a basic feature and an ongoing concern, i.e. proactive approach.

3.2.2 Product

Product features and how it can be manufactured and in what quantity it can be produced is a basic strategic decision. With today's global manufacturing, product life cycles are getting shorter and shorter. More pressure to reduce costs and customer insatiable requests for more features represents an extremely demanding pressure on MPC systems to be faster and more capable in adaptation and providing mangers and product developers with flexibility needed and cost trade-offs that can help them to create a better perceived value and a better brand image of their products.

3.2.3 Manufacturing Technology

Manufacturing systems have kept developing throughout the years; today's manufacturing is depending heavily on industry software packages to plan and control its operations. JIT, OPT, and MRP have become de facto standards and have proven success in so many industries. These technologies are built to fit many solutions by making their embedded algorithms very generic. Nevertheless, adapting such solutions needs a lot of effort and flexibility to tap their potential. A better way to develop MPC inherent heuristics and replace them with sophisticated models is strongly endorsed by this research. Making logic that governs the MPC functionality more tailored and sophisticated is a first class objective of defining the CMPC niche.

3.2.4 Information Technology

All manufacturing planning and control systems are software solutions unless we are talking about a very trivial system that can be executed by hand. Development in the information technology and software engineering has a direct impact on so many developments that have been made in the MPC world. In this research, several principles of Component Based Software Engineering, Object Oriented Programming, and Automata Based Programming are utilized as technical enablers of CMPC as would be described here and in later chapters.

3.2.5 Industry

MPC systems serve so many industries with each has its own level of volatility and pace of development. According to the industry of interest, a certain philosophy and characteristics may need to be satisfied; that makes what should be right and very effective in one industry may be inappropriate in another. Ability to reveal what are the characteristics and requirements of a certain industry identifies the models and the algorithms that can be developed to meet their specifications.

3.3 The New MPC System Characteristics

In order to manage change in a manufacturing system or its environment or both, Change-ready manufacturing planning and control systems (CMPC) have to evolve without losing stability that can undermine the underlying system/process strength. In order to hit a changeability-stability balance and to make a CMPC system a value adding component by itself, a set of characteristics has to be maintained:

3.3.1 Modularity

Modularity became a phenomenal characteristic in many domains: products, manufacturing systems, micro-chips, software systems, organizations and the MPC systems are not an exception. Modularity brings larger scale structures without losing manageability. CMPC are composed of loosely coupled sets of interacting components with its predefined and ready to adapt set of responsibilities and requirements. Components may encapsulate core competencies and core values in their logic. Components can be added/removed to extend/change the system capabilities.

3.3.2 Evolvability

Modularity facilitates system evolvability and is an essential prerequisite for changing needs. Change can be unmanaged, i.e. should be mitigated, or managed one—an action may be taken to create it. Examples of unmanaged change include abnormal conditions, demand fluctuations, launching new products by competitors etc. Extending system capabilities, scalability, switching strategies, policies, and procedures are examples of managed change. In order to address change, evolvability should be a culture more than just a characteristic. Evolvability instruments CMPC systems to recover from its shortfalls over time: less efficient algorithms, ignored parameters, violating some constraints and the like; furthermore, systems can be expanded and get more sophisticated, and closely customized to the underlying process, which epitomizes system scalability and development.

3.3.3 Balanced Performance

Promoting system performance is the ultimate objective of change-ready MPC systems. CMPC systems performance depends on the performance of its components and how synergies among these different components can be magnified. Most MPC literature works on cost as a sole objective; in practice, prices change overtime and profitability is not usually a fixed percentage of cost. Bottom line financials is also very vulnerable to inventory accumulations. Speed and reliability of orders delivery are of a main concern and consequently have major implications on the company competitive edge. In CMPC context, all its components should be aware of such holistic approach in defining its balanced goals or objectives. This should strengthen system sustainability, stability, and competency.

3.3.4 Socio-Technicality

The embedded algorithms should facilitate the interaction with system users, top management, and specialists in order to improve its performance and guide the solution process. Realizing peak performance, through well-defined mathematical models and solution approaches, is not sufficient in the next generation of MPC systems. The interaction with system users, especially the senior management, is sometimes required to find solutions beyond the system capabilities. Corrective actions, continuous auditing of system performance, reviewing functional strategies and policies are needed for all the activities of the proposed system. Industrial practice reports some failure stories about ERP systems and JIT implementations because of the false belief that well developed systems or philosophies are what make the difference. It is human beings and their engagement in harnessing the power and unleashing the potential of these systems is what makes the difference.

3.3.5 Universality

The implemented algorithms, models, guiding policies etc. should be general enough to respond to different scenarios and easily customized. With today's advanced optimization algorithms such as evolutionary algorithms, tabu search, swarm algorithms and advanced software technologies, a more advanced logic and models can be developed to replace today's simple intuitive heuristics.

The aforementioned characteristics were the main catalyst to develop Progressive Modeling (PM). PM represents a paradigm shift in developing a modern and forward looking methodology of analyzing and modeling industrial problems that capitalize on so many advances in optimization, software engineering, operations research, best business practices, and many related disciplines. PM is introduced to define how problems can be handled, solved, and developed in the CMPC context. The next chapters of this dissertation are dedicated for this part.

Since Component Based Software Engineering is a technological enabler of CMPC systems, it is presented in the next section and also contrasted against the object oriented approach.

3.4 Component Based Software Engineering (CBSE)

CBSE is the process of defining, implementing, and integrating loosely coupled components into systems (Sommerville 2004). CBSE emerged in the late 1990s as a reuse-approach to software system development. Component-Oriented Development (COD) enables systems to be constructed from pre-built components, which are reusable, self-contained blocks of code. These components have to follow certain predefined standards including interfaces, connections, versioning, and deployment (Heineman and Councill 2001). There are three major goals of Component Oriented Programming COP: conquering complexity, managing change, and reuse (Wang, Qian et al. 2005).

Conquering Complexity: COP provides an effective way to deal with the complexity of software: divide and conquer.

Managing change: Software engineers have come to the consensus that the best way of dealing with constant changes is to build systems out of reusable components conforming to a component standard and plug-in architecture.

Reuse: COP supports the highest level of software reuse including white-box reuse, gray-box reuse, and black-box reuse.

Component-enabling technologies such as COM (Box 1998), J2EE (Johnson 2002), CORBA (Pritchard 1999; Slama, Garbis et al. 1999), and .NET (Chappel 2006) provide the "plumbing" or infrastructure needed to connect binary components in a seamless manner, and the main distinction between these technologies is the ease with which they allow connecting those components.

3.5 Component-Oriented Versus Object-Oriented Programming

The fundamental difference between the two methodologies is the way in which they view the final application. In the traditional object-oriented world, all the classes share the same physical deployment unit, process, address space, security privileges, and so on. On the other hand, a component-oriented application comprises a collection of interacting binary application modules that are bonded to each other via well-defined protocols or interfaces.

Component-oriented applications usually have a faster development time because they can be selected from a range of available components, either from in-house collections or from third-party component vendors, and thus avoiding repeatedly reinventing the wheel.

Component-oriented programming promotes black-box reuse, which allows using an existing component without being concerned about its internals as long as the component complies with some pre-defined set of interface requirements. Instead of investing in designing complex class hierarchies which epitomizes the classical white box/gray-box of object oriented approach, component-oriented developers spend most of their time factoring out the interfaces used as contracts between components and clients (Bruccoleri, Amico et al. 2003).

3.6 Change-Ready MPC Frameworks and CBSE

CMPC frameworks may be broken down into components. These components have well-defined protocols that govern their communication. In order to make change an intrinsic characteristic, all these components can be modified or updated as long as they honour the purpose they created for and abide by the protocols defined among their boundaries. The CBSE as a first class enabler gives CMPC systems many enabling requirements out of the box. What is left for us is to define the inherent functionality and how it could be executed. In industry, large-scale solutions such as ERP systems provide generic solutions. CMPC is about highly customized and specialized solutions. This perspective gives CMPC systems an advantage to embed manufacturers' identities, cultures, and competitive edges in their custom built solutions. Every company has its own sails that can be adjusted to stay afloat and promote its competitive edge. The better the job that a CMPC component or a system can execute, the better the management and the control of the value created. Components improve manageability, better quality solutions, and much easier focused development.

Unified Modeling Language (UML) defines a component as a logical, replaceable part of a system that conforms to and provides the realization of a set of interfaces (Booch, Rumbaugh et al. 2004). An interface is a communication protocol between a couple of interacting components. Interfaces are like contracts; they should never be broken unless an approval is granted by all the stakeholders. As reported earlier, COM+, CORBA, .NET framework, and Enterprise Java beans are industry standards available to implement component-based solutions. Nevertheless, CMPC is concerned with the logic regardless of the implementation that may be used. Therefore, a well-designed spreadsheet for a small manufacturing enterprise can give the same functionality and advantages similar to a high-end software solution. The job is to maintain the same CMPC mindset and have a strong grip of the system at hand.

A CMPC system is usually composed of a group of subsystems: an input/output (user interface system), core MPC subsystems, support subsystems, and data subsystems. The input/output subsystem is the part that connects the whole CMPC system to its users. Depending on the implementation utilized, both the input and output can be defined in many ways; for example during that research, the problem data was hard coded while the output was exported to txt files. Excel can be used as a COM server to illustrate the results for post analysis. For individual research projects, like those ones conducted in this study, this could be an appropriate solution. In a commercialized or team-projects, definitely a more sophisticated I/O ways should be preferred.

The core CMPC components are the ones that achieve or accomplish a well-defined function that directly related to common MPC activities. Demand management, aggregate production planning, capacity planning, scheduling are some examples. In the next section, the light would be shed on some of this component from CMPC perspective.

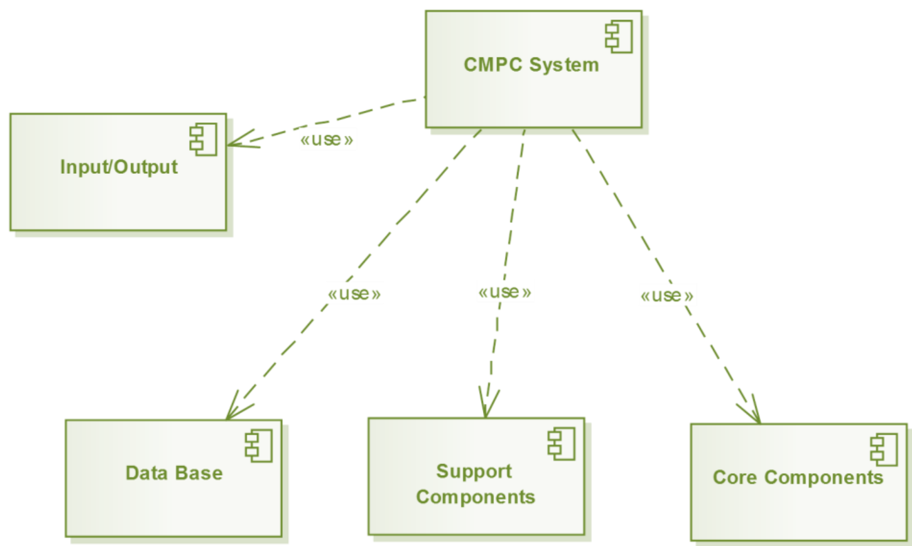


Figure 3-1: CMPC components

The third component is the support subsystem(s). Common tools that can be shared by core components can be embedded as integrated parts of these components: optimizers, forecasting models, and statistical tools can help the core components in facilitating their functionalities.

The last component is the system data available. It could be a high-end data base system or a small data files. The size of the manufacturing firm under study is the one that decides.

A change ready MPC framework wires all components together in a way that promote further development. Therefore, a well-prescribed and crisp definition of a component or system functionality is urgent for a successful implementation of a CMPC system. The most distinguished feature of CMPC systems, other than its new in-house logic development and its intrinsic change-ready characteristic, is the sophisticated logic that should be encapsulated inside its components. Progressive Modeling, a forward-looking modeling approach proposed by this research, is a key enabler of CMPC systems. All the remaining chapters of this dissertation will be dedicated for PM and illustrate how it works.

3.7 CMPC core components: Some Examples

3.7.1 Demand Management

Accurate and timely demand plans are a vital component of any good MPC system. Inaccurate demand forecasts should result in system imbalance between demand and supply and unsatisfied customers. In planning contexts, both long-term and short-term forecasts are needed. Inaccurate forecasts in the short-term means lost sales, lost customers, excess inventories and the like. Statistical models, such as time series forecasting, may be a good solution for short-term forecasts. Integration with other system components can even solve some of the short-term forecasting inadequacies, such as promotion and advertising, better effort of the sales-force, and the like. A CMPC system is a part of a wider system that has some levers that could counter the effect of inadequate demand forecasts. Therefore, time series forecasts could be a good and sufficient choice at this level.

Long-term forecasts are very important for capacity planning and mid-term initiatives. Based on these forecasts, resource related decisions could be made: people can be hired and fired, capacity— especially, in an RMS environment— can be scaled up/down, backordering, and subcontracting decisions can be planned. Causal models such as regression models can be used for this kind of forecasting. Unlike statistical models, forecasting using artificial neural networks became much popular nowadays. ANN forecasts have the ability to capture demand nonlinearity and do not assume a specific functional relation between the input data set and the resulting forecasts. Both the statistical and Artificial Neural Network (ANN) forecasting could be embedded in a demand-forecasting component.

Change-ready MPC systems encourage cross boundaries solutions given by synergistic relations among manufacturing enterprise functional units, sales, marketing, and operations, to overcome problems result from insufficient logic such as forecast errors and unexpected surprises or obstacles.

Sales and Operations Planning (S&OP)

Most top-level decisions and performance measures (production levels, service levels, capacity levels, inventory levels, and others) are decided through this critical component. S&OP is a new industrial practice that appeared and became prevalent in recent years. The objective of S&OP is to hit a balance between the demand and supply. Critical decisions like demand mix and demand volume are identified by S&OP. it is no wonder that S&OP is considered a top component under which demand, operations planning, and resource management have to be synchronized.

In this research, the APP progressive models bring better pragmatic solutions that would provide the top management and S&OP a group of well-crafted solutions from which better alternatives and highly effective decisions can be made. The Reconfiguration and Operations Planning (ROP) problem presented later is a new esoteric and holistic version of both capacity and operations management in an RMS environment. PM brings a new concept called "Optimized tandem" in which a highly educated S&OP decision can be made. The outcome is a best-balanced decisions set that maximize the system performance criteria.

3.7.2 Resources Management

Better resource management is a key value driver in terms of systems profitability, stability, and key performance indicators KPIs. APP problems are actually mid-term capacity management and operations planning problems. Having such numerous numbers of publications with very limited applicability was the greatest catalyst to develop the Progressing Modeling approach presented by this research. PM can create new models that should be effective and should enable manufacturers to leverage their resources without losing system stability.

3.7.3 Other Components

Other core components might include MPC inventory control (used to decides and monitor inventory levels), Master production schedule (which takes the output of sales and operations and disaggregates it into weekly production plans), materials requirement planning, and production activities control. CMPC promotes ingrained logic and PM is simply the methodology that drives the development and the implementation of that logic.

3.8 Summary

In this chapter, today's dynamic changes and its relations to Manufacturing Planning and Control systems were discussed. Many contradictory and conflicting issues could push MPC into quite disordered zones: the market that need harmonizing the strategic strength and strategic scope, the product that materializes all the efforts being done in the background, the process that make the product, and the competition where so many surprises can pop out. Change is constant and the question remains how to be change-ready and how our sailings could be adjusted accordingly to stay afloat. As an initiative to answer this question from the MPC perspective, a new MPC framework was proposed—Change-ready MPC systems. The CBSE is utilized as an enabling technology for the new MPC proposition. Change drivers are identified and the new framework is designed to be aware of both expected and unexpected changes and to be ready for these changes both reactively and proactively. CBSE empowers the new system as well as the system management with many characteristics that enable the design and the implantation of the proposed system.

Even though Change-ready MPC systems are designed to be versatile and encapsulate many sophisticated algorithms, they are not able to handle these new challenges by themselves. Human interaction and collaboration with other enterprise level sub-systems is of essence to stay resilient in front of these new challenges. The aforementioned human-system interrelation is made intrinsic by defining the socio-technicality as a core CMPC characteristic.

CMPC is about the mindset, the culture, and the design aspects of CMPC systems and components. Progressive Modeling, presented in the remaining chapters, will jump into the black boxes to describe how to develop and manage their logic and make it ready for many changes.

Chapter 4 PROGRESSIVE MODELING I: AND THE FIRST APPLICATION

4.1 Introduction

The prevalence of change and how it propagates form the outermost scope of business strategies to the lowest level of functional areas of Manufacturing Planning and Control (MPC) systems, and vice versa requires more dynamic and adaptive modeling and analysis approaches. Progressive Modeling (PM) started as an initiative to address many industrial problems in today's dynamic manufacturing environments from systems perspective. The proposed approach adopts the concepts of Component-Based Software Engineering (CBSE) (Sommerville 2004; Wang, Qian et al. 2005) to analyze MPC problems and decompose them into several fundamental interacting components. The problem at hand is analyzed from systems perspective and deployed into several interacting components that should have well-defined functions and embedded models. These models are linked to a solver in order to control the whole process. The objective is to find a balanced set of alternatives that can be presented in an appropriate format to decision makers in order to help them to monitor, promote, and optimize the whole system performance.

Every component has its own set of interfaces that represents sub-set of the specifications of the system, or problem under study. In this study, MPC problems are treated as if they were systems. The componentized nature of developed system emphasizes the model design, functionality, and modularity, and de-couples their detailed implementation. This allows implementations to be updated to reflect model changes to be commensurate with variations in the MPC system.

The mathematical model specifications go beyond what is known as model assumptions by introducing the concept of assumption relaxations. This represents one of the basic requirements to make developed models more realistic and ready to be re-modeled or updated in the future as conditions or boundaries may be changed. A set of objectives should be defined a priori regardless of the subsequent evaluation methods (e.g. linear or non-linear). Similarly, constraints and their formulation may be added, modified, or removed readily. Likewise, variables can be integer, binary, or real numbers. Non-linear, rather than linear, modeling is the default.

Intelligent optimization techniques, such as Genetic Algorithms, Artificial Neural Networks, and Tabu search, are typical solution algorithms. Unlike exact methods, these techniques are loosely coupled with the problems and their assumptions and their capabilities can be independently up-graded as needed as better solution algorithms become available.

This chapter is organized as follows: First, the propagation of balance as a governing philosophy is presented. The Aggregate Production Problem as already described in earlier chapters is chosen as an application problem used to illustrate the new perceptions and advancements brought by PM. PM Process is presented and illustrated by applying it to the APP problem. A numerical example is presented and results are discussed.

4.2 Propagating the Balance: a PM Governing philosophy

Change in manufacturing environments propagates from markets to products, manufacturing system, process planning, manufacturing planning and control (MPC) and enterprise organization. The changes on these multiple fronts do not occur in isolation but are often interdependent. The real challenge is to reach and maintain a balance among all hierarchical levels in order to stay competitive in today's turbulent manufacturing environment.

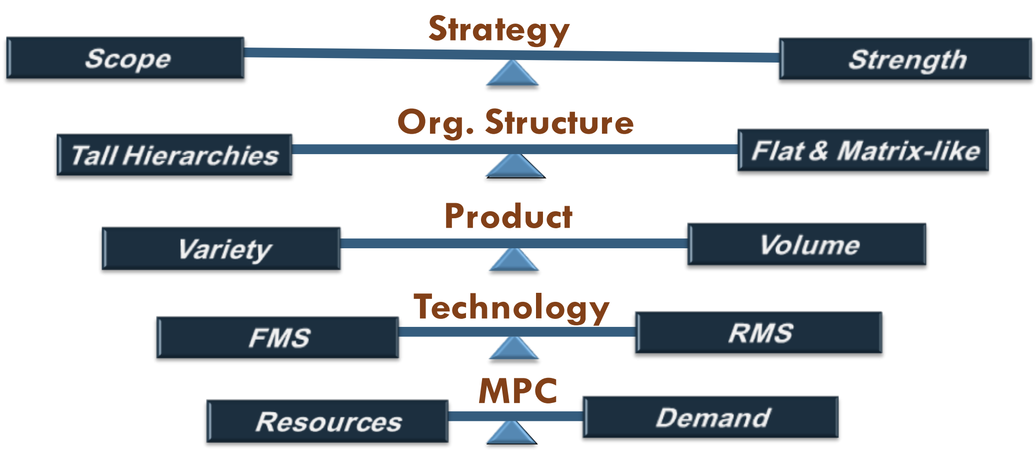


Figure 4-1: Maintaining the balance at all levels

Companies strive to excel at the strategic scope and strategic strength dimensions in order to achieve a competitive advantage. The strategic scope focuses on the composition and size of the target market and strategic strength considers the core competencies of the manufacturing enterprise.

With many advances in today's information and communication technologies, there is a clear shift from taller hierarchies to flatter and matrix-like organization structures that leads to improve responsiveness, autonomy, and increase the ability of manufacturing enterprises better address these changes.

From Products decision perspective, deciding on product quantity/variety, i.e. economies of scope versus economies of scale, places certain constraints on the design of manufacturing systems and their production control strategies. Mass-customization is growing rapidly with serious attempts to lower prices. Companies now compete on being both responsive and efficient. A mix between agile and lean practices is essential to fit these new requirements.

Advances in manufacturing technologies move the changeability boundaries and its limits forward, i.e. Reconfigurable Manufacturing Systems (RMS) with its incremental change of functionality and capability versus Flexible Manufacturing Systems (FMS) with built-in abilities to change its functionality within a pre-defined scope. The future changes and the evolution of RMSs is by definition uncertain at the outset. Today's manufacturing systems need co-evolving MPC systems able to adapt market changes and products requirements both efficiently and effectively.

MPC systems represent a gateway between the manufacturing system resources or supply side and its environment (i.e. market or demand). The ability of an MPC system to capture and achieve the balance between those competing goals is a real challenge. Maintaining the balance at all fronts (strategies, organization structure, products, technologies and MPC systems) and under varying conditions governs the driving philosophy of Progressive Modeling. The goal is to remove the restrictive and problem- or solution-specific constraints and embrace modular component-oriented design to provide future possibility for modifying or replacing any function or module without changing the pre-designed and streamlined system structure and components' interaction protocols and specifications. This approach maximizes the flexibility and changeability of MPC systems in light of changes in objectives, models, solution methods, and data. This newly developed Progressive Modeling methodology has been implemented and is applied in this chapter to aggregate production planning as an illustrating example.

4.3 Aggregate Production Planning: a brief Introduction

Aggregate production planning (APP) is a mid-term capacity planning system responsible for transforming forecasted sales and system resources (machinery and personnel) into feasible operation plans for the following 6 to 18 months. The goals of production planning are to define a combination of production rates, inventory patterns, workforce levels, reduce production costs, achieve required customer service levels, smooth-out resource fluctuations, and maximize resources utilization. Development of production plans starts with identifying the long-term objectives, analyzing existing marketing strategies and estimated demand, analyzing available resources and adjusting them to meet the fluctuating demands. Production operation plans and resource schedules that are able to hit a balance among all these objectives represent a real challenge in the existing turbulent environment. Nam and Logendran (1992) conducted a survey of APP techniques and identified the most frequently used techniques including: 1) Trial and error methods, 2) Graphical techniques, 3) Parametric production planning, 4) Production switching heuristic, 5) Linear programming, 6) Goal programming, 7) Mixed integer programming, 8) Transportation method, and 9) Simulation models. More recent research adds AI optimization, Decision support systems, and fuzzy logic to the list. A detailed APP literature review is already introduced in chapter two (section 2.5) for the interested reader.

Stockton et al. (1995) analyzed existing models limitations and solution techniques and pointed out that: None of the existing APP techniques can identify optimal or near optimal plans for real world problems that involve a range of planning variables. Also, those techniques that can identify optimal plans do so by achieving only cost-related objectives ignoring many other non-cost objectives often sought by managers. In addition, within many organizations the cost relationships used by these methods do not adequately represent actual costs. The mathematical procedures used by existing methods are also complex; hence, managers are often reluctant to use such techniques in practice. The proposed progressive modeling approach addresses these shortcomings in addition to the need to adapt, incrementally and progressively, as needed and when needed, to the frequent variations and changes.

4.4 Progressive Modeling: The Process I

The Progressive Modelling approach can be summarized into three main steps: Analyze the problem at hand, build the mathematical model, and define the solution methodology. As an example, aggregate production problem is considered to illustrate these principles. The remaining parts of this chapter show in details how these principles are applied and the new potential that Progressive Modelling introduces to the industrial research field. The process presented in this chapter was the first PM version. A better and a more generalized version will be presented later in chapter 5.

4.4.1 Analyze the problem

An Aggregate production planning system is visualized as an MPC component that keeps the balance between the manufacturing system resources and its output represented by products as depicted in Figure 4-2.

Using The CBSE principles, the APP system is decomposed into several interacting components: Modeler, Products, Workforce, Machinery, Optimizer, and User Interface. Products encapsulate all product data related definitions. Workforce & Machinery components hold resources data. The modeller executes the logic of the APP mathematical model. It encapsulates several components that are responsible for generating, evaluating, and optimizing feasible APP plans. The user interface component isolates system users from the internal intricacies and displays the results in an appropriate format.

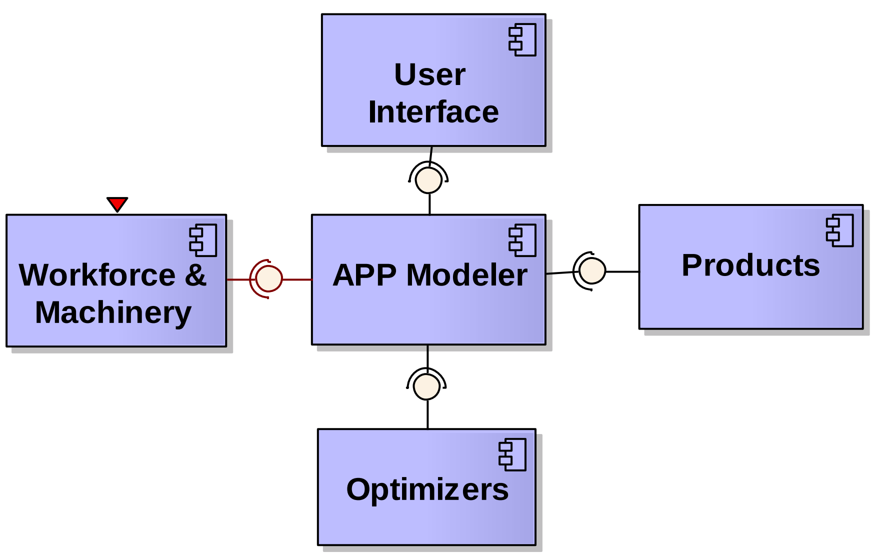


Figure 4-2 APP simplified component diagram

These components interact with each other via a well-defined set of protocols called interfaces. Every interface is composed of a set function definitions of inputs and outputs. Only these interfaces can be implemented in a variety of ways via their implementing classes. This kind of decoupling is an essential feature of progressive modeling. For example, hiring and firing costs may be modeled by linear or quadratic functions or even higher-order non-linear functions at different chronological model development stages. The interface definition never changes regardless of the utilized evaluation methods. Consider hiring and firing costs for example, the workforce schedule and cost factors are the inputs and the total cost is the output. This is called function black boxing. It allows better management of model evolution and handles ill-formulated problems in a progressive fashion. Therefore, resorting to the simplifying assumptions of linearity is not needed at the outset, which increases the flexibility and scope of problems to be handled.

Every component has a set of provided/required interfaces. A product component, for example, provides the following Modeler interfaces: IDemand, ISetUp, ISubContract, IInventroy, and IBackOrders (Figure 4-3). These interfaces control data validation and all related objectives to be evaluated. There are specification and implementation components. The specification components are defined once prior to building the math model. The implementation components, however, can be replaced and updated as needed.

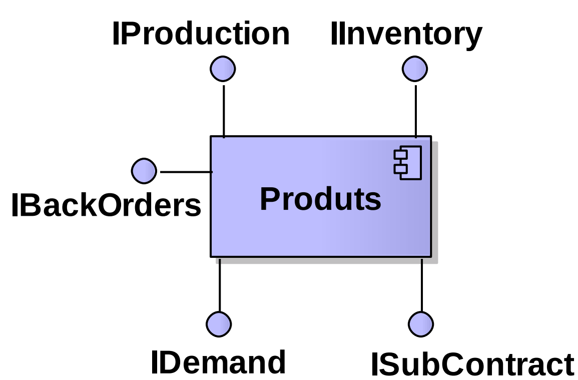


Figure 4-3: Products module/component interfaces

4.4.2 Build the Mathematical Model

The mathematical model has built-in flexibility to define several objectives and constraints that can be updated dynamically by replacing the implementation component(s). Function templates are introduced to define ill-formulated or hard to define objectives or constraints or those that need better research or future updates. The templates define the input and output as shown in equations (4.1) through (4.4) in the next APP model. Function templates represent a mathematical equivalent to a component interface function. Similar to interfaces, function templates can be represented using several formulations and should be the least changed.

4.4.2.1 APP model definition

Notation

T number of planning periods

C_r regular time production cost per unit in period t $(\$/\text{unit})$

C_o overtime production cost per unit in period t $(\$/\text{unit})$

C_h inventory holding cost perunit in period t $(\$/\text{unit})$

C_s subcontracting cost per unit in period t $(\$/\text{unit})$

C_b backorder cost per unit in period t $(\$/\text{unit})$

C_F firing cost per worker in period t $(\$/\text{worker})$

C_H hiring cost per worker in period t $(\$/\text{worker})$

R_t regular time production volume in period t (units)

O_t overtime production volume in period t (units)

S_t subcontracted volume in period t (units)

$I_{\max}$ maximum inventory level in period t (units)

B_{max} maximum backorder level in period t (units)

$S_{\max}$ maximum subcontracted volume in period (units)

$W_{\max}$ maximum allowed workforce level (worker)

Δw prefered Incremental workforce change

W_0 initial workforce level (worker)

I_o initial inventory (units)

δ worker's productivity (unit/hr)

β number of regular hours per worker in a planning period (hrs)

γ number of overtime hours per worker in a planning period (hrs)

Decision Variables

P_t = total product supply in period t (units)

W_t = workforce level in period t (workers)

Objectives Templates

$$Min / Max\ Z_1 = g_1(R_t, O_t, C_t, I_t, B_t, r_t) + g_2(W_t, H_t, F_t) \quad \textbf{(4.1)}$$

$$Min\ Z_2 = g_2(W_t, H_t, F_t) \quad \textbf{(4.2)}$$

$$Min\ Z_3 = g_3(I_t) \quad \textbf{(4.3)}$$

$$Min\ Z_4 = g_4(I_t) \quad \textbf{(4.4)}$$

:

Constraints

Product Balance

$$I_t = I_{t-1} + P_t - D_t \quad \forall t \quad \textbf{(4.5)}$$

$$P_t = R_t + O_t + S_t \quad \forall t \quad \textbf{(4.6)}$$

$$S_t \leq S_{\max} \quad \forall t \quad \textbf{(4.7)}$$

$$\max\{0, I_t\} \leq I_{\max} \quad \forall t \quad \textbf{(4.8)}$$

$$\max\{0, -I_t\} \leq B_{\max} \quad \forall t \quad \textbf{(4.9)}$$

Workforce-Product Balance

$$R_t \quad \leq \quad \delta\beta W_t \qquad \forall t > 1 \tag{4.10}$$

$$O_t \quad \leq \quad \delta\gamma W_t \qquad \forall t \tag{4.11}$$

Workforce Balance

$$W_t \quad \leq \quad W_{\max} \quad \forall t \tag{4.12}$$

$$W_t \quad = \quad W_{t-1} + H_t - F_t \quad \forall t \tag{4.13}$$

$$H_t F_t \quad = \quad 0 \qquad \forall t \tag{4.14}$$

$$: \tag{4.15}$$

$$P_t, R_t, O_t, S_t, H_t, F_t, W_t, I_t, B_t \in N^+ \tag{4.16}$$

4.4.2.2 Model Description and Implementation Example

Several objectives can be defined related to financials, workforce variability, inventories, and customer service level. Financial considerations could be represented by a profit or cost function. If the price is constant over the planning horizon or if it is a fixed mark-up percentage of incurred costs, then using a cost function would be sufficient. Financials are decomposed into resource-related and cost-related objectives (Eq. (4.17)).

Product related financials could be decomposed into revenue, materials, overtime, subcontracting, holding and backordering costs. Workforce financials include hiring and firing and payroll costs. Managing Workforce variability is an essential resource side objective as well as minimizing workforce force (considering both hired and fired workers). Variability could be measured by evaluating the variance of the independent workforce variable (Wt). However, mangers in practice prefer having few major discrete changes in workforce compared to continuous minor changes. Progressive modeling (PM) aims to capture these practical considerations by using a negative exponential function. The goal is to achieve a good workforce profile compatible with best industrial practice. Minimizing inventories and their holding costs enhances the company financial health by promoting lean manufacturing practices. Better customer service can be achieved by minimizing the back orders variables as it promotes agile practice. Therefore, the four objectives can be formulated as follows:

Templates Sample Implementation

Optimize Financials

$$Max\ Z_1 = \sum_{t=1}^{T} \left\{ \begin{array}{c} \underbrace{\left\langle p_t P_t - \begin{bmatrix} C_m (R_t + O_t) + C_o O_t + C_s S_t + \\ C_h \max\{I_t, 0\} + C_b \max\{-I_t, 0\} \end{bmatrix} \right\rangle}_{\text{Product financials}} - \\ \underbrace{\left\langle C_w W_t + C_F F_t + C_H H \right\rangle}_{\text{Workforce financials}} \end{array} \right.$$

$$or \qquad \textbf{(4.17)}$$

$$Min\ Z_1 = \sum_{t=1}^{T} \left\{ \begin{array}{c} \underbrace{\left\langle \begin{bmatrix} C_m (R_t + O_t) + C_o O_t + C_s S_t + \\ C_h \max\{I_t, 0\} + C_b \max\{-I_t, 0\} \end{bmatrix} \right\rangle}_{\text{Product financials}} - \\ \underbrace{\left\langle C_w W_t + C_F F_t + C_H H \right\rangle}_{\text{Workforce financials}} \end{array} \right.$$

Optimize Workforce Profile

$$Min\ Z_2 = \sum e^{-\frac{(Ht+Ft)}{\Delta w}} \qquad \textbf{(4.18)}$$

Minimize Total Inventory

$$Min\ Z_3 = \sum \max\{I_t, 0\} \qquad \textbf{(4.19)}$$

Minimize Total Backorders

$$Min\ Z_4 = \sum \max\{-I_t, 0\} \qquad \textbf{(4.20)}$$

Equations (4.5) through (4.16) represent model constraints and more constraints such as the end of planning horizons workforce levels and inventory levels can be added. There are product, workforce-product, and workforce constraints groups. A constraint manager (a sub-component of the Modeler) is responsible for ensuring solutions feasibility. Equation (4.5) handles the balance between demand, product supply, inventory, and backorders. Equation (4.6) decomposes the total product supply into regular, over-time, and sub-contracted volumes. Inequalities(4.8)-(4.9) check the inventory upper bounds, subcontracted and backorders volumes. Inequalities (4.10) and (4.11) ensure that produced quantities are within the available resources. Inequality (4.12) sets the upper bound of the available workforce. Equation (4.13) handles workforce balance; and finally, Equation (4.14) ensures that hiring and firing are mutually exclusive.

4.4.3 Develop the Solution Approach

Since a progressive math model is, by design, not fully defined a priori, a generic optimizer to handle different potential model versions is needed. AI optimization techniques, such as genetic algorithms, are best suited for this purpose. An Evolutionary Multi-objective Optimization (EMO) algorithm is used in this study for illustration purposes. EMO has never been applied to the multi-objective APP problems before, and it is used for its ability to generate simultaneously optimized sets of solutions i.e. Pareto front. For interested reader, Appendix A provides some foundations related to the SPEA2 algorithm implemented in this research.

A chromosome represents a feasible plan for an APP problem. An APP chromosome is coded as a string of composite genes. Every gene is composed of two parts/values, representing the total product supply and the available workforce, which propagates the balance even at this low level. Figure 4-4 shows an example and its decoded solution. Every solution represents a product plan and workforce plan. To decode a chromosome, for every planning period, the total product supply (Pt) is decomposed into regular part (Rt), over-time part (Ot), and sub-contracted part (St) respectively. The value of (Rt) is determined after evaluating its maximum value by checking the value of workforce (Wt) and transforming it into its equivalent production units (Rt). If the value of total product supply (Pt) exceeds the available regular production (Rt), the over-time variable is used for the residual (Pt-Rt). If the value of the total available over-time hours is consumed, the remainder is sub-contracted (St). The inventory and back orders are updated according to the required demand. For the workforce plan, the value of workforce Wt is checked against its preceding period value (Wt-1) and the values of workers hired (Ht) and workers fired (Ft) are updated accordingly.

Chromosomes that satisfy the model constraints are randomly generated. After decoding chromosomes into their equivalent plans, objectives are evaluated and passed on to the optimizer. The optimizer uses the Strength Pareto Evolutionary Algorithm SPEA2 (Zitzler, Laummans et al. 2002) (EMO algorithm) to evaluate individuals fitness and update the Pareto front, which represents a non-dominated set of best solutions. The Pareto front size (called archive size by SPEA2) is the number of solutions in the set and is determined a priori. SPEA2 uses some internal diversification and clustering algorisms to maintain a fixed archive size. Once the selection process is done, the recombination process starts using cross over and mutation operators.

	Jan	Feb	Mar	Apr	May	Jun
Total Product Supply P	**2595**	**2743**	**2756**	**2632**	**2592**	**2769**
Workforce W	**54**	**54**	**54**	**70**	**55**	**55**

a) Sample Chromosome

	Jan	Feb	Mar	Apr	May	Jun
Demand	1600	3000	3200	3800	2200	2200
Regular Units R	2160	2160	2160	2632	2200	2200
Overtime Units O	135	135	135	0	137	137
Subcontracted Units S	300	448	461	0	255	432
Inventory I	1995	1738	1294	126	518	1087
Back orders B	0	0	0	0	0	0

	Jan	Feb	Mar	Apr	May	Jun
Workforce W	54	54	54	70	55	55
Workers Hired H	0	0	0	16	0	0
Workers Fired F	26	0	0	0	15	0

b) De-coded Product and Workforce Schedules (Plans)

	Name	Mode	Value
1	**Back Orders**	Min	0
2	**Inventory**	Min	6758
3	**Total Costs**	Min	472782
4	**WF Variability**	Min	0.1063

c) Evaluated Objectives

Figure 4-4: APP Chromosomes, Plans, and Objectives

If a cross over operator fails to produce a feasible solution that satisfies all constraints, a maintenance algorithm intervenes to obtain a feasible one instead. All mutation operators produce feasible solutions. The algorithm iterates the selection and recombination steps N number of generations until it stops.

4.5 Numerical example

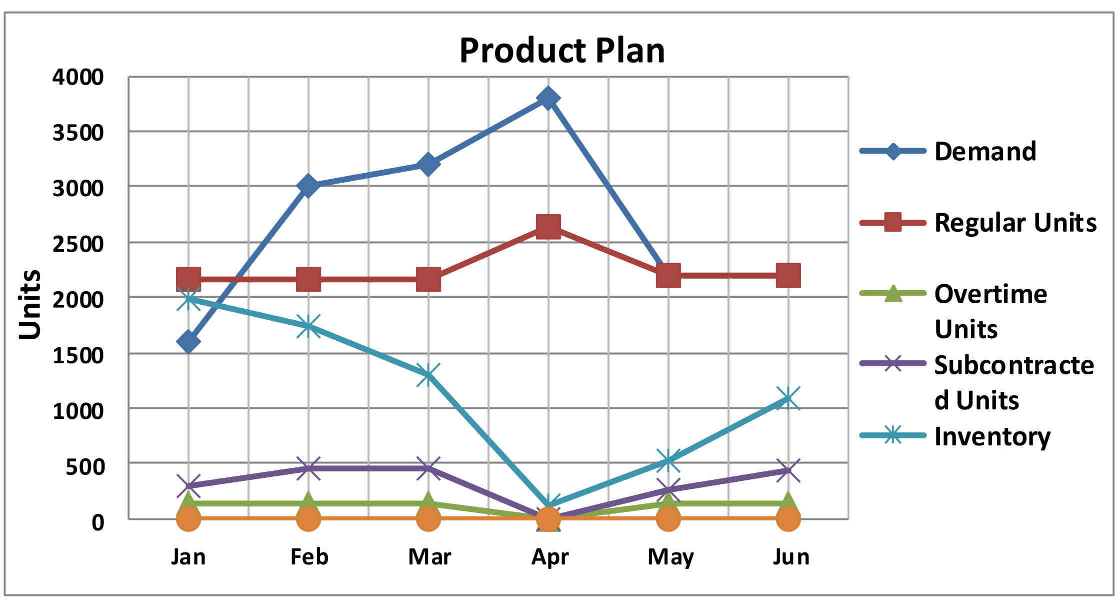
Table 4-1 and Table 4-2 include the test problem data (Chopra and Meindl 2007): Demand forecast, cost related data, and initial conditions. These are suitable for the developed multi-objective form, in addition, the preferred workforce incremental change (ΔW_t) that was defined to be 5. Some genetic algorithm parameters are used including: population size =100, archive size =20, number of generations=1000, and 25% of population size is chosen to be reproduced at every generation.

Table 4-1: Forecasted Demand (Chopra and Meindl 2007)

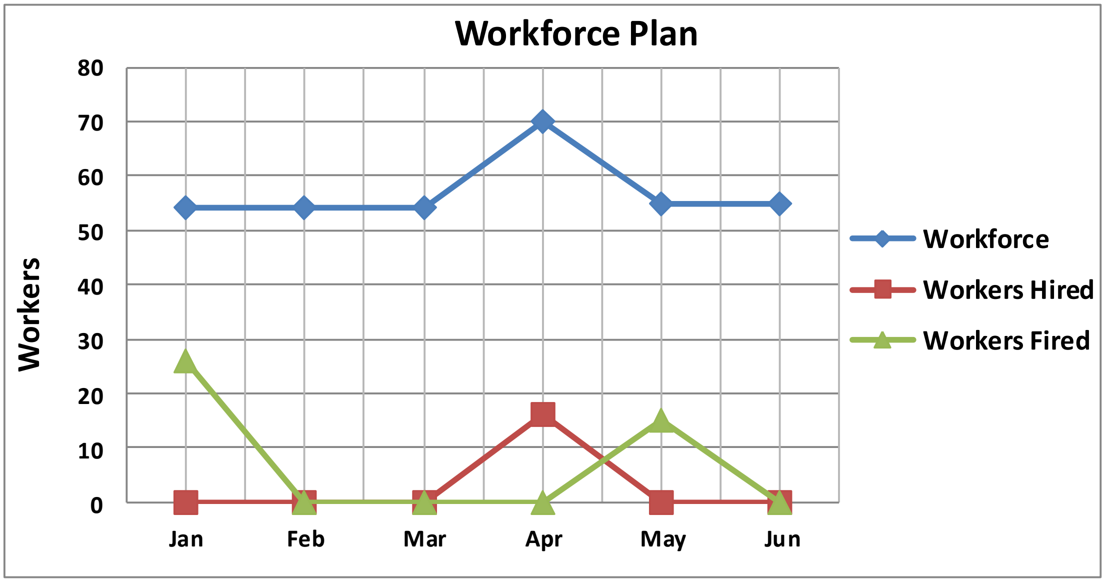
	Jan	Feb	Mar	Apr	May	Jun
Demand	1600	3000	3200	3800	2200	2200

Table 4-2: Cost related data and initial conditions (Chopra and Meindl 2007)

Item	Value
Materials cost/unit Cm	10
Inventory holding cost/unit/month Ch	2
Marginal cost of stock out/unit/month Cb	5
Hiring and training cost/worker CH	300
Layoff (firing) cost/worker CF	500
Labor hours required/unit	4
Regular time cost/month Cw	640
Over time cost/hour Co	6
Marginal subcontracting cost/unit Cs	30
Initial inventory (units)	2000
Initial workforce (workers)	80

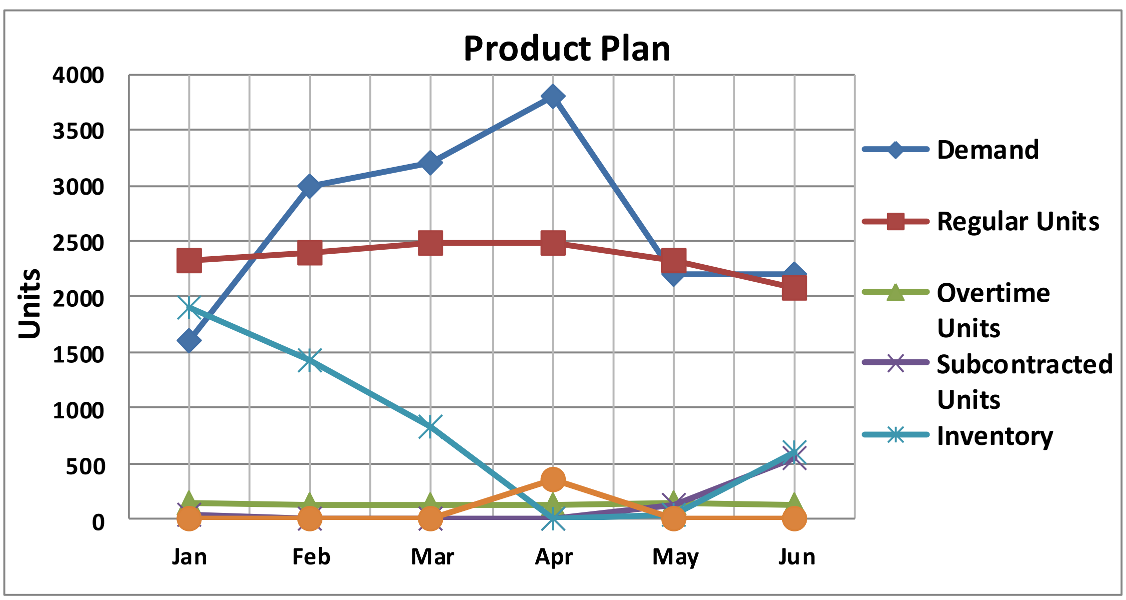


d) Product Plan

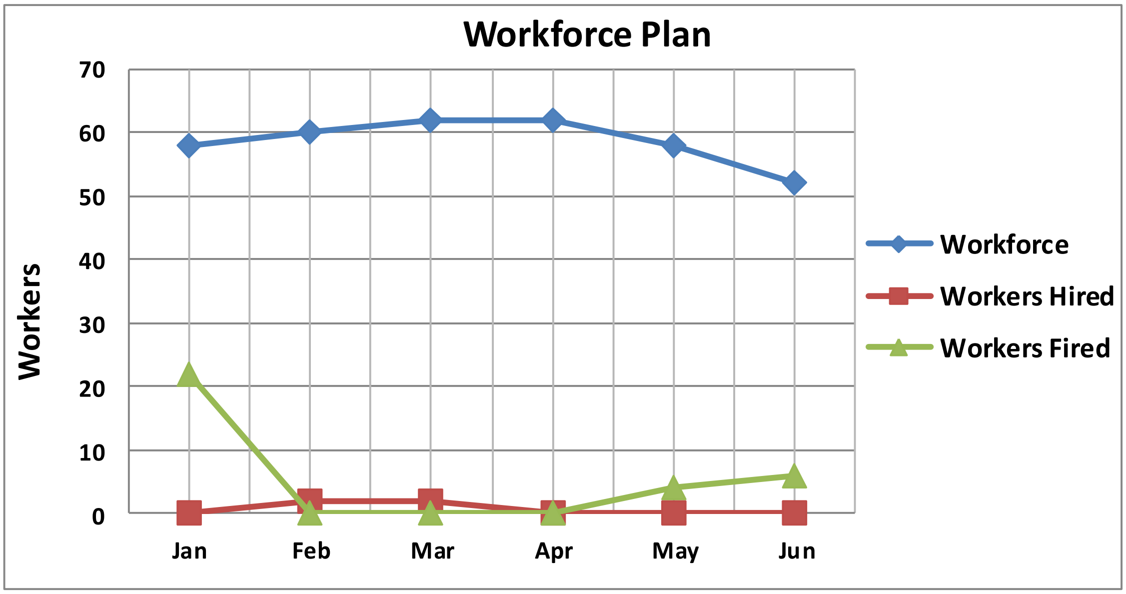


e) Workforce Plan

Figure 4-5: Best Customer Service Plan (Agile plan)

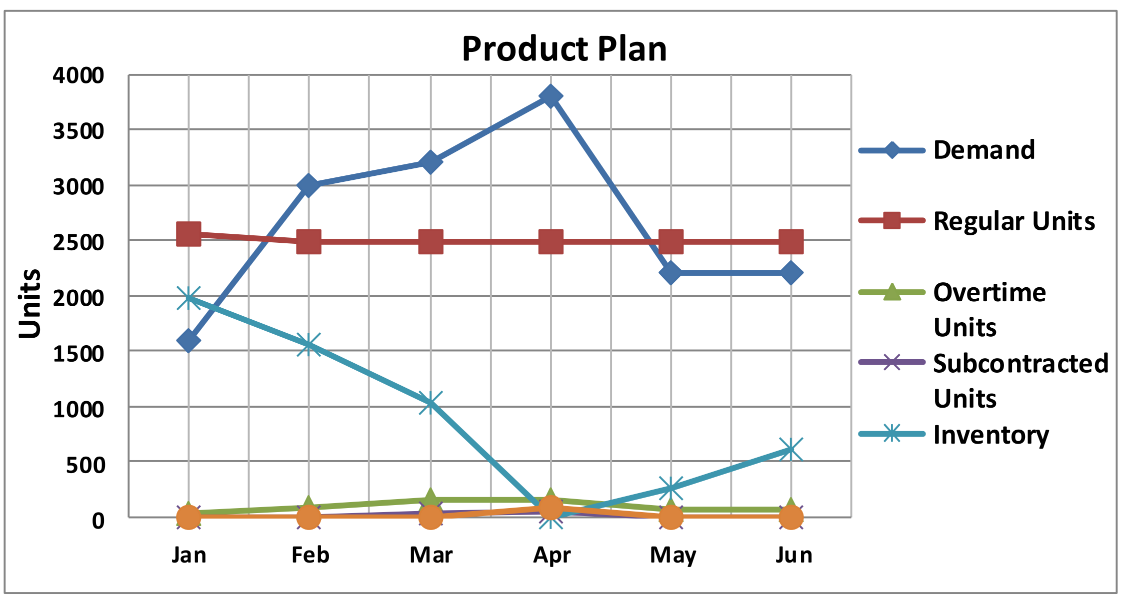


a) Product Plan

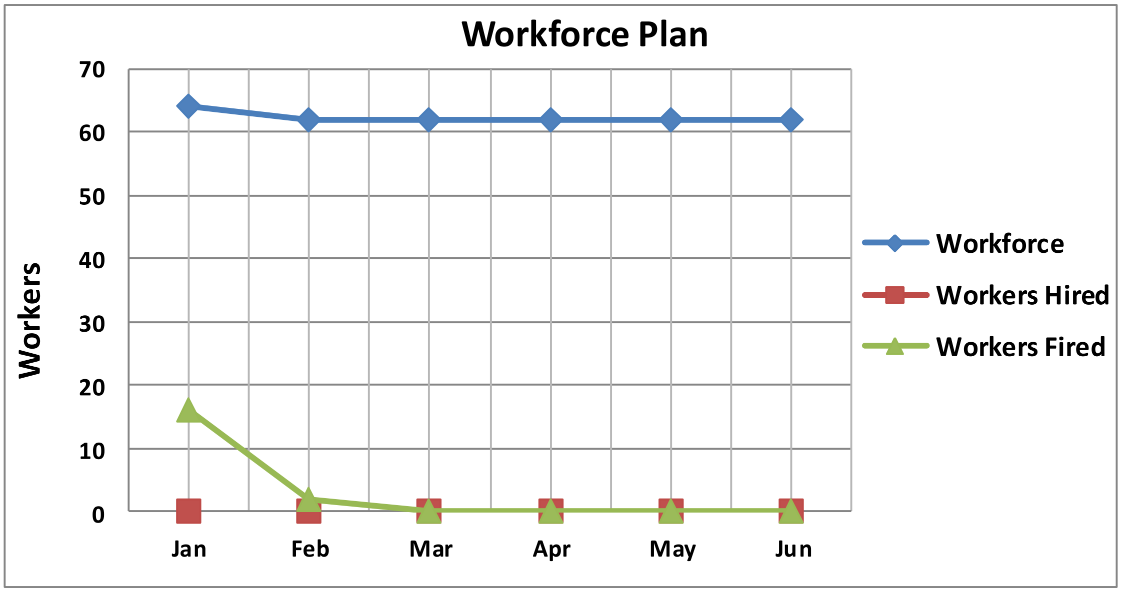


b) Workforce Plan

Figure 4-6: Best Inventory Plan (Lean Practice & financial posture)

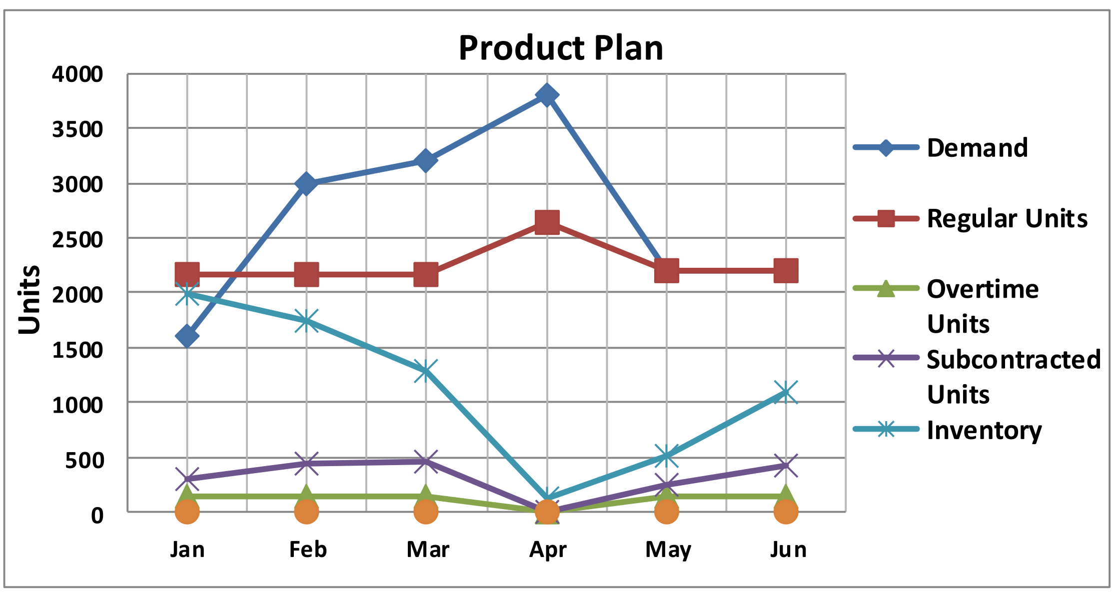


a) Product Plan

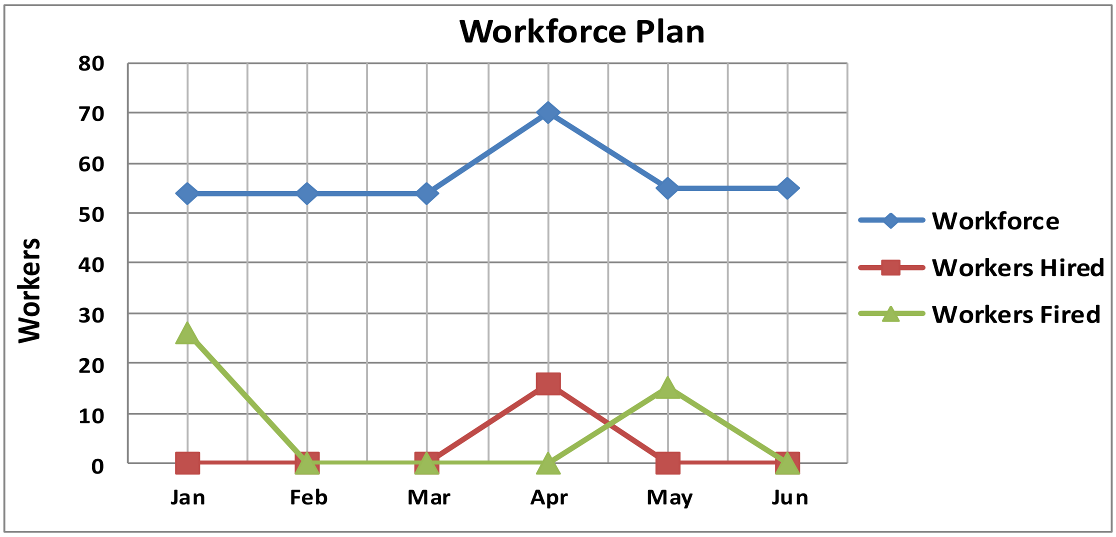


b) Workforce Plan

Figure 4-7: Best financials plan (production costs)



a) Product Plan



b) Workforce Plan

Figure 4-8: Best workforce profile plan (System stability & employee morale)

Table 4-3: Pareto front members

	Back Orders [MIN]	Inventory [MIN]	Total Costs [MIN]	WF Profile [MIN]
1	0	5836	448606	3.1220
2	355	4802	443207	3.3862
3	95	5428	431059	1.0498
4	0	6758	472782	0.1063
5	94	5417	434146	3.0498
6	61	5510	435035	3.0498
7	317	4893	437159	3.0183
8	209	5124	439687	3.0183
9	146	5233	440906	4.4177
10	20	5894	441654	2.0183
11	329	4891	443873	4.4177
12	344	4837	445174	3.7541
13	2	5908	445954	1.0498
14	11	5871	447279	4.0183
15	6	5928	450064	0.7541
16	0	6042	455318	0.7425
17	0	6137	457438	0.5100
18	18	5801	458862	2.5100
19	29	6120	467103	0.4127
20	0	6559	469644	0.1919

Table 4-3 shows the Pareto front solution members and Figure 4-5 to Figure 4-8 show the extreme points solutions, which correspond to the first four members of Table 4-1 respectively. Every solution represents an APP plan that is decomposed into product side and resource (workforce) side respectively. Regardless of the archive size, the Pareto front always keeps the best extreme points and a well-diversified set of trade-off solutions. Every one of these solutions shows how the concept of multi-facetted balance is illustrated at the objectives level. The best customer service level reflects best agile practice, while low inventories solutions reflect lean production and emphasize good financial posture (i.e. better liquidity ratios – financial strength measure). The workforce profile is optimized to make no-change or abrupt changes desirable. The preferred business practice is to make major infrequent changes in the workforce rather than phasing out the changes slowly. Achieving the best balance among all these competing objectives provides a real advantage. These results would help the managers in charge to make informed decisions as to the best plan to follow under given circumstances.

The mentioned example shows a snap shot of an APP progressive model. The cost function is linear and the generated best cost solution (431,059) differs only from the optimum solution (423,659) mentioned in (Chopra and Meindl 2007) by less than 2%, which shows the accuracy of the solution algorithm in optimizing the original single objective problem in addition to optimizing the other non-financial objectives (see Table 4-3).

The power of progressive modeling comes from its ability to provide a very good set of model development options. The hiring and firing costs could be nonlinear; the model can be updated easily by removing the resources component and replacing it with an updated one. The back orders are added to cost function; however, in practice it is very difficult to quantify the cost of lost sales. Minimizing the backorders as one of the multi-objectives solves this problem. The GAs solver can be replaced and updated with any other optimization technique such as Tabu search and simulated annealing. In addition, some operators can be added or removed very easily. Products data also necessitates an update of that model component. If more constraints are to be added, then the modeller structure itself should be updated, and so on.

4.6 Summary

In this chapter, the first initiative to present Progressive Modeling was introduced. Addressing industrial problems from the systems perspective and developing problem models that easy to adapt and grow, i.e. change ready technically, was the early goal. In the next chapter, the progressive modeling process will be redefined in a much formal way and the solution algorithms will be the next main target to make them change ready too. The next application will discuss the multi-objective multi-product aggregate production problem (MMAPP) with set-up decisions.

Chapter 5 PROGRESSIVE MODELING II AND THE MMAPP PROBLEM

5.1 Introduction: PMII and the Industrial-Academic Gap

In industry, several tools and practices have evolved to strengthen the core competency of manufacturing firms. Most developed applications counts a lot on simple algorithms and best industrial practices. Even though mathematical modeling epitomizes the sophistication of the academic world, most managers and practitioners find them hard to grasp and implement. The parties from both worlds seem isolated and do not learn from each other. In the academic field, manufacturing planning and control problems are usually discussed from a problem perspective: assumptions, mathematical models, and solution methodologies/algorithms or more specifically the classical operations research approach. Managers and operations engineers do not like to deal with such esoteric models and find them lacking applicability; software packages are preferred. Unfortunately, the implemented algorithms in these packages are very generic and should be very trusted due to the public image and legal liabilities. Utilizing generic solutions has two drawbacks or consequences: first, it undermines our ability to create efficient solutions; second, it sidetracks our attention from creating a distinguished competitive advantage, which should lay down hard barriers to competition. Software packages are usually black boxes with no control on their inner workings and their development; however, they enjoy a great power of pragmatism and addressing real problems. So many great lessons could be learned from the software development technologies especially in terms of how we analyse problems and create evolving solutions. The sophisticated models of the academia should be presented also in a way that makes them more appealing to industry and more applicable.

In this chapter, Progressive Modeling (PM) is redefined as an integrated multi-disciplinary forward-looking problem analysis, modelling, and solution approach. The vision is to lessen the gap between industrial and academic worlds by creating sophisticated yet simple and pragmatic solutions. Unlike chapter 4 that briefly described the PM process, the process in this chapter is formalized and described in a separate context from the application problem.

As an illustrative application, the multi-objective multi-product aggregate production-planning problem will be discussed. The problem is presented as a compilation of several interacting components with well-defined responsibilities. A mathematical model that incarnates some new principles of PM approach is presented. A set of solution algorithms are compiled and enriched with some innovative thoughts in order to add flexibility and creates feasible solutions at the outset are also presented. Finally, an illustrative numerical example and its results are discussed. The generated solutions show how decision makers can capitalize on several options and abide by best industrial practices.

This chapter is organized as follows: the formalized progressive modeling process is described first. The multi-objective multi-product aggregate production planning is also presented as a case problem. The focus of Progressive Modeling at this stage is developing progressive solution algorithms. System envelop constraints, couplers, incomplete chromosome definitions are some new gadgets that will accompany the MMAPP problem. The chapter ends by a numerical example for results demonstration.

5.2 Progressive Modelling II—the Process

5.2.1 Systemize, Analyze, and Componentize

The first step in Progressive Modelling is to handle problems from system perspective. The problem at hand, sub-system, or component helps to achieve a certain function or goal within a wider system. Demand forecasting, aggregate planning and scheduling are some examples that can be redefined from that perspective. By systemizing problems, they can be looked at as an integrated set of synergistically interacting components with solid definitions of their roles and well-sought objectives. Componentizing problems works greatly when the problem at hand is relatively large and addresses a real system that can be grasped, analysed, and modelled. Systems analysis and design gives a set of formal ways of how to break down and design a well-established system or a group of sub-systems. System requirements, structure, and behaviour can be described in a very formal and expressive way. The Unified Modelling Language (UML) makes the task now much easier by standardizing all related diagrams. UML captures most of the requirements and advancements of most existing software technologies available up to date.

PM starts with analysing the problem at hand and decomposing it into several interacting components. Every component encapsulates a certain part of logic that governs and performs a pre-specified task that adds up the main task or mission of the wider system. Component Based Software Engineering (CBSE), which represents the state of the art of software engineering, helps a lot in that regard. In the context of CBSE, The new technology promotes the separation of concerns of interacting components in a black-box communication fashion. Communications are strictly formalized by a set of protocols called interfaces. These protocols should be kept invariable all over the lifetime of the designed system. Whenever any component needs to be replaced or updated, a new one is to replace the older one provided that it honours the pre-specified protocols. With today's several available technologies, the process is done in a seamless way with a minor effort.

5.2.2 Define the Logic That Governs

In order to be well-understood, controlled, or managed, systems behaviour should be modelled. If this behaviour can be described in a sophisticated way by governing equations, a mathematical model can be defined. Operations research defines decision variables, constraints, and objectives as the building blocks of math models. In PM context, math models are defined in an open-ended forward-looking fashion. Some assumptions like linearity are ignored at the outset; non-linearity is the case. Since problems are turned into systems, they could have beginning and ending states. The beginning state reflects the initial values of system state variables, while the ending state is the target values of these variables. Multiple objectives to be achieved are the mission. Systems cannot be judged by a single criterion; otherwise, they would never last for long. There are several stakeholders: shareholders, customers, suppliers, workers, and may be others; keeping all those parties satisfied leads to long-standing system stability.

The math modelling is enriched with the introduction of what is called function templates. Templates are just function definitions of the governing inputs and outputs relations. The exact definition is considered an ongoing concern. This is very important because problem definition is tied to our level of progressive knowledge. Lack of knowledge or uncertainty of how relations should be defined should not be an obstacle. In software implementation, function templates can be implemented via interfaces. This is called black box modelling. By introducing this notion, math model development is defined. In that regard, math model themselves are a subject of enhancements, which will lead to a better understanding of the underlying systems and promoting their performance in a scientific-like way.

Math models are distributed among several interacting components in a process called model deployment. Some objectives and constraints can be confined into a certain component. Some can be defined by only gathering and comparing information from more than one component. In that case, a controlling, an intermediary, or a brokering component can execute that logic. This has a great impact of twofold: First, it reduces the complexity of existing models and makes them more manageable; second, it enables extensibility of existing models by making them grow as knowledge and information unfolds.

5.2.3 Optimize and Control the Logic: Finding Best Alternatives

Logic controllers or optimizers facilitate its evolvable nature and manages its performance. Generic solvers or optimizers should be available and ready for any change of that logic. Intelligent optimization algorithms such genetic algorithms, tabu search, particle swarm, and others are good candidates for this part. The tenets of these algorithms may be broken if necessary. Mimicking natural phenomena gain some rigidity by trying to abide by their rules even though, fortunately, we have no obligation to honour them. The APP problem solved later exemplifies this approach. Progressive Modelling is an integrated solution approach that synergizes several technologies, disciplines, and algorithms in order to create more practical and novel solutions as an ultimate goal.

5.3 Application: Multi-Objective Multi-Product Aggregate Production Planning (MMAPP) Problem

Due to the lack of agreement among academicians of how the problem should be formulated, the APP problem was classified as an ill-formulated problem by Ismail and ElMaraghy (2009). Although aggregate production planning is a system wide problem, the vast majority of models developed to address the problem focused on it as a single objective problem. Baykasoglu (2001) stated that this might be reasoned to the difficulty of solving the multiple decision making problems. The multi-product aggregate production problem has been well studied in the literature (Silver 1972; Hax and Candea 1984; Silver and Peterson 1985; Mazzola, Neebe et al. 1998). Mazzola (1998) illustrated that the complexity of the mutli-product APP problem makes it strongly NP-hard problem. Adding the multi-objective aspect and addressing it in an open-ended manner—progressively modelled—makes it much harder. The remaining part of this chapter elaborates on how the MMAPP is addressed from the PM perspective.

5.4 Model Components

In order to address the MMAPP model complexity and enable its evolvability, the model is arranged into several interacting components. The machinery and workforce components are counter balanced by product component via the APP modeller. The modeller is the central component where the logic related to the problem at hand is defined; it also controls the inter-relations among the interacting components and utilizes the optimizer to find an optimized and compromised set of solutions. The whole process is guided and the final solution is presented via the system user interface. Eventually and just before discussing the numerical example, the inner workings of these components would be described after introducing the suggested model and solution algorithm details.

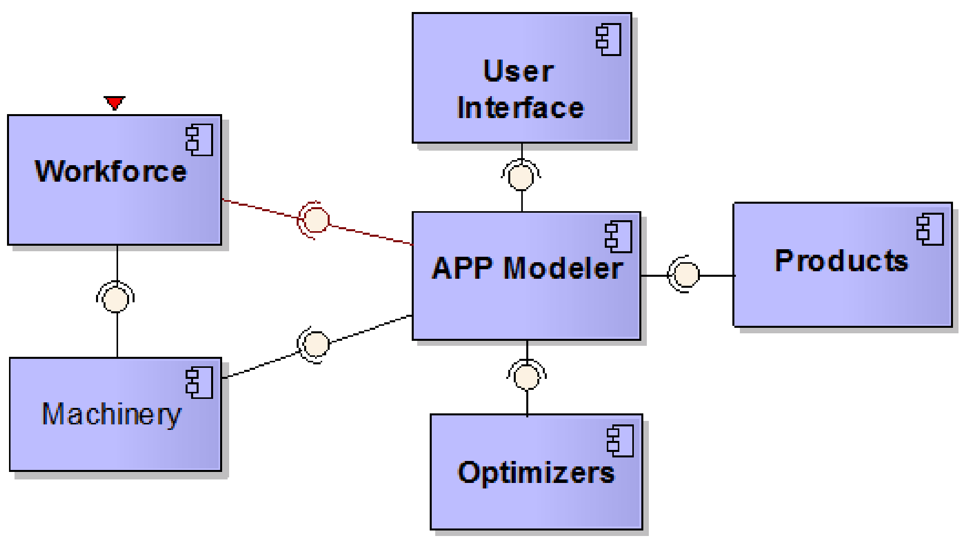


Figure 5-1: MMAPP Component diagram

5.5 Mathematical Model

In the suggested MMAPP Model, the decision variables are different product supply and workforce levels. Every product supply could be split into regular, overtime, and outsourced. Any other variable, such as inventory or backorder of a certain product, either is given as an input or can be evaluated as an output or as a dependent one.

The model starts by defining objective templates, equations (5.1) to (5.4). Equations (5.5) to (5.8) represent some objective implementations. The objectives considered include optimize financial measures, optimize workforce profile, minimize capital investment in inventory, and minimize backorders.

System constraints are divided into several related sets: workforce-machinery or system resources balance (equations/inequalities (5.10) to (5.13)), product balance (equations/inequalities (5.14) to (5.15), and System envelop constraints (5.16) to (5.20). In addition, there are setup constraints (5.21) and non-negativity/integer constraints represented by inequalities(5.22).

Equation(s) (5.10) & (5.11) maintain work force variables in balance; equation(s) (5.11) ensure that workforce is below or at its threshold; inequalities (5.12) and (5.13) reflects resource consumptions balance. Inequality (5.12) ensures the production time consumed during setup or active regular time production is lower than the available total workforce provided production time. Inequality (5.13) checks that the production time consumed within overtime periods does not exceed the allocated time allowed by overtime margins.

Equation (5.14) & (5.15) keeps inventory, total quantities supplied and demand in balance. Equations (5.16) to (5.20) are called system envelop constraints: Constraint 16 turns the production system throughout the planning horizon as a black box: Initial state, target state at the end of planning period, total product supply of each product and total product demand. These constraints not only turn the APP problem into APP system but also reduce the complexity and boost the performance of the solution algorithm as will be elaborated later. Finally, the last constraints (5.22) limit all decision variables to have both non-negative and integer values.

Notation

T number of planning periods

N number of products, product families, or product groups

D_{it} demand of product i in period t (units)

I_{it} inventory level of product i at the end of period t (units)

B_{it} backorder level of product i at the end of period t (units)

R_{it} regular time production volume of product i in period t (units)

O_{it} overtime production volume of product i in period t (units)

S_{it} subcontracted volume of product i in period t (units)

H_t workers hired in period t (man-day)

F_t workers fired in period t (man-day)

W_t work force level in period t (man-day)

W_{it} work force proportion allocated to product i in period t (man-day)

t_{li} labor time to produce one unit of product i (*hours*)

$t_{\sigma i}$ setup time of product i (*hours*)

C_{mi} *material* cost of product i (\$ / uint)

C_w labor cost in period t (\$/man-hour)

C_o overtime labor cost in period t (\$/man-hour)

C_{hi} holding cost per unit of product i (\$/uint)

$C_{\sigma i}$ set-up cost for product i in period t (\$/uint)

C_{si} subcontracting cost of product i (\$/uint)

C_{bi} backorder cost of product i (\$ / unit)

C_F firing cost per worker in period t (\$ / worker)

C_H hiring cost per worker in period t (\$ / worker)

p_i price per unit of product i (\$/unit)

ψ_i Set-up decision variable of product i

Δw prefered incremental workforce change (man-day)

$W_{t_{max}}$ maximum work force available in period t (man-day)

β_t fraction of regular work force available for overtime use in period t

δ regular time per worker (man-hour/man-day)

Initial values

Decision variables:

W_{it}, O_{it}, S_{it}

Dependent Variables:

$\psi_i, I_{it}, B_{it}, R_{it}, W_t$

Objectives

Templates

$$\begin{aligned} Min\,/\,Max\ Z_1 \quad = \quad & g_1(R_{it}, O_{it}, S_{it}, I_{it}, B_{it}, p_{it}) \\ & + g_2(W_t, H_t, F_t) \\ & + g_3(M_t, \psi(R_{it}), \psi(O_{it})) \end{aligned} \tag{5.1}$$

$$Min\ Z_2 = g_2(W_t, H_t, F_t) \tag{5.2}$$

$$Min\ Z_3 = g_3(I_{it}, C_{Ii}) \tag{5.3}$$

$$Min\ Z_4 = g_4(B_{it}, p_i) \tag{5.4}$$

:

Sample Implementation

$$Max\ Z_1 = \sum_{t=1}^{T}\sum_{i=1}^{N}\left\langle \underset{\text{Revenue}}{p_{it}Min(D_{it},P_{it})} - \begin{pmatrix} \left[C_o(t_{l_i}O_{it}) + C_{s_i}S_{it} + C_{h_i}I_{it}\right] + C_{mi}(R_{it} + O_{it}) \\ \text{Product costs} \\ +\left\langle C_w W_t + C_F F_t + C_H H_t \right\rangle \\ \text{Workforce } costs \\ +\sum_{i}^{N} C_{\sigma_i}\psi_i(R_{it}) \\ \text{Machinery } costs \end{pmatrix} \right\rangle$$

Or **(5.5)**

$$Min\ Z_1 = \sum_{t=1}^{T}\sum_{i=1}^{N}\begin{pmatrix} \left[C_o(t_{l_i}O_{it}) + C_{s_i}S_{it} + C_{h_i}I_{it}\right] + C_{mi}(R_{it} + O_{it}) \\ \text{Product costs} \\ +\left\langle C_w W_t + C_F F_t + C_H H_t \right\rangle \\ \text{Workforce } costs \\ +\sum_{i}^{N} C_{\sigma_i}\psi_i(R_{it}) \\ \text{Machinery } costs \end{pmatrix}$$

Minimize changes in workforce

$$Min\ Z_2 = \sum e^{-\frac{(Ht+Ft)}{\Delta w}} \qquad \textbf{(5.6)}$$

Minimize capital investment in inventory

$$Min\, Z_3 = \frac{1}{T}\left\{\sum_i\sum_t p_i \max\left\{I_{it}, 0\right\}\right\} \qquad \textbf{(5.7)}$$

Minimize backorders:

$$Min\, Z_4 = \frac{1}{T}\left\{\sum_i\sum_t C_{bi} \max\left\{-I_{it}, 0\right\}\right\} \qquad \textbf{(5.8)}$$

: (5.9)

Constraints

Work force Balance

$$W_t = W_{t-1} + H_t - F_t \qquad \forall\, bucket\, t \qquad (5.10)$$

$$W_t \le W_{t\max} \qquad \forall\, bucket\, t \qquad (5.11)$$

Resource Consumption Balance

$$\sum_{i=1}^{N} t_{\sigma i}\psi(R_{it}) + \sum_{i} t_{l_i} R_{it} \le \delta W_t \qquad \forall\, Product\, i\, \&\, \forall\, bucket\, t \qquad (5.12)$$

$$\sum_{i} t_{l_i} O_{it} \le \delta \beta_t W_t \qquad \forall\, Product\, i\, \&\, \forall\, bucket\, t \qquad (5.13)$$

Product Balance

$$I_{it} - B_{it} + D_{it} = I_{it-1} - B_{it-1} + P_{it} \qquad \forall\, Product\, i\, \&\, \forall\, bucket\, t \qquad (5.14)$$

$$P_{it} = R_{it} + O_{it} + S_{it} \qquad \forall\, Product\, i\, \&\, \forall\, bucket\, t \qquad (5.15)$$

System Envelop Constraints

$$\sum_{t=1}^{t=T} P_{it} + \{I_{i0} - I_{iT}\} + \{B_{iT} - B_{i0}\} = \sum_{t=1}^{t=T} D_{it} \qquad \forall\, Product\, i \qquad (5.16)$$

$$B_{i0} = A_{1i} \qquad \forall\, Product\, i \qquad (5.17)$$

$$I_{i0} = A_{2i} \qquad \forall\, Product\, i \qquad (5.18)$$

$$B_{iT} = \mathfrak{M}_{1i} \qquad \forall\, Product\, i \qquad (5.19)$$

$$I_{iT} = \mathfrak{M}_{2i} \qquad \forall\, Product\, i \qquad (5.20)$$

Set up constraints

$$\psi_{it} = \begin{cases} 1 & \text{if } \mathrm{R_{it}} > 0 \\ 0 & \text{if } \mathrm{R_{it}} = 0 \end{cases} \tag{5.21}$$

Other constraints

$$R_{it}, O_{it}, S_{it}, W_{it}, D_{it} \geq 0 \text{ and an integer} \tag{5.22}$$

5.6 Solution Approach

5.6.1 The Algorithm—Brief Introduction

The multi-objective multi-product aggregate production problem is a relatively complex problem with an increasing number of constraints and decision variables. The simplest problem with a single cost objective is an NP-hard problem. A novel solution algorithm that has its roots in genetic algorithms and evolutionary multi-objective optimization is introduced in this chapter. Some of the basic tenets of GAs are broken in order to address the problem complexity and much more importantly to overcome the ever-growing number of constraints. As a solution methodology for APP problem, GAs literature considers neither the multi-objective aspect nor the multi-product problem; this most probably is due to the tremendous amount of data needed and the large number of constraints involved as has been shown earlier. In this section, several issues related to the solution algorithm are presented and then the whole algorithm summarized to show how all parts can add up to develop a relatively large scale compiled solution algorithms.

5.6.2 Problem Coding and Incomplete Chromosomes

As shown in Figure 5-2, a chromosome is list of tuples that equals to the number of planning buckets. Every tuple is an array of available workforce in terms of man-periods proportions assigned to each product. In reality, the system is setup for a certain product k and all the available workforce are to be engaged in the manufacturing of that product a proportion of time equals to $w_k / \sum_{1}^{N} w_i$.

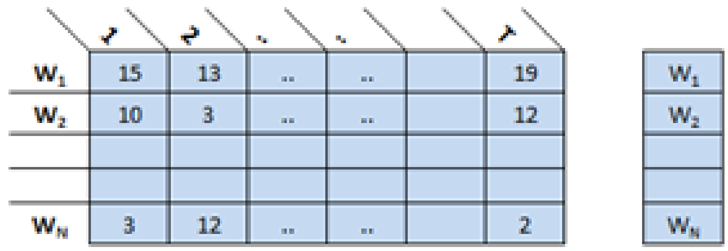


Figure 5-2: Multi product APP chromosome

Although a decoded version of a chromosome can generate a complete workforce plan or schedule, it is not enough to generate complete individual product plans. In the traditional genetic algorithms, usually every chromosome, or genotype, is decoded into its counterpart decision point or solution, i.e. phenotype. By removing the boundaries between the genospace, where the chromosomes are located, and the phenospace, where meaningful solutions are being mapped (our APP plans in that case), a basic tenet of Genetic Algorithms was broken to generate semi-coupled decision variables, i.e. A coded solution cannot be mapped to a complete plan and vice versa. The reason why only partial solutions are encoded as described earlier cannot be described without discussing the new utilization of constraints as incomplete coupling mechanisms. A workforce plan and system constraints can give so many clues to generate product plans but the information content are still insufficient to generate complete plans. This leads us to discuss constraints from a new perspective and introducing the algorithmic couplers.

5.6.3 Constraints: Coupling Mechanisms and Searching Guards

Taking the problem at hand as an example, constraints represents coupling links among decision variables and they work as a multi-dimensional envelops that surround their values. The more constraints we have the harder the mission to find a feasible solution, i.e. constraint-added complexity. With a problem that encompasses an exploding number of constraints like the APP problem, using penalty functions and repairing mechanisms cannot be a solution approach of choice. Instead of being a source of trouble, constraints hold a lot of information if utilized efficiently the dynamics of searching for a feasible solutions can be greatly improved. The premise is that in order to make the constraints guide the solution process, and while searching for better alternative, the hopping process, moving from a feasible point to another feasible point, should happen in the feasible domain. In this chapter, we have jumped many steps ahead by developing system envelop constraints which make use of the initial and desired end state information as described before. Using decoded workforce plans, system constraints, and couplers, which is described very shortly, and continuing to move through only feasible pathways are the guiding foundations that are taken into consideration throughout the solution algorithm journey that starts with the initialization algorithm.

5.6.4 The Initialization Algorithm

Starting from the point that the system is bounded by its available maximum workforce level $\{W_t \leq W_{t\max} \qquad t = 1......T\}$, the total workforce power (man-period) can be initiated easily. How the planning bucket is sliced to produce all or some of N products creates the role of the *initialization coupler*. Simply put, the initialization coupler is a micro-heuristic (typically, a helping function, or set of functions) that can be hooked to the initialization algorithm. An algorithm lifetime is now extended through the couplers notion: the workforce power can be sliced randomly across products, or can be made proportional to individual product demand, or using any other proportioning criterion. Delegating the workforce proportionating process to a coupler makes the definition of the initialization algorithm progressive, open for development, and keep the other parts of initialization algorithm invariable with time.

::::::::::::::::::::::::::::::::::::::Geno-Space:::::::::::::::::::::::::::::::::::::

	1	2	3	4
W1	19	19	16	5
W2	4	4	7	11

::::::::::::::::::::::::::::::::::::::Pheno-Space:::::::::::::::::::::::::::::::::::::

Workforce Plan

	P. State	1	2	3	4
Wmax		24	24	24	24
W	20	23	23	23	16
H		3	0	0	0
F		0	0	0	7

Setup Sequencing Plan

	P. State	1	2	3	4
Slot#1	A	A	B	A	B
Slot#2		B	A	B	A

Product#1 Plan

	P. State	1	2	3	4
D		3500	6000	4000	1300
Rmax		3800	3699	3200	973
R		3800	3699	3200	973
Omax		633	633	533	167
O		633	633	533	167
Smax		2000	2000	2000	2000
S		393	674	449	146
Imax		NA	NA	NA	NA
I	0	826	0	14	0
Bmax		NA	NA	NA	NA
B	500	0	168	0	0
W		19	19	16	5
SetUp		No	Yes	No	Yes

Product#2 Plan

	P. State	1	2	3	4
D		1200	2000	2800	3600
Rmax		1200	1152	3300	2016
R		1200	1152	3300	2016
Omax		200	200	550	350
O		200	200	550	350
Smax		2000	2000	2000	2000
S		31	51	72	92
Imax		NA	NA	NA	NA
I	500	731	134	1256	114
Bmax		NA	NA	NA	NA
B	0	0	0	0	0
W		4	4	11	7
SetUp		No	Yes	No	Yes

::::::::::::::::::::::::::::::::::::::Objective-Space:::::::::::::::::::::::::::::::::::::

Objectives

	Directive	Name	Value
1	Min	Financials	8931199
2	Min	Backorders	2735.1
3	Min	Inventory	50400
4	Min	Workforce	2.00012

Figure 5-3: Genomes, Plans, and Objectives

5.6.5 Decoding

5.6.5.1 Step 1: Develop Set-up Sequencing Plans

Using the workforce proportion allocated in the genome, product-sequencing plans are developed as depicted in Figure 5-3. If the workforce power proportion is greater than zero, a setup process may be necessary. If that product is already set up at the end of a planning bucket, it should not undergo a setup process at the beginning of the next period. Based on the last premise, a micro-heuristic is developed to identify the setup cycles needed. For a case of a 2-product problem, as shown in Figure 5-3, almost one product may need a set-up per planning bucket. Nevertheless, some idiosyncrasies could happen: for once, one product could occupy a completely planning bucket; for another, a product could occupy the last slot at the end of planning bucket and it might be terminated at the next; just out of a neighbourhood search operation (mutation if the dialect is genetic algorithms). The micro-heuristic algorithm takes care of these idiosyncrasies. The generated setup sequencing plans are all feasible and the incurred costs are only estimated if the set-up process actually happens. This corrects a common mistake in modelling APP problems in textbooks and some research papers, which considers the setup costs, incurred blindly whenever the production volume of a certain product is above zero. Constraint (5.21) should be rewritten as follows.

$$\psi_{it} = \begin{cases} 1 & \text{if } \mathrm{R_{it}} > 0 \text{ and the prodcut set-up can't be saved by reshuffling} \\ 0 & \text{otherwise} \end{cases} \quad \textbf{(5.23)}$$

In addition to set-up cost saving, the time that is mistakenly allocated for setup is used in active production reducing more costs.

5.6.5.2 Step 2: Update Workforce Plan & Regular Workforce Output

Decoding the incomplete chromosome into a workforce plan is straightforward. The workforce power proportions are added up for every tuple to evaluate W_t. Having W_0 known from the beginning and utilizing constraint(s)(5.10), the hiring and firing data is updated consequently. The regular workforce quantities can be updated using constraint(s)(5.12). Since the hiring and firing are allowed, all the generated workforce power is transformed into individual product units. Using constraint(s)(5.12), the maximum overtime allowed can be calculated.

5.6.5.3 Step 3: Update Overtime Output

Deciding on the amount that should be produced during the overtime-period can be evaluated using system envelop constraints (5.16) to (5.20). First, the individual total product volume required is evaluated using the soft constraints (5.16) to (5.20). For every product k, its total product supply required $\Delta_{k_{\text{Required}}}$ is calculated according to equation(5.24). Since the man-period allocated for each product is already defined from the incomplete chromosome definition and by consulting constraint(5.13), the max overtime production can be calculated for every planning bucket. As a result, the total maximum overtime output can be calculated by equation(5.25). After doing these calculations: The following logic is executed.

$$\Delta_{k_{\text{Required}}} = \sum_{t=1}^{t=T} D_{kt} - \sum_{t=1}^{t=T} R_{kt} + \{I_{k0} - I_{kT}\} + \{B_{kT} - B_{k0}\} \tag{5.24}$$

$$Sum_{omax} = \sum_{t=1}^{t=T} O_{k\max} \tag{5.25}$$

```
If
```
$\Delta_{k_{\text{Required}}} \leq 0$

```
   Mark plan state as regular
   Skip to step 5
```

```
If
```
$\Delta_{k_{\text{Required}}} \geq Sum_{omax}$

```
   Set all
```
$O_{kt} = O_{k\max} \ \forall t$

```
   Skip to step 4
```

```
If
```
$\Delta_{k_{\text{Required}}} \leq Sum_{omax}$

Similar to initialization coupler, an overtime coupler can be linked to profile the overtime distribution curve.

Mark plan state as overtime

Skip to step 5

Product#1 Plan	1	2	3	4
Demand	3500	6000	4000	1300
Rmax	3800	2725	2400	0
R	3800	2725	2400	0
Omax	633	467	400	0
O	633	467	400	0
S	473	811	541	176
Supply	4906	4003	3341	176
I	906	0	0	0
B	0	1091	1750	2874

Figure 5-4: Updating Overtime buckets

5.6.5.4 Step 4: Update Subcontracting Quantities

Again, the remaining total product supply required $\Delta^2_{k_{\text{Required}}}$ is recalculated as described by equation(5.26). The total maximum overtime output can be calculated as described earlier by equation(5.25).

$$\Delta^2_{k_{\text{Required}}} = \sum_{t=1}^{t=T} D_{kt} - \sum_{t=1}^{t=T} R_{kt} - \sum_{t=1}^{t=T} O_{kt} + \{I_{k0} - I_{kT}\} + \{B_{kT} - B_{k0}\} \tag{5.26}$$

After doing these calculations: The following logic is executed.

If $\Delta^2_{k_{\text{Required}}} \geq S_{max}$

Use subcontracting coupler to profile S_{max} throughout the planning buckets

Else

```
Use subcontracting coupler to profile $\Delta^2_{k\ \mathrm{Required}}$ throughout the
planning buckets
```

Product#1 Plan				
	1	2	3	4
Demand	3500	6000	4000	1300
Rmax	3800	2725	2400	0
R	3800	2725	2400	0
Omax	633	467	400	0
O	633	467	400	0
S	473	811	541	176
Supply	4906	4003	3341	176
I	906	0	0	0
B	0	1091	1750	2874

Figure 5-5: Updating sub-contracting

5.6.5.5 Step 5: Update Inventory and back-orders

The last stage of decoding is to evaluate inventory or back orders. Using constraint (5.14), the difference between product demand D_{kt} and product supply P_{kt} is evaluated for every product.

For every planning bucket

$$\Delta_{kt} = P_{kt} - D_{kt} \qquad \textbf{(5.27)}$$

If $\Delta_{kt} \geq 0$

$$I_{kt} = \Delta_{kt}$$

Else

$$B_{kt} = -\Delta_{kt}$$

5.3.4.5 Plans or State Machines:

According to the developed decoding algorithm and after noticing the work of couplers and how they are linked as add-ons or micro-heuristics to the solution algorithm, every generated product plan could have one of these different states: regular, overtime, subcontracting, under inventory i.e. below targeted inventory, and satisfactory. It is important to assert that the created solutions satisfy the hard constraints and may or may not satisfy soft constraints {(5.16), (5.19) and (5.20)}. Every product plan could be described as a finite state machine (Brookshea 1989). Finite state machines are imported from automata programming that captures objects behaviour as entities that moves from one well-definite state to another according to certain conditions. When all states can be identified a priori, the set of its states are called finite state machine. As already described in the decoding algorithm, several possible states can be identified for an APP plan as shown in Figure 5-6. At some states, a plan can identify a certain locale around which better solutions can be obtained.

Figure 5-6: Product Plan Possible States

Utilizing the concept of state machines has created better solutions and reduced the computational effort—several steps of decoding can be saved at once. More elaboration will come when discussing the phenospace operators in the next section.

5.6.6 Exploring the Search Space

In order to explore the search space for new solutions, there are two options: the first is to tweak the incomplete chromosomes to generate newer workforce power curves, as would be described in next sub sections, and decode them into their counter parts plans. The second is to tweak the plans themselves according to their states without touching the workforce power curves. According to the new algorithm presented, recombination could happen in both the genospace and phenospace.

5.6.6.1 Genospace Operators:

5.6.6.1.1 Cross Over

Single Point Cross Over

In a single point cross over, a random point in the range of [1, Number of Planning buckets-1] is selected and set it to n. The first n tuples of the first parent are swapped with the first n tuples of the second as shown in Figure 5-7. The generated children are feasible provided that the mating parents are already feasible.

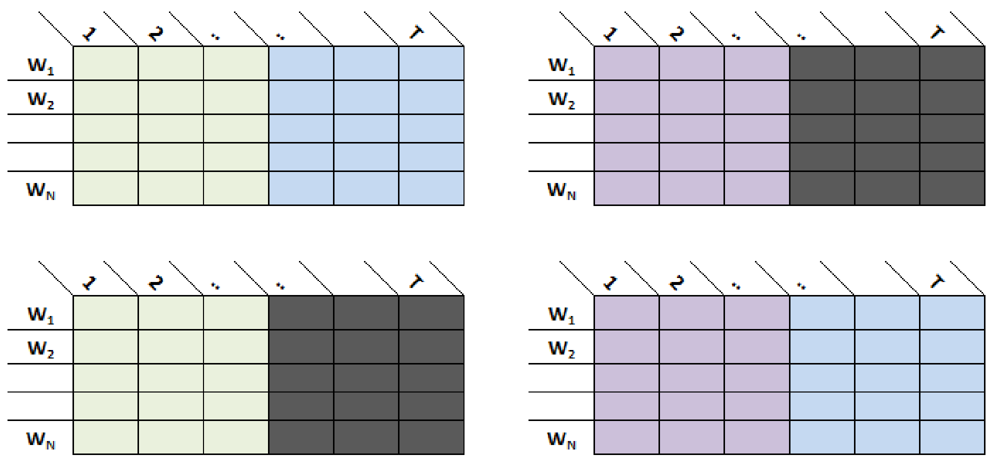


Figure 5-7: Single Point Cross Over

Product Production Time Proportionates Cross Over

During the initialization algorithm the workforce power available to all products are generated randomly. In order to distribute this total among different products, a coupler is used to do the job. In that operator, the generated proportionating curve is swapped between the mating parents. Every tuple total work force is maintained as shown by Figure 5-8.

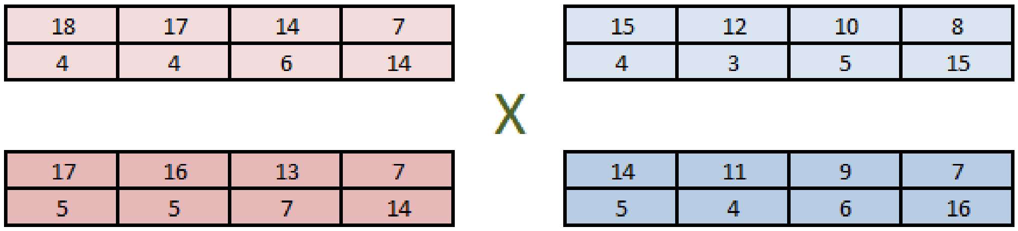


Figure 5-8: Product production time proportionates cross over

Workforce Power Arithmetic Cross Over

The total workforce at each tuple is crossed arithmetically according to the following relations:

$$W_{child1} = \lambda W_{parent1} + (1-\lambda)W_{parent2}$$
$$W_{child2} = (1-\lambda)W_{parent1} + \lambda W_{parent2}$$
$$,\ 0 < \lambda < 1$$

The workforce power is redistributed by maintaining the original proportionating relations. Again, provided that the original parents are feasible, the children are feasible too.

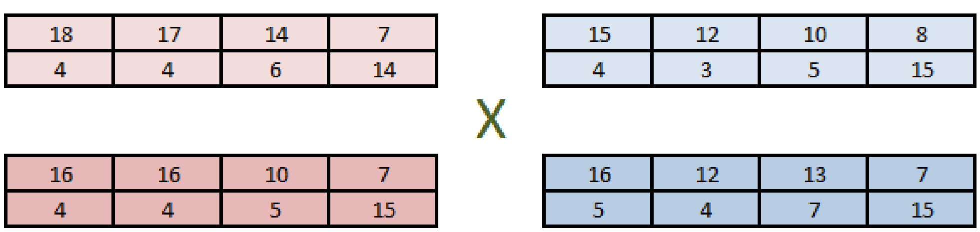


Figure 5-9: Workforce Power Arithmetic Cross Over

5.6.6.1.2 Mutation

Swap Work force Across Products

In this operator, a certain tuple is selected randomly and a couple of its workforce proportions are swapped as illustrated in Figure 5-10.

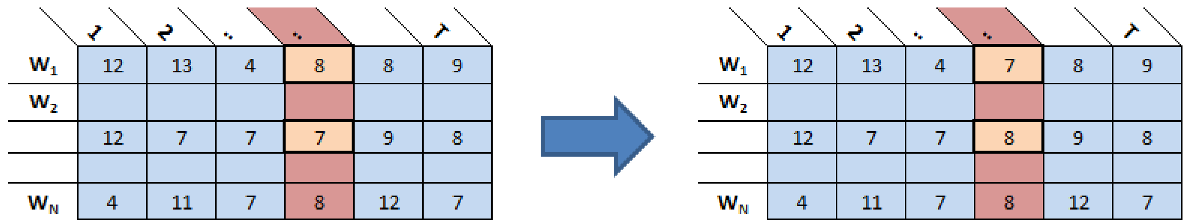


Figure 5-10: Swap workforce power across products

Products Workforce Power Cannibalization

In that operator, a tuple is chosen randomly; man-period assigned to products (w_i,w_j) are selected and their difference is assumed to be Δ. A certain part of that delta is added to a certain product workforce power assigned and subtracted from the other as illustrated by Figure 5-11.

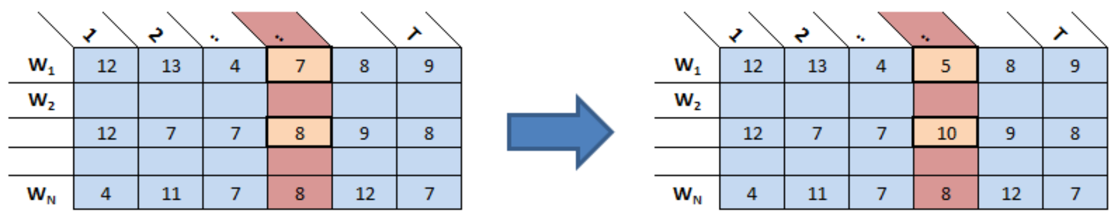


Figure 5-11: Products Workforce Power Cannibalization

Consume/Release Resources

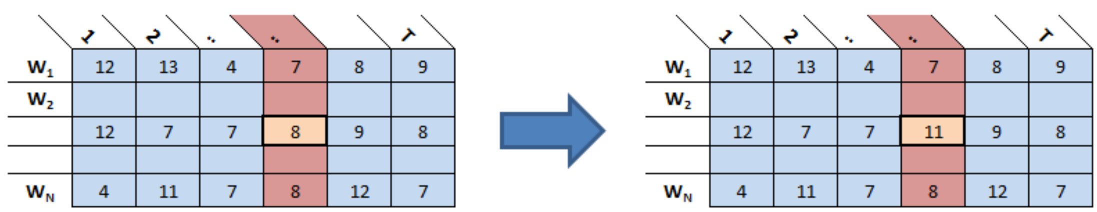


Figure 5-12: Consume/Release Resources

In that operator, a tuple is selected randomly. Within that tuple, the work force power is selected randomly as well. The total work force power is evaluated and the difference between that total and the maximum workforce power is calculated as delta. A certain amount of that delta can be added to workforce proportion w_i. In case we want to release some resource, certain value between (1, w_i) is chosen to be subcontracted.

Insert

The insert operator is a cut and paste operator. A certain tuple is cut randomly and then pasted in another place. The tuples between are shifted from their locations one-step either forward or backward as illustrated in Figure 5-13.

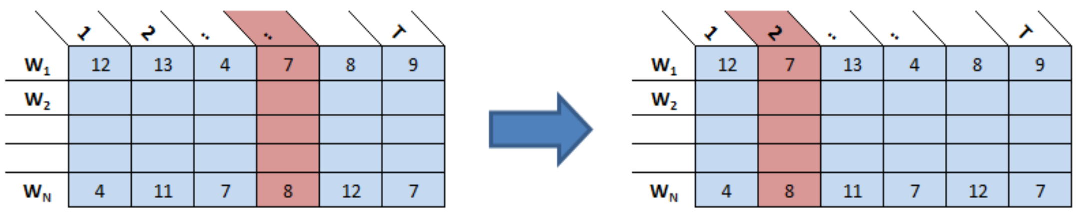


Figure 5-13: Tuple Insert Operator

Inverse

In this operator, a group of tuples (at least two) are selected and their locations are inverted.

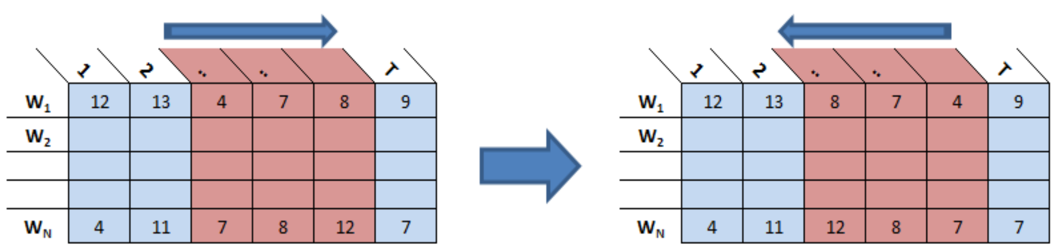


Figure 5-14: Tuple Invert Operator

Swap

In this operator, two tuples are selected randomly (they must have different locations) and their locations are swapped as illustrated in Figure 5-15.

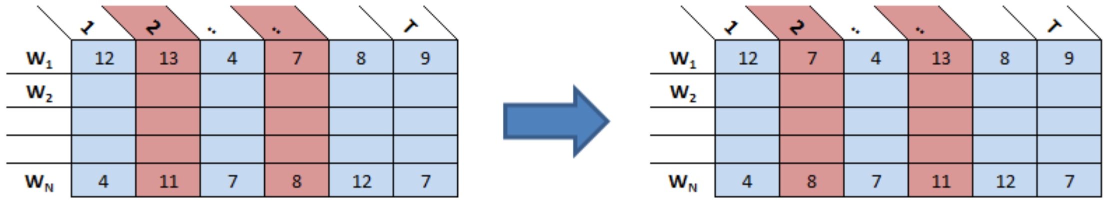


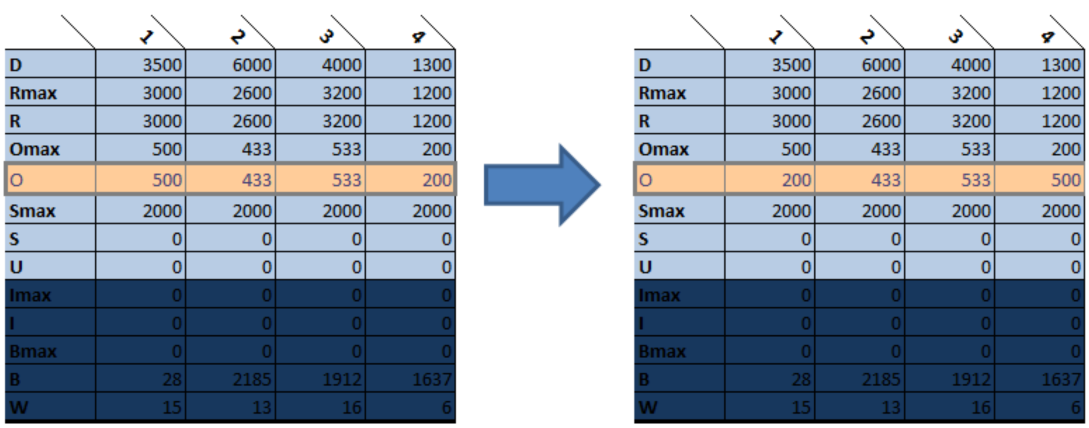
Figure 5-15: Swap Operator

5.6.6.2 Phenospace Operators:

Swap O_t

This operator is applied only if the plan as has the state of "Overtime below Ceiling". Similar to swap operator applied to chromosome tuples, swap O_t is applied to swap the O_t data of two planning buckets provided that each bucket O_t value does not exceed the other's O_{max} value. i.e. [O_{1t} <= O_{2max} and O_{2t} <=O_{1max}]. If this condition is not met, another operator, Delta O_t Operator, is tried; otherwise an infeasible plan would be the outcome, which is never allowed.

The O_t mini-heuristic embedded in its coupler is tweaked here to find a better plan. Two buckets are selected randomly and their associated O_t values are swapped. The outcome is a new plan with the same state. Only step 5 of decoding (Update inventory and backorders are needed). See Figure 5-16 for illustration.



	1	2	3	4
D	3500	6000	4000	1300
Rmax	3000	2600	3200	1200
R	3000	2600	3200	1200
Omax	500	433	533	200
O	500	433	533	200
Smax	2000	2000	2000	2000
S	0	0	0	0
U	0	0	0	0
Imax	0	0	0	0
I	0	0	0	0
Bmax	0	0	0	0
B	28	2185	1912	1637
W	15	13	16	6

	1	2	3	4
D	3500	6000	4000	1300
Rmax	3000	2600	3200	1200
R	3000	2600	3200	1200
Omax	500	433	533	200
O	200	433	533	500
Smax	2000	2000	2000	2000
S	0	0	0	0
U	0	0	0	0
Imax	0	0	0	0
I	0	0	0	0
Bmax	0	0	0	0
B	28	2185	1912	1637
W	15	13	16	6

Figure 5-16: Swap Ot Operator [Place holder only]

Swap S_t

This operator is applied only if the plan as has either state of "Subcontracting below Ceiling" or "Subcontracting Ceiling". Swap S_t is relatively similar to Swap O_t. Since S_{max} represents the maximum outsourced quantity over the planning horizon, the same constraint that governed Swap O_t is not applicable here. If S_t of a couple of planning buckets has different values, they can be swapped creating a new plan with identical state. It could be executed for only plans with only subcontracting state. The outcome is a new plan with a new identical state that can be updated using decoding step 5 similar to O_t.

	1	2	3	4
D	3500	6000	4000	1300
Rmax	3000	2600	3200	1200
R	3000	2600	3200	1200
Omax	500	433	533	200
O	500	433	533	200
Smax	2000	2000	2000	2000
S	0	0	0	0
U	0	0	0	0
Imax	0	0	0	0
I	0	0	0	0
Bmax	0	0	0	0
B	28	2185	1912	1637
W	15	13	16	6

	1	2	3	4
D	3500	6000	4000	1300
Rmax	3000	2600	3200	1200
R	3000	2600	3200	1200
Omax	500	433	533	200
O	200	433	533	500
Smax	2000	2000	2000	2000
S	0	0	0	0
U	0	0	0	0
Imax	0	0	0	0
I	0	0	0	0
Bmax	0	0	0	0
B	28	2185	1912	1637
W	15	13	16	6

Figure 5-17: Swap St Operator

Other Operators

Both O_t and S_t cannibalization, insertion, and inversion operators can be defined and applied. The logic is almost the same of similar earlier versions of genospace operators.

5.6.7 Objective Space and Selection Algorithm

After discussing how the new solutions can be created and generated from the already existing ones, the next stage is to sort out the good solutions from the bad ones. Once all plans are updated and their objectives values are evaluated, they undergo a selection process in order to maintain the best among them and to choose a group of them as parents for regeneration. All genomes, phenomes (set-up, workforce, and product plans), and objectives are encapsulated within individuals. An individual is an alternative that needs to be evaluated against others. Since individuals are multi-objective, mere comparisons can sort them as either dominated or non-dominated ones. In order to find an optimal set, an algorithm is needed to identify their solution quality and which ones should be selected for regeneration. EMO is a research field that is dedicated only for this purpose. SPEA2 (Zitzler, Laummans et al. 2002) takes charge of this part. The list of non-dominated solutions, Pareto front, can dynamically grow and shrink. Sometimes the list size can grow tremendously that can degrade severely the search process. The sorting process in the objective space may be very time consuming. EMO researchers prefer pre-defining the size of this list. In SPEA2, the related parameter is called archive size. SPEA2 archive is updated at every generation; Existing archive individuals with offspring are merged with new individuals, those who come from the recombination process. After sorting out those non-dominated from those that are dominated, the archive size limit is checked: if the number of non-dominated solutions is greater than the archive size, a truncation algorithm is executed to do wise elimination of non-dominated solutions. If that number is less than the archive size, another substitution algorithm is executed to fill the archive with the best of the remaining dominated solutions. A more interested reader can review the details in Zitzler and Thiele (2001) and Appendix A.

5.6.8 Adding Pieces Together: The Algorithm and the Inner Workings of Components

Step1: All the information data that is needed to define the problem at hand and the solution algorithm is compiled. The user interface component is responsible for this part. Instead, some data files can be also utilized or it might be even hard coded, the last option is not a good practice but it can be used during the development stages of progressive models. Once the data is read, it becomes available and ready to be distributed to other components: modeller, workforce, product, and optimizers

Step2: The solution process is sparked by the user via user interface and the modeller starts to trigger the optimization algorithm in the optimizer. The modeller implements an interface called "IGenerator" which takes charge of the initialization and recombination algorithms. Initialization and recombination are population based. The modeller takes care of the encoding, decoding, and evaluation. The modeller attaches itself with its entire internal component to the optimizer, which controls the solution process.

Step3: Once the *Modeler* activates the optimization process, the optimizer takes hold of everything. First, through the "*IGenerator"*, the communication protocol between the modeller and the optimizer, the optimizer asks for the initial population, which is the responsibility of the hooked modeller (internally it is the "Initialize" procedure or operation). Then, it does the selection process and the Pareto-front update, which is an internal issue. Once more, the modeller is asked for newer individuals to join (the modeller uses its internal component *Recombiner* for this purpose). The process is iterated for a certain number of iterations, i.e. stopping criterion. The results obtained are printed on screen or saved for either a later analysis or post processing or both. The developed mathematical model and algorithms are embedded into components and built using the C# language 3.0 and the .NET Framework 3.5.

5.7 Numerical Example

5.8 Problem Data

As a numerical example, a sample problem from Vollman et al (2005) was extended in order to address and illustrate the novel Progressive Modelling approach at work. Table 5-1 to Table 5-3 show the numerical and extended data used. The problem originally defines the demand for the upcoming four quarters. The number of products is set to two: both of them needs set up to be produced. Hiring and firing are allowed and the original objective was limited to the total costs over the planning horizon. The data was extended to include target inventory and back-orders as a final system state. Initial workforce and maximum workforce limit is specified as well. Minimizing cost, workforce variability, inventory investment, and backorders are recognized as the multi-objective set to be handled and optimized simultaneously.

Table 5-1: Product Demand Data (Units)

Product Family	1	2	3	4
A	3500	6000	4000	1300
B	1200	2000	2800	3600

Table 5-2: Product # 1 related Data

Parameter	Description	Unit	Value
C_h	Holding Cost	$/unit	1.5
C_b	Back Ordeing Cost	$/unit	300
C_s	Subcontracting Cost	$/unit	500
C_σ	Setup Cost	$/change over	3000
t_l	Labour time required	Hr/unit	3
t_σ	Setup time	hr/changeover	16
B_0	Initial back order	unit	500
C_m	Materials cost	$/unit	390
I_T	Target Inventory level	unit	500

Table 5-3: Product #2 related data

Parameter	Description	Unit	Value
C_h	Holding Cost	$/unit	0.66
C_b	Back Ordering Cost	$/unit	200
C_s	Subcontracting Cost	$/unit	250
C_σ	Setup Cost	$/change over	1800
t_l	Labour time	hr/unit	2
t_σ	Setup time	hr/change over	24
B_0	Pre-Planning Back order	unit	0
C_m	Materials cost	$/unit	210
I_T	Target Inventory level	unit	500

Table 5-4: Workforce data

Parameter	Description	Unit	Value
C_F	Firing Cost	$/worker	2000

C_H	Hiring Cost	$/worker	1500
C_r	Regular time hourly rate	$/hour	12
C_o	Overtime hourly rate	$/hour	18
W_{max}	Maximum workforce level	worker	24
W_{min}	Min workforce level	worker	15
W_o	Pre-Planning workforce level	worker	20

5.9 Results

The four objectives: cost, workforce variability, Inventory Investment, and backorders described by equations (5.5)-(5.8) are optimized simultaneously. The number of GA generations is set to be 1000, the population size is set to 100 candidates, and the recombination rate is 30% of the population size. The archive size could be defined a priori as 10, 20, or even 100 members. Presenting 100 solution or even 20 could be very tiresome even for a group of decision makers. Therefore, 5 or 10 at most Pareto front solutions could be enough and easy to select from. SPEA2 did a good job in this regard by developing a truncation algorithm that utilizes the distance among objectives in the objective space. The outcome is almost evenly distributed front members. During the solution process, the archive size is set to be hundred, but at the end, it is reduced to be just 10 or 5 members. Table 5-5 lists a Pareto front of 5 members and Table 5-6 lists a 10-member Pareto front. The Pareto front provides a set of extreme points (best financials, lowest Inventory investment, most stable work force curve etc.) and a group of compromised solutions. Decision maker has several options to work on: capitalize on best financials, respond to customers by minimizing back orders, promote lean practice by reducing inventory, heighten employee morale, and maintaining system stability. The word “short” in table captions is used to mark the listed solutions identified with only their objectives. A complete solution point is a one that lists all the plans and their associated objectives as describe in earlier in Figure 5-3. The data obtained can be depicted graphically using charts as show in Figure 5-18. Charts are designed to be expressive and give immediate insights for the decision makers. Product plans are divided into demand and supply and inventory and back-orders plans. Demand and supply should be kept in balance. Inventory and back orders should be maintained at their minimum level—balanced lean and agile practice. Workforce charts reveal workforce changes. It reflects system stability. These changes should be either minimal or of a big magnitude to comply with best industrial practice. The last chart shows objectives values in radar charts. Since objective values are tremendously non-commensurate, a logarithmic scale is used instead. If objectives represent the results, plans are the steps of the system behaviour that bring those results. Progressive Modelling brings the attention to the ends without forgetting the "hows" that brings those ends. System behaviour can be analyzed and more corrective actions can be planned.

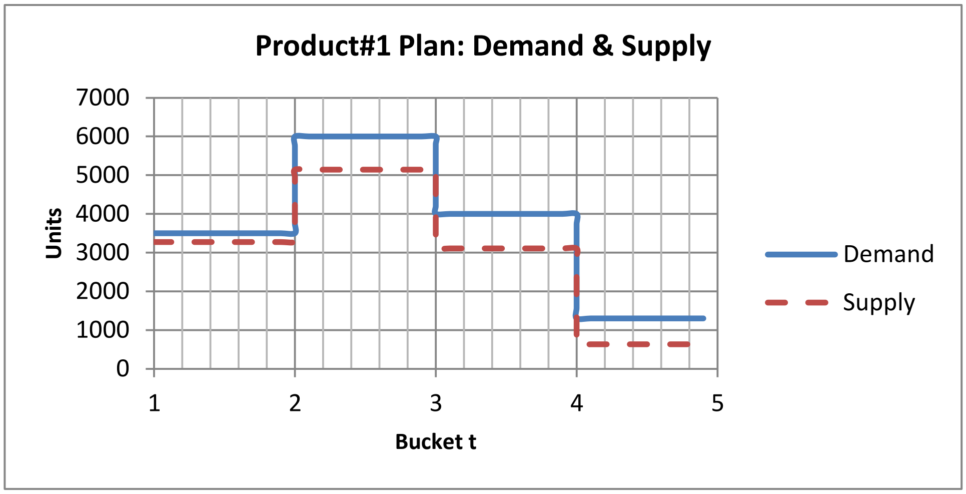
Table 5-5: Parto Front 5 short

	Financials[Min]	Backorders[Min]	Inventory[Min]	Workforce[Min]
1	10462231.54	177.54	2905300	2.78E-11
2	9509327.9	1497.9	1046800	0.000123747
3	10631305.04	1877.04	2016100	1.39E-11
4	8932839.08	4777.08	169200	0.735882292
5	9656241.28	137.28	2126300	0.018315864

Table 5-6: Pareto 10 Short

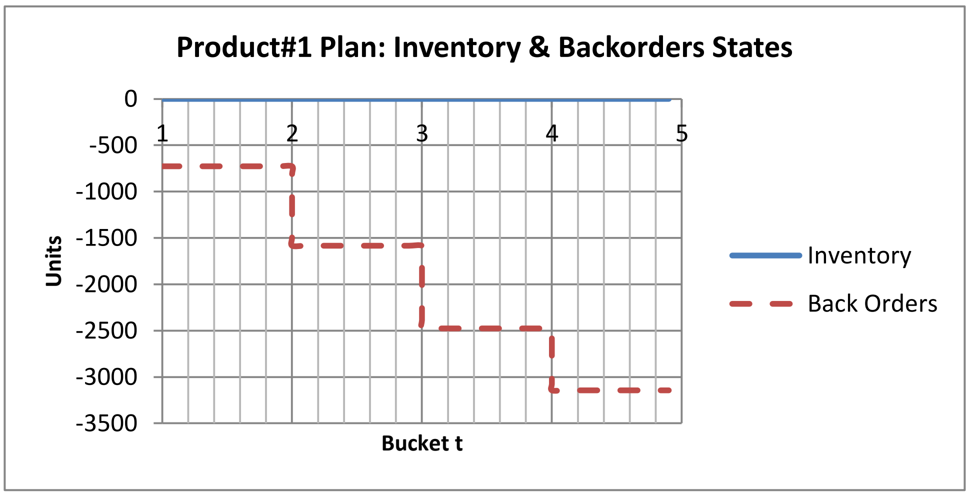
	Financials[Min]	Backorders[Min]	Inventory[Min]	Workforce[Min]
1	10462231.54	177.54	2905300	2.78E-11
2	9402801.92	1261.92	1662700	0.018439049
3	9509327.9	1497.9	1046800	0.000123747
4	10631305.04	1877.04	2016100	1.39E-11
5	10032869.36	129.36	2700500	3.38E-07
6	9423509.4	2501.4	352400	0.000370342
7	8932839.08	4777.08	169200	0.735882292
8	9994142.76	2438.76	1041900	0.000123522
9	10264053.04	1877.04	1455700	0.00012341
10	9656241.28	137.28	2126300	0.018315864

(a)



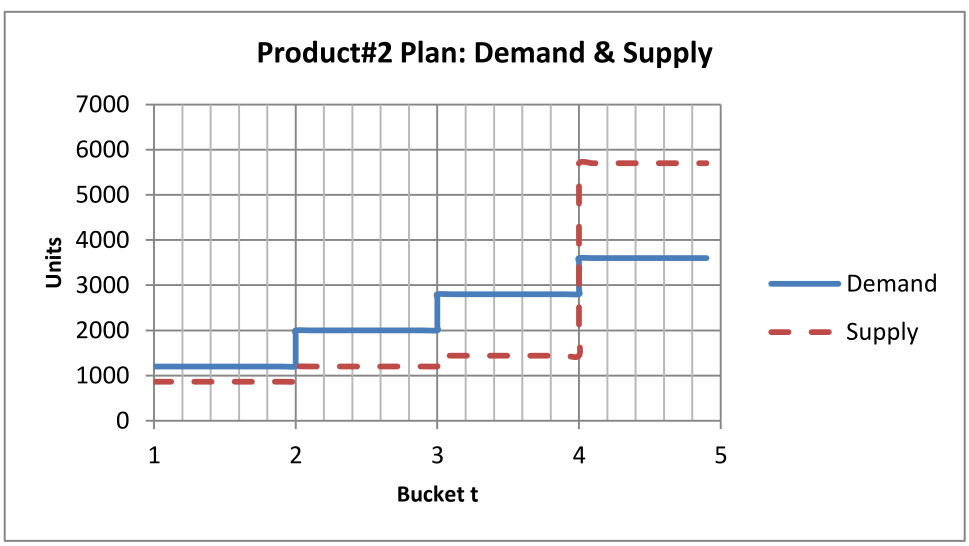
Product#1 Plan: Demand & Supply
Units
7000
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
Bucket t
Demand
Supply

(b)



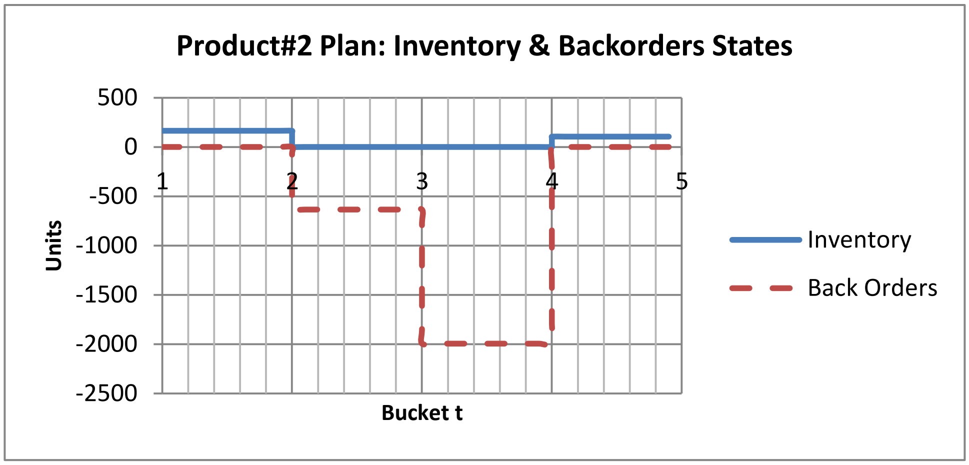
Product#1 Plan: Inventory & Backorders States
Units
0
-500
-1000
-1500
-2000
-2500
-3000
-3500
1
2
3
4
5
Bucket t
Inventory
Back Orders

(c)

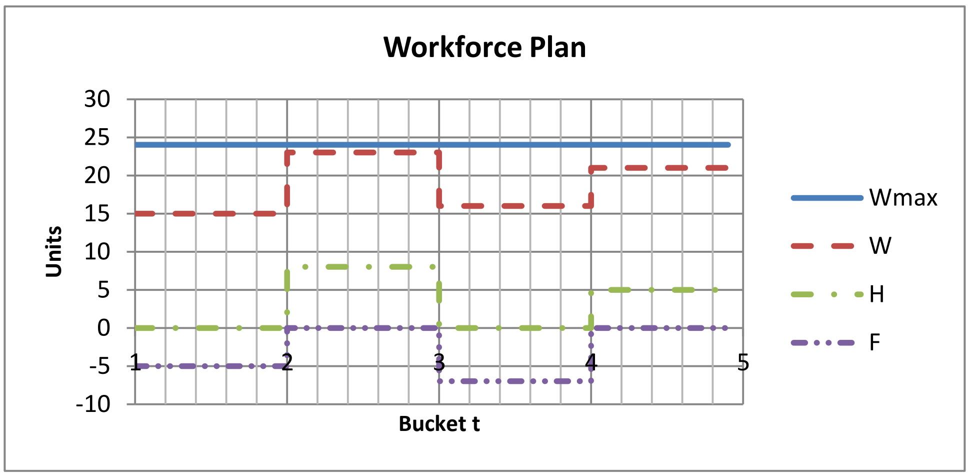


Product#2 Plan: Demand & Supply
Units
7000
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
Bucket t
Demand
Supply

(d)



(e)



(f)

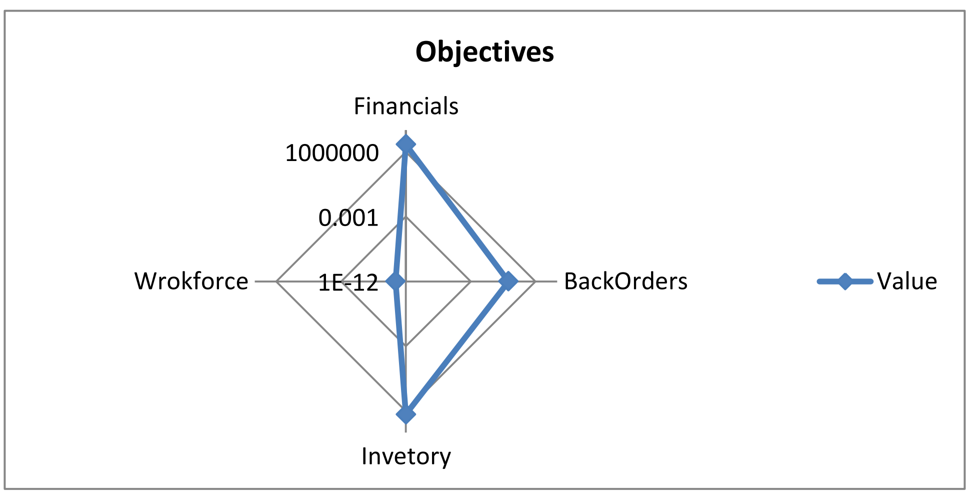


Figure 5-18: Solution Point charts: a-d Product plans, e: Workforce plans, f: Objective radar charts

5.10 Summary

In this chapter, PM as an integrated modelling approach was presented. In addition, the multi-objective multi-product aggregate production planning was introduced as a case problem with a numerical example. Several innovations were presented such as function templates, model deployment, turning problems into systems, couplers or micro-heuristics, incomplete chromosomes definitions, and defining APP plans as state machines. The MMAPP problem was redefined and modelled from a system perspective. A new system-oriented forward-looking mathematical model that embraces system states as an enclosure was developed. A novel solution algorithm that compiles several algorithms was also presented with some major changes that show how genetic algorithms and evolutionary multi-objective optimization algorithms could be adapted to make them progressive.

Everything related to PM was created and the challenges of the RMS were always in mind. Starting from the next chapter, an application related to RMS will be introduced. While working on RMS application, the mindset was to lessen the industrial-academic gap early introduced in this chapter.

Chapter 6 Reconfiguration and Operations Planning Problem: Foundations and Problem Statement

6.1 Introduction

Over the years, manufacturing technology has kept evolving profoundly to offer manufacturers around the globe many technological solutions to help them to be more competitive and to be able to create the orientation they seek in their markets and the image they want in their customer minds. Today's manufacturers use either Dedicated Manufacturing Lines (DML) or Flexible Manufacturing Systems (FMS) or a portfolio of both. Driven by economics of scale, DML are able to produce massive volumes of individual products with very competing unit costs as long as demand exceeds supply. In order to address the mid-volume and mid-variety production zones, Flexible Manufacturing Systems (FMS) is there to achieve what is known as economics of scope. FMS can produce economically a variety of products with different volume ranges. Nevertheless, that comes at the cost of having a capital-intensive system with overly estimated flexibility. In an initiative to overcome these shortcomings and to introduce a better agile manufacturing technology, the Reconfigurable Manufacturing Systems (RMS) concept was introduced in the late nineties. RMS promise a cost-effective response to market changes by combining the high throughput of DML and the flexibility of FMS (Koren, Heisel et al. 1999; Koren 2003). Mehrabi et al (2000) identified many aspects that present important research and practical challenges for reconfigurable manufacturing: reconfiguration of factory software, reconfiguration of new machine controllers, reconfiguration of modular machines, and reconfiguration of production systems. In order to provide the functionality and capacity needed when needed, system configuration changes can be in the form of adding/removing machines/stations to/from the system, adding/removing axes/spindles to/from machine tools, changing configuration of machine tools (Landers, Min et al. 2001). The main objective is to minimize the unused capacity and functionality, which is a new system flexibility lever missed by other manufacturing technologies. H. ElMaraghy (2005) classified manufacturing systems reconfiguration activities into two main types: hard/physical and soft/logical. Hard/physical reconfiguration activities may include adding/removing of machines, adding/removing of machine modules and changing material handling systems. Soft/logical reconfiguration activities include re-programming of machines, re-planning, re-scheduling, re-routing and increasing/decreasing the number of shifts or the number of workers.

RMS brings many challenges to the manufacturing research arena in terms of modeling, managing, and controlling the new technology. In fact, the early mindset that governed this research at its early stages was to develop a new manufacturing planning and control system specifications to handle the new technology. The issue of compiling all research related to RMS under one umbrella led to the advent of the concept of changeable manufacturing(Weindahl, ElMaraghy et al. 2007; ElMaraghy 2009). Many advancements brought by CMPC systems and Progressive Modeling were developed to address many challenges of RMS systems. RMS is an evolving system by nature and its development is an ongoing concern. The technology itself is still vague in researchers' minds rather than a materialized full-fledged one, which make analyzing and modeling it a hard task. Progressive Modeling is a forward-looking, multi-disciplinary modeling approach that has the flexibility to handle many challenges of the immature yet very promising technology and overcome the lack of data availability. With the new modeling paradigm, RMS would be treated like any other manufacturing technology. The Reconfiguration and Operations Planning (ROP) analytics, math models, solution algorithms, and case study described in this chapter and the next three ones reveal the new PM potential.

This chapter is about introducing the ROP problem and its related foundations. After presenting some ROP related literature review, the new amorphous reconfigurable manufacturing process is presented in order to define the ROP data model. In addition, some issues related to RMS are discussed before formulating the scope and the objective of the ROP problem statement. The chapter concludes with defining the component model of ROP problem.

6.2 Related Literature

Today's manufacturing environment has many requirements summarized by Bi et al (2008) as follows: shorter lead times, more product variants, low product volumes, and low prices. They stated the importance of these major requirements in choosing the appropriate production paradigm, and identified three major critical issues that should be involved in any type of RMSs: architecture design, configuration design, and control design. Architecture design defines the system relations and their interactions. Configuration design determines the system configuration under a given system architecture for a specific task. Control design determines the appropriate process variables so that a configuration can be operated to fulfill the task satisfactorily. Even though the Bi et al work reports the state-of-the-art of reconfigurable manufacturing till recently, the challenge of dealing with the new RMS amorphous process and its underlying changeable system has not been addressed. In the following subsections, the related literature to the Reconfiguration and Operations Planning (ROP) is presented:

6.2.1 Product Configuration Linkage

The product family – configuration linkage has attracted the attention of many researches. ElMaraghy (2009) developed a hierarchy of product variants from individual product features to product families, portfolios, and platforms, and illustrated the effect of these variations on several manufacturing support functions and enablers of change at the levels of product design, process planning and parts/sub-assemblies/product families' definition. The concept of evolving product families was also presented. Abdi and Labib (Abdi and Labib 2003; 2004; Abdi and Labib 2004) linked the market and the manufacturing system through a design loop in order to group products into families and to select the most preferred product family over each configuration stage. A case study was presented to illustrate their analytical hierarchical process (AHP) model for designing RMSs. Xiaobo et al (2000; Xiaobo, Wang et al. 2000; Zhao, Jiancai et al. 2001; Zhao, Wang et al. 2001) proposed a framework for a stochastic model of an RMS. The issues of optimal configurations in the design stage, the optimal selection policy in the utilization stage, and the performance measure used in improving these systems were discussed. Each family of products was mapped to one configuration of the RMS. In Xiaobo et al (2001) the problem of selecting the optimal configuration for each product was formulated using stochastic model and two algorithms were devised to solve it. Ohiro et al. (2003) modified the work done by Xiaobo et al by choosing the best configuration according to order quantities instead of associating each product.

The ROP consider the system design process independent from the configuration selection process. From ROP perspective, the configuration selection is an operational decision that is mainly identified to respond to both product volume and product mix changes.

6.2.2 Reconfiguration Planning and RMS System Design

Spicer et al (Spicer 2002; Spicer, Koren et al. 2002) suggested that scalable reconfigurable manufacturing systems (scalable-RMS) should consist of standardized modular equipment that can be quickly added or removed to adjust the production capacity. Machining systems can be arranged in parallel, series, hybrid, with or without crossover. For the same number of machines, they argued that pure parallel configurations should have the best throughput and scalability performance yet with more quality streams than other types of configurations. Spicer and Carlo (2007) discussed the optimal configuration path of a scalable-RMS that minimizes investment and reconfiguration costs over a finite horizon of a well-known demand. Their model comprehends labour costs, lost capacity costs, and investment/salvage costs due to system reconfiguration and ramp up. They used the dynamic programming (DP) to find an optimal solution model for the multi-period scalable-RMS. A combined integer programming/dynamic programming (IP-DP) heuristic was also presented to allow the user to control the number of system configurations considered by the (DP) in order to reduce the solution time while still providing a reasonable solution. Since it is considered the first model to define the reconfiguration costs, the Spicer and Carlo model is analyzed further later in chapter 7. Son (2000) developed a methodology to design economical Reconfigurable Machining Systems (RmSs) for a deterministic demand scenario for the early stage of configuration design. This methodology generates configuration paths for changing demand by considering reconfigurations between demand periods, using a configuration similarity index, as well as the cost efficiencies for each demand period utilizing Genetic Algorithms (GAs).

Kuzgunkaya and ElMaraghy (2007) developed a fuzzy multi-objective mixed integer optimization model to evaluate RMS investments used in a multiple product demand environment. Their model incorporates in-house production and outsourcing options, machine acquisition and disposal costs, operating costs, and re-configuration cost and duration for the utilized modular machines. The resulting system configurations are optimized for lifecycle costs, responsiveness performance, and system structural complexity simultaneously.

Youssef and ElMaraghy (2007) presented an RMS configuration selection approach consisting of two phases: the first deals with the selection of the near-optimal alternative configurations for each possible demand scenario over the considered configuration periods and uses a constraint satisfaction procedure, Genetic Algorithms (GAs), and Tabu Search (TS) for the continuous optimization of system's capital cost and availability. The second phase utilizes integer-coded GAs and TS to determine the alternatives, from those produced in the first phase that would optimize the degree of configuration smoothness over the planning horizon. configuration smoothness is a metric that provides a relative measure of the expected cost, time, and effort required to change from one configuration to another rather than estimating the exact time and cost of the reconfiguration process, which is difficult to evaluate (Youssef and ElMaraghy 2006).

Progressive modeling eliminates the need to use metrics, realistic evaluation criteria could be defined easily now. Chapter 7 illustrates the mathematical modeling and chapter 9 describes the case study development. The data model presented epitomizes how to deal with the evolving nature of RMS as well.

6.2.3 RMS Operations Management

Spicer and Carlo (2006) developed an integer programming based iterative algorithm for finding the minimum configuration cost of a multi-product system. They developed a mathematical formulation to minimize the system investment and operational costs in a multi-product scalable-RMS. They also proved that if the inventory control policy is incorporated during the system design process, a costly inventory control may result. They concluded that the simultaneous approach yields significant improvement over the traditional (decoupled) approach. Liu et al (2006) proposed a methodology for the cost-effective reconfiguration planning of the multi-module-multi-product RMS that best reflect the market demand changes. They formulated the problem as an optimization procedure and defined it as the best reallocation of part families to production modules of an RMS. A Genetic Algorithm (GA) approach is proposed to overcome the computational difficulties caused by the problem complexity. Effectiveness of the proposed methodology is demonstrated with a case study. Abbasi and Houshm (2009) introduced a methodology to adjust rapidly and productively scalable production capacities and the functionality of system to market demand. It is supposed that arrival orders follow Poisson distribution and they are missed, if they are not available. According to these assumptions, a mixed integer nonlinear programming (MINLP) model is developed to determine optimum sequence of production tasks, corresponding configurations, and batch sizes. A tabu search based procedure is used to solve the model. Freiheit (2004) reported that the science of manufacturing requires quantitative models to predict key performance metrics of flexible and reconfigurable manufacturing systems in order to reduce the set of all possible manufacturing system configurations to a feasible set, and then make a selection of the best configuration for a specific production circumstance.

6.2.4 Progressive Modeling and the Gap in the RMS Science

The work done in the literature is greatly appreciated; however, Progressive Modeling brings a new paradigm and an assorted tool-kit that not only addresses the RMS related problems but also is able to redefine many of them. Some issues and foundations are presented in this chapter and the next three ones. Progressive modeling brings a new set of scientific foundations upon which a new generation of quantitative models could be defined for the first time in the RMS field. Aiping and Chao (2009) argued that a study of reconfigurable manufacturing systems modeling method that can effectively analyze the dynamic characteristics and enable the system has good reuse, integration and scalability has become an urgent need. The ROP problem shows to what extent progressive modeling can bind the reconfigurable manufacturing problems to a new paradigm.

6.3 RMS: The New Amorphous Process

With the advent of RMS, the manufacturing process became incredibly amorphous and a new process capability spectrum is defined, see figure 1. At one extreme, a high volume production system—almost dedicated manufacturing system with scalability options is located, and at the other lies a highly functional production system with possible scalability options—a flexible manufacturing system with flexibility variants. The spectrum defines a countless number of volume/functionality pairs. The volume can be expressed in throughput figures (e.g. machine hours available/day or /month). The functionality could be expressed in variety terms (product sets that can be manufactured). For every configuration of any RMS system or RMS system module, both throughput and product list and workforce attached can be defined. An RMS system module is a group of machines that can be laid out together or reconfigured to produce a specific product or execute a certain manufacturing process. RMS system modules could be also described as reconfigurable cells. Same product can be produced using different cycle times and operating costs as will be described in this chapter and the next three ones**.** Since an RMS is an evolvable system, these values can be updated over time. Configurations can be added or removed, system modules can be extended or retrenched, and both process and product can be further developed. This new amorphous process is a seminal RMS characteristic and has greatly inspired some foundations and more innovations of progressive modeling itself by defining data models, introducing the attributive and advanced mathematical notation, hierarchical binaries and others which would be described in this chapter and the next three ones. These foundations and innovations prepare the ground upon which ROP is defined, modeled, and optimized. From the progressive modeling perspective, the objective is to overcome all the challenges of the new amorphous process and make the modeling process seamless and easy to develop. In this chapter, the Reconfiguration and

Operations Planning Problem (ROP) is defined for the first time. Since the RMS is not well-discussed from the operations management perspective, many principles related to RMS itself are introduced to serve as a foundation for presenting the ROP data model, logic, planning structures, and the case study presented.

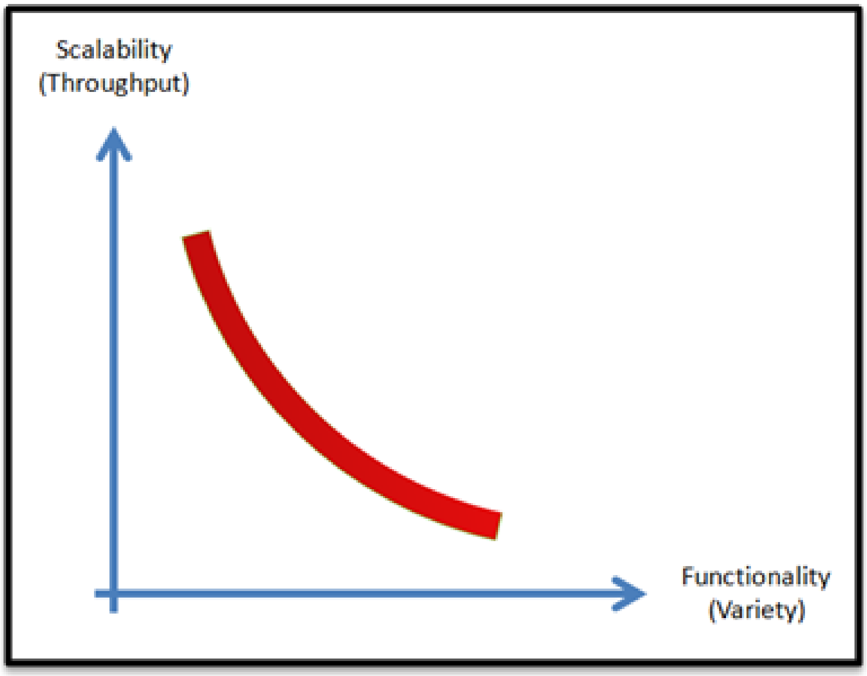


Figure 6-1: The New Amorphous Process Capability Spectrum

6.4 Manufacturing Process in an RMS Environment

6.4.1 RMS: Data Model Perspective

Keeping the new process capability spectrum defined by RMS process in mind, any RMS implementation, i.e. a physically established system, could have one or more modules and their associated libraries of configurations. In RMS context, modules can be reconfigured using their reconfiguration libraries that define a corresponding set of optimal configurations for each one of them. These libraries could be stored in system repository, as either database or manual documents. The system database could be a relational or a hierarchical one e.g. XML. All system/module configurations can be stored in configuration tables with other related data, which defines module data. In this study, an RMS module is defined as an integrated part of a reconfigurable manufacturing system that can be reconfigured to change either its capacity or functionality, or both and is designed to produce a predefined set of products. It defines a configuration domain or space, i.e. a list of configurations that could replace each other within the same physical (space) or/and logical (purpose) boundaries. The word module has been used in the RMS literature to disseminate two different meanings: a machine module (which is an integrated part of reconfigurable machine tool) and system module (which is a group of machines that are laid out and operated in order to produce a product or a set of products). In this study, the least level of granularity that ROP addresses is the configuration level, so the term module refers to RMS system module. Liu et al (Liu, Wang et al. 2006) was the first to present the concept of multi-module multi-product RMS in order to define their reconfiguration planning problem. They defined product, module, and planning views. Even though the ROP is different in scope, details, and problem definition, the work of Liu et al is accredited for being the first to define such a system structure. Liu et al defined scaling, conversion, scaling/conversion, and expansion as distinct reconfiguration operation that can be executed to change module configurations. In this study, a single product scalable module or system is called scalable system for simplicity while a multiproduct module or system is called functional system even if it have some scalability options.

Unlike Liu et al (Liu, Wang et al. 2006), the module definition here is thoroughly defined. The data model presented in this chapter assumes all Liu's et al reconfiguration operations to happen and considers the configuration level is the least granularity level in describing the RMSs. The internals of a system/module configuration are system design issues. A couple of configurations may differ only by just adding or removing just one machine module. It could be also a big change in the number of machines, their layout, buffers existing etc. From the ROP perspective, what counts are product sets, production rates, and other reconfiguration and set up data. A module defines a workforce skill set; therefore, hiring, firing, and operational costs should be module specific. All RMS modules have a unified inventory and its market link is independent of how the system is internally structured. This conceptual framework of RMS should be strongly valid for any RMS implementation.

6.4.1.1 ROP Configuration Data Model

A system configuration is an arrangement of machines that can produce a product or a set of products. From ROP perspective, every product has its own manufacturing parameters: product cycle time, a ramp/set up time, unloading time, and their associated costs. Every configuration has also a certain workforce level that is responsible for the manufacturing process. Every configuration has a corresponding list of configurations that can be reconfigured from. All the data necessary for the ROP purposes are defined in Figure 6-2. Both reconfiguration time and cost are configuration path dependent. Both of them depend on precedence relations between different process configurations. Ramp up process is a configuration-product dependent. Over time, both ramp up spike-count and width should grow shorter through continuous and better understanding of configurations dynamics and interrelations. Corresponding data files are updated accordingly and should be ready to be reflected on prospective operations. Configuration design should be built around both the functionally and capacity needed. Product variants and throughput (1/cycle time) are very important pieces of information for ROP. Other system capacity levers can help in reducing the number of configurations used and make the incremental capacity steps larger. This approach has very good implications on the capital equipment investments decisions that might be made throughout the lifecycle of an RMS system.

Configuration "3302"

Ramp Up, Workforce, etc.

Item	Units	Value
WF Level	Man/Period	16

Reconfiguration Data

#	ID	Time (Hrs)	V.Cost ($/hr)	F.Cost ($)
1	3301	4	600	0
2	3302	6	600	0

Product List

#	1	2
ID	105	108
Ramp/SetUp Cost ($)	12000	14000
Ramp/SetUpTime (Hrs)	36	30
Unloading Cost ($)	1600	1400
Unloading Time (Hrs)	6	4
CycleTime (min)	8	8
V. Mach Cost ($/Unit)	0.65	0.7
APP. Max. Thrpt (Units)	14580	14580

Figure 6-2: ROP Configuration Data Model

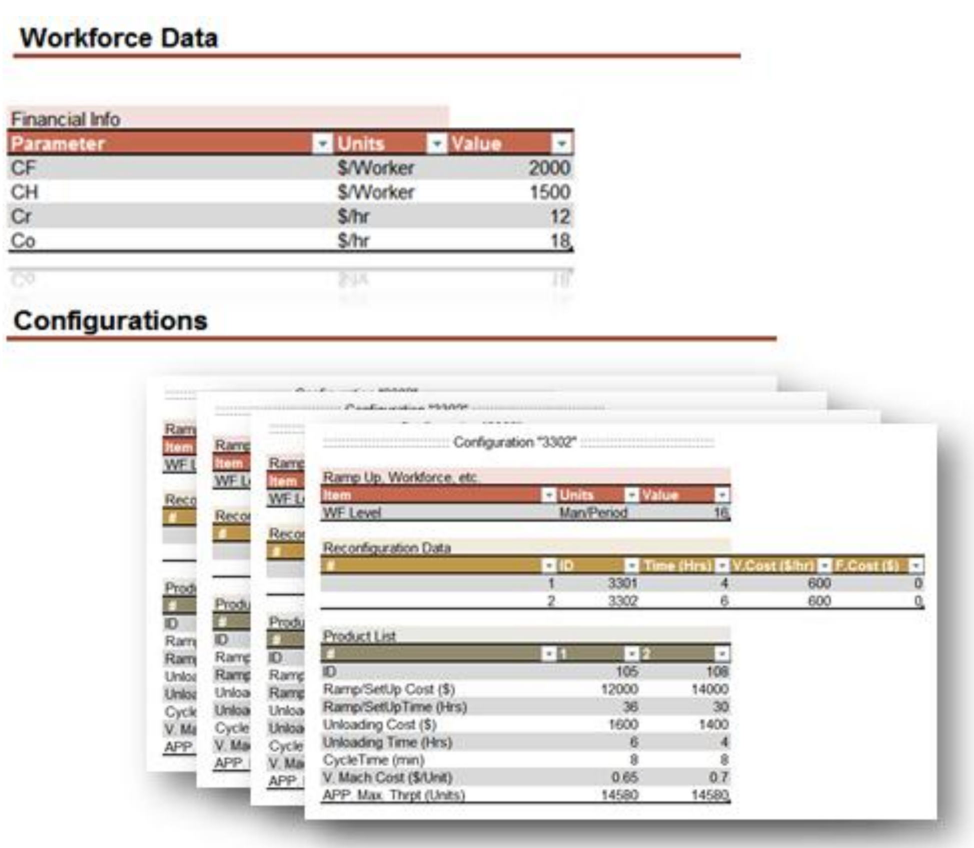


Figure 6-3: an ROP Module Data Model

Whenever a module needs a reconfiguration process, an existing product should to be unloaded first. If a module can produce multiple products, the loaded product has to be unloaded before the new ones are loaded; therefore, the unloading operation could happen as a pre-step before either system/module reconfiguration or new product loading operations. By decoupling the unloading operations, both reconfiguration time and setup times become more concise and expressive and the interdependent relations are just limited to products and their configurations. The reconfiguration time became independent too from the already loaded products or those that should be immediately loaded. The unloading operation definition is introduced for the first time in the RMS literature. Ramp up/set up time, production time, and unloading time define a product make cycle that will be described later. The ramp up/set up time are used interchangeably in this study. Every configuration registered in the library has a given ID. If the RMS under study consists of several modules, each should have its own ID. From time to time, configurations can be upgraded or updated. Some configuration could be unregistered form the library for good. There could be many reasons for configuration termination; examples may include obsolete products, replaced machines, inefficient configurations etc. Similarly, new configurations may join the library for any of the following reasons: better configuration layout, newer products, newer material handling equipment, process improvements, etc.

6.4.1.2 Product Data Model

RMS divides the product data into two main parts: the first is configuration dependent data and is encapsulated within every configuration definition. Product setup, unloading, machining cost, cycle time are all configuration dependent. The other part is operations independent part that includes product demand, holding costs, subcontracting costs, initial and target inventories. Figure 6-2 shows the configuration dependent part, and Figure 6-4 shows the configuration independent part.

Product Data

Demand

ID		1	2	3	4	5	6	7	8
1	101	2500	4000	2000	1500	6000	3000	1500	
2	103	3000	2000	4500	3800	4200	5000	4500	
3	106	1500	1200	1000	2500	2000	1400	1700	
4	108	3000	2400	1500	1500	3400	2000	1100	

Product Data

	ID	Io (Uni	Bo (Un	Ch ($/U	Cb ($/U	Cm ($/U	Cs ($/U
1	101	0	2200	0.5	40	20	35
2	103	1300	0	0.6	27	17	0
3	106	0	2200	0.5	40	20	0
4	108	0	1000	0.7	30	19	30

Figure 6-4: ROP Product Data—Configuration Independent data

Both the products and modules data models are encapsulated in the system data model. Calendar data and Working regulations are additional important system level data needed to perform ROP operations.

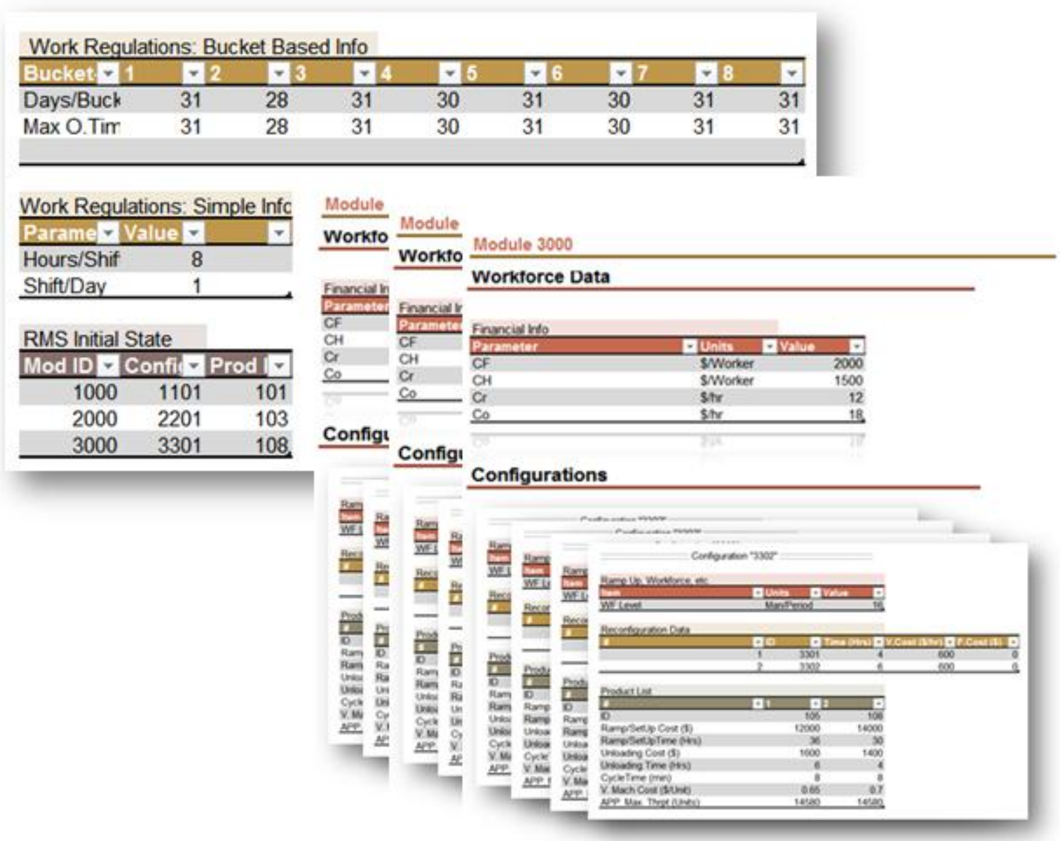


Figure 6-5: an ROP System Data Model

6.5 RMS Important Issues

6.5.1 Man Power: Reconfiguration impact

With every configuration, there should be a workforce group attached. Hiring and firing and module reconfigurations should happen at calendar boundaries, i.e. months or quarters. The size of workforce is system design consideration that should be identified with every configuration joins a module configuration library. The new technical consideration hedges morale consequences relatively. Hiring and firing processes should be known a priori. In the RMS literature, considering the workforce in the analysis of RMS are usually ignored and even those who considered labour costs did not take into consideration workforce adjustments. According to the ROP data model, every configuration should report its workforce level and every module should define its workforce related cost parameters: hiring and firing costs and regular and overtime costs. RMS modules rather than configurations define workforce skill pool. That is why workforce related data should be module level data.

6.5.2 Scalability Options

In the traditional manufacturing systems, scalability options were limited to facility expansion, which was a long-term decision. The reconfigurable manufacturing technology brings scalability options to the shop floor as a low-cost short-term alternative. Holding stocks of Inventory, overtime production, and sometimes subcontracting could be considered an immediate and cheap scalability solutions. For the same product list (i.e. a certain defined system functionality), providing a wide spectrum of scalability variants should never prove a cost effective solution. Capital equipment is not the right place to freeze system financial resources. Configuration utilization index could be defined as a warning index for the system designers, configuration managers, and financial officers to identify that there are some machines or configurations that might need reconsiderations. A balance of immediate scalability and short-term scalability should be taken into considerations. From ROP perspective, the system structure variants, system or module configuration libraries are assumed to be maintained. If any of these variants is ever to change, this should be considered a milestone at which the planning process has to be restarted. All the scalability levers should be orchestrated to optimize the performance of reconfigurable manufacturing systems and make their products highly competitive.

6.5.3 Demand

RMS, FMS, or DMS may be able to produce the same product but the question will always be in what quantity and for how much. RMS creates its own competitive edge by being able to produce almost mid-variety and mid-volume products with the option to make variants over time. The last characteristic or option is intrinsic and distinguishable to RMS. Product introduction or major design changes might affect the already existing configurations, machines, machine modules, and other equipment. RMS libraries are supposed to be stable at the short rang i.e. minor configuration changes. As reported earlier, whenever the configuration library is updated with a new product or a new estimated demand, a re-planning process should be executed on a rolling horizon basis. Demand management is separated from operations management. The demand that may be presented to the operations planners or controllers represents a sales plan that prepared carefully as a collaborative work of sales, marketing, and may be other departments. In this case, the operations managers can be hold responsible for their plans and their decisions. CMPC defines roles and responsibilities very strictly without undermining the synergies that might be created out of collaborative interactions among different business functions. The philosophy propagates to modeling process as the ROP mathematical model reflects later in chapter 7.

6.5.4 Capital Equipment Investment

Unfortunately, most of researcher who addressed the RMS from operational level perspective embedded the machine investment costs in their analyses and models developed. Investment costs are irrelevant to the short term planning processes. Machines are acquired to stay for years, a technology is chosen to identify a manufacturing enterprise competitive edge: Targeting new markets, raising market share, creating a certain level of added value etc. machine costs are committed costs and they are mainly relevant to identifying business direction. Only if a certain machine or machine module is bought specifically for a certain configuration is planned to be sold within the ROP planning horizon, only then it should be relevant and it should be taken into consideration. The capacity needed when needed made many researchers assume that the machines would be bought and sold during reconfiguration periods. This should not materialize later as a realistic assumption or a feasible process. Even if this would be the case later, this should be a fixed reconfiguration cost rather than investment decision. The ROP ignore the buying and selling of machines concept as a relevant planning decision.

6.6 ROP—Problem Statement

The Reconfiguration and Operations Planning (ROP) problem describes a manufacturing planning and control system function that defines a new approach of planning and managing manufacturing processes in a reconfigurable manufacturing environment. On the supply side, the system is composed of a set of modules. Every module defines its process domain. The process domain is defined by a number of configurations that an RMS module can be reconfigured from. Every configuration defines its workforce level, configuration variants with their associated reconfiguration costs and time, product lists and their related product make operations parameters—setup times and costs, throughputs, operation costs, etc. ROP treats RMS like any other well-established system; the system has an inventory, may extend its working time hours—overtime, and *can outsource some of its products—subcontracting. On the demand side, there is a portfolio of products with their forecasts estimated over their planning horizon. Capacity adjustment (system reconfiguration), Inventory, overtime, and subcontracting are considered an array of levers by orchestrating them, demand fluctuations can be mitigated and a competitive edge can be created. ROP put all these options under one frame in order to decide how they all can be best planned to maximize the system operational and strategic objectives. By analyzing and modeling the internals of an RMS manufacturing process, ROP should contribute to promoting and identifying the RMS as a technology of choice within a wide range of circumstances.*

6.7 Planning Foundations: Buckets, Slots, and Life Cycles

Since ROP has its roots in the classic aggregate planning problem, the same perception of time horizon length is assumed (6-18 month is most appropriate). The planning period length is also maintained monthly. The planning process is concerned with which configuration to be loaded, at what time, and for how long. It also defines which products to be produced by a certain configuration and their sequencing during these operations. Once a configuration is identified and a sequence of products is determined, the planning period is divided into different time slots during which a certain operation is executed. ROP is concerned with the following operations: reconfiguration, ramp/set up, production (regular time and overtime), and unloading. Since the ROP is a novel problem in the RMS literature, many structures need to be defined and grasped first before addressing the ROP problem as an industrial optimization problem. All the remaining subsections define different time-related planning structures called buckets.

6.7.1 Demand Buckets

Demand bucket defines market dependent time frame during which aggregate quantities of product mix that a manufacturing firm introduces to its market can be quantified. Demand periods usually resolve to be monthly periods. A balance between demand and supply is always a well-sought objective of any manufacturing firm. Since the market is the major driving force of any manufacturing firm, the concept of demand bucket is used as the leading planning frame to define other planning buckets. If a demand bucket is identified on a monthly basis, other buckets are supposed to be measured by months and so on.

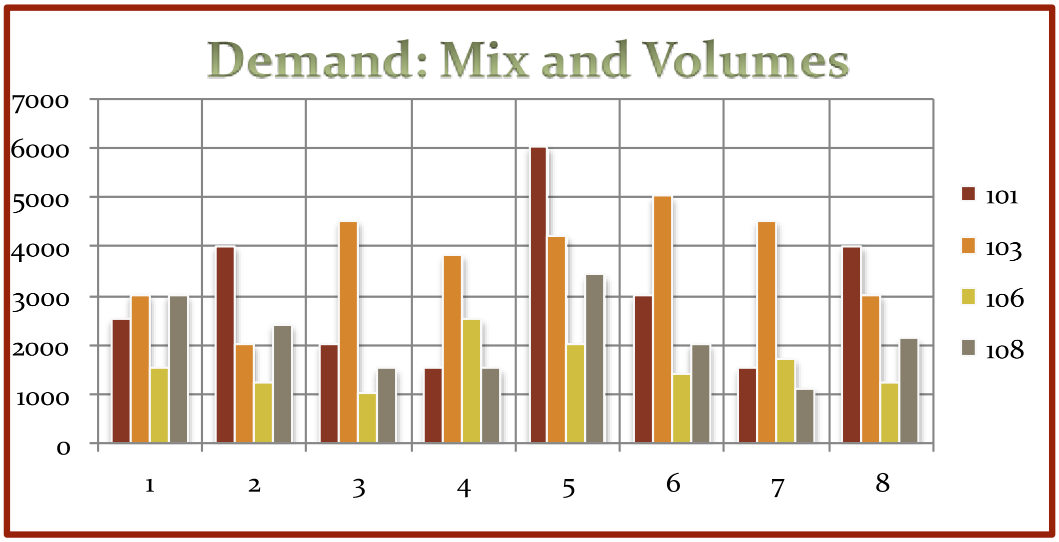


Figure 6-6: Demand Buckets: mix and volumes

6.7.2 Operations Bucket

In this study, an operations bucket is defined for every corresponding demand bucket. Even though demand and operations buckets are assumed monthly, ROP data model can define other shorter or longer bucket durations. An operations bucket holds an ID of configuration loaded and whether a reconfiguration is needed or not (Reconfiguration binary variable). Once the reconfiguration operation is identified and after consulting the system calendar, the number of hours available for manufacturing operations can be determined. Every configuration bucket contains at least one product to be produced. According to product-mix demand and configuration specification, i.e. products that can be made, this list can be updated. For every product loaded, product set up/ramp up, production, and unloading times can be updated. A product tuple holds

5 pieces of information, product ID, ramp/set up time, unloading time, regular production time, and overtime. Figure 6-7 shows a configuration bucket and its different pieces of information. Configuration "2201" is the active configuration; it was also the active configuration during previous bucket, so there is no reconfiguration is needed. The time slot gives the time available for operations. Since there is no reconfiguration is encountered, it is equal to the planning bucket available time. The month of February is assumed to have 224 of working hours (28*8). At the bottom row of the bucket, the list of product tuples is shown. A configuration bucket always starts by the unloading slot, which holds the time of unloading of previous product if it has to be unloaded. Actually, the unloading and set up slots are represented by two variables: the first is a binary that represents any of such operations are needed and the other reflects the time consumed. Schematically both variables are shown by just one slot. A time value indicates an operation is encountered and a fraction of time represented by its equivalent slot value is consumed. An "n/a" slot means the operation is not executed for the current operations bucket and the time consumed is known implicitly to be zero. The time consumed during the production process of product p is divided into regular time and overtime if allowed. In the next couple of chapters, all the related logic by which all these slot values are updated and optimized are described.

6.7.3 The Planning Horizon

The planning horizon determines for how long the planning process should be valid. The length of the planning horizon is measured by the number of buckets sales and operations plans should be maintained. Operations, sales, inventory policies should be described and maintained in sync over the planning horizon. The different planning views or structures are discussed further in the upcoming three chapters. Change ready MPC systems encourage synergistic and collaborative solutions that can be generated among different system functions. Taking into consideration the planning horizon notion, configuration paths and maps can be defined.

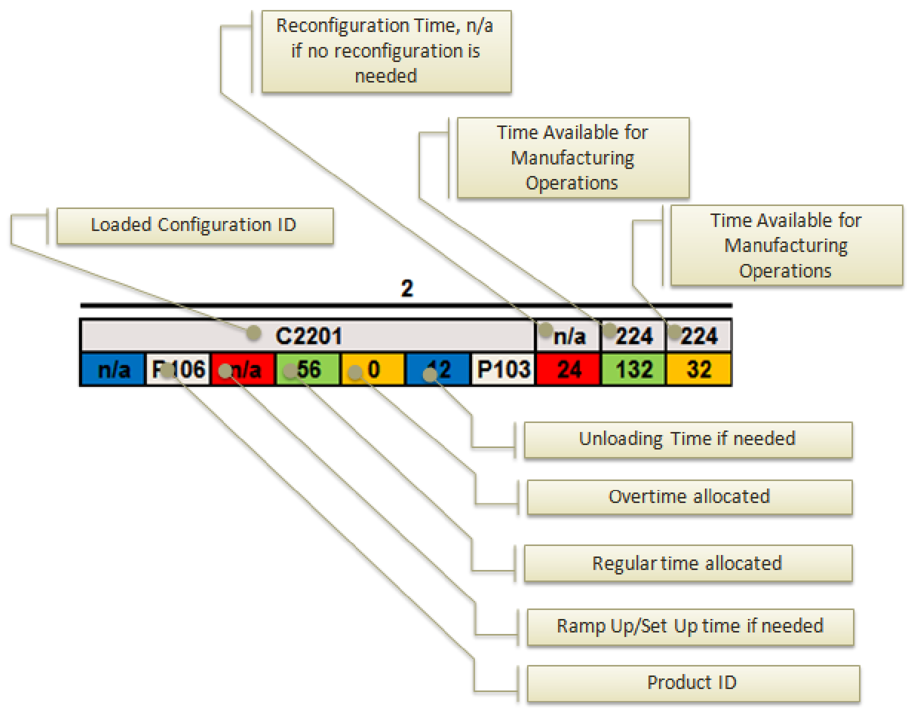


Figure 6-7: an ROP Operations Bucket

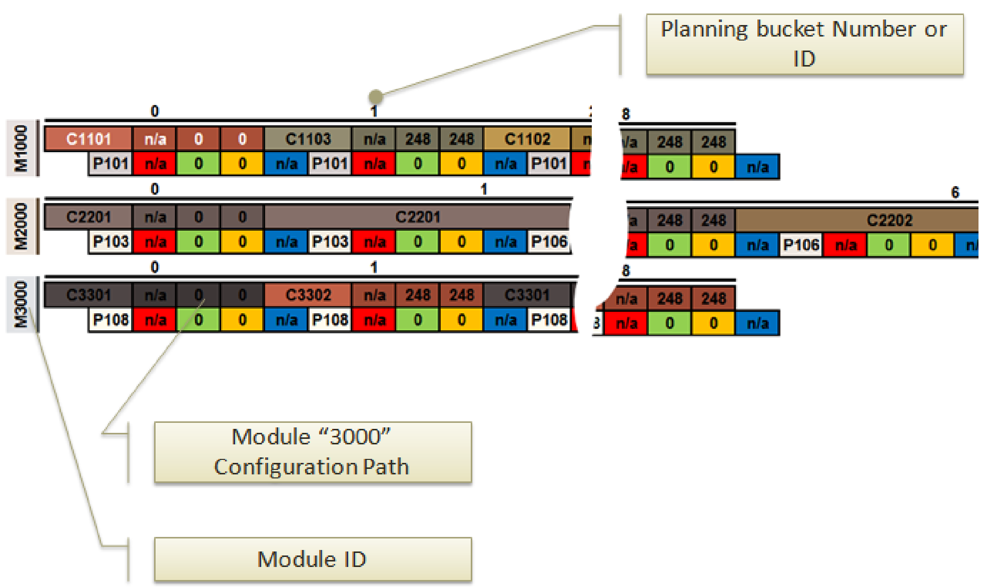


Figure 6-8: Configuration Paths/Maps

6.7.3.1 Operations Path and Configuration Path:

An Operations path (Figure 6-8) is the list of operations buckets defined for an RMS module. Configuration path is the list of configurations over the planning horizon defined by their configuration IDs. This definition is similar to the one provided by son (2000). In this study, both configuration path and operations path can replace each other however. From that perspective, an operations/configuration path defines all the system operations from time perspective. Every module defines its own configuration path. Detailed building process of configuration paths are described in chapter 8.

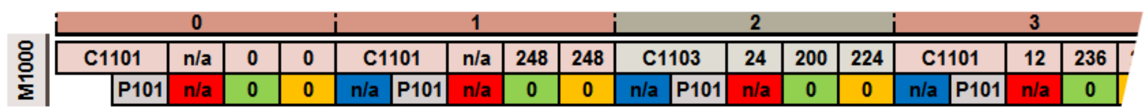


Figure 6-9: Module 1000 configuration path

6.7.3.2 Configuration Map

A Configuration map (Figure 6-10) is the set of all configuration paths that the system may encompass over its planning horizon. The configuration map encapsulates all the system and product operations defined by an RMS system and its allocated times. Further details about configuration maps are described in chapter 8.

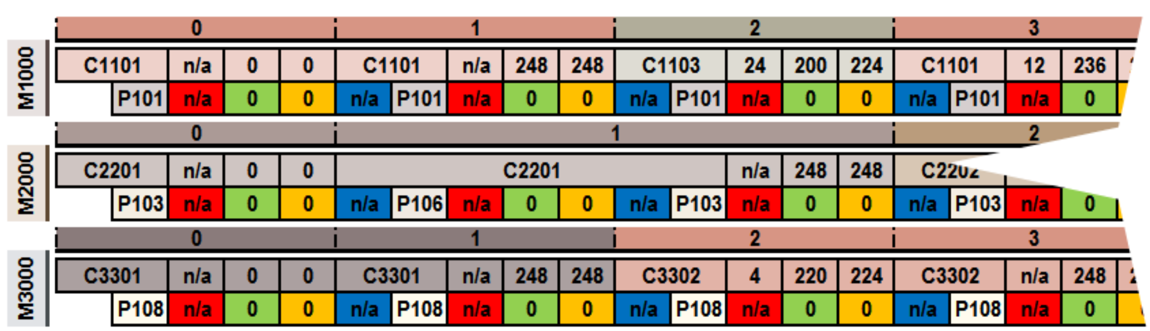


Figure 6-10: RMS configuration map of a 3-module system

6.7.3.3 Planning Views/Structures

Configuration maps represent operations planning statement in a very concise format. An RMS configuration map serves as the seed of all other system plans. Other system plans, which can be also called planning structures or planning views, include workforce plans, product plans, inventory/back orders plans. ROP presents a comprehensive planning system in an RMS environment. All the levers are synchronized in order to create a balanced system performance. Figure 6-11 shows product make plans, Figure 6-12 shows product supply plans, and Figure 6-13 shows workforce plans. More details are given in chapter 8 and chapter 9 illustrates a comprehensive case study to illustrate the results and sharpen the concepts and foundations developed in this chapter.

Product::103 Module::2000

Column1	1	2	3	4	5	6	7	8	
Reg Time	N/A		68	80	176	N/A	112	112	92
Reg Quantity	N/A		360	600	1056	N/A	780	840	504
Max Over Time	N/A		16	20	44	N/A	26	28	22
Max OT.Quantity	N/A		96	150	264	N/A	195	210	132
Ov Time	N/A		16	20	44	N/A	26	28	22
OvQuantity	N/A		96	150	264	N/A	195	210	132

Figure 6-11: Product make plans

Product 101 Plan

P. State	1	2	3	4	5	6	7	8	
D		2500	4000	2000	1500	6000	3000	1500	400C
Rmax		1584	1944	1224	1520	1984	2136	1224	216C
R		1584	1944	1224	1520	1984	2136	1224	216C
Omax		400	504	312	384	496	552	312	552
O		400	504	312	384	496	552	312	552
S		1020	1633	816	612	2449	1224	612	163
	0	0	0	0	0	0	0	242	587
	2200	1696	1615	1263	247	1318	406	0	C

Figure 6-12: Product Supply and Inventory and back orders plans

Module 1000 Workforce Plan

	P. State	1	2	3	4	5	6	7	8

Module 2000 Workforce Plan

	P. State	1	2	3	4	5	6	7	8

Module 3000 Workforce Plan

	P. State	1	2	3	4	5	6	7	8
W	12	12	16	16	16	16	16	16	16
H		0	4	0	0	0	0	0	0
F		0	0	0	0	0	0	0	0

System Workforce Plan

	P. State	1	2	3	4	5	6	7	8
W	34	34	45	46	46	38	42	42	46
H		0	11	1	0	0	4	0	4
F		0	0	0	0	8	0	0	0

Figure 6-13: Work force module and system plans

6.7.4 Product Make Life Cycles

In an RMS environment, scholars defined reconfiguration and ramp-up processes as production preparation operations. ROP defines a new product manufacturing cycle. The cycle starts with setting up (ramping up) the system for a product, executes product-manufacturing operations, and ends up with unloading that product. During a certain planning bucket, demand time frame, a product is allowed to be loaded once. A configuration life cycle defines the time frame that starts with unloading the previously loaded product at the end of the last operations bucket, the system reconfiguration is executed, and a series of product cycles except for the last product are triggered. The last product unloading process always marks the beginning of a new configuration cycle as reported earlier. The product tuple is a virtual product life cycle. Within the time frame of an operations bucket, all the operations of product life cycle are not guaranteed except the production operation. For example, if product “101” is decided to be produced during the current bucket and it was already loaded at the last slot of the previous bucket, no setup is needed during the current bucket. A virtual product life cycle is a trio of set up, production and unloading slots. The set up and unloading might not be encountered depending on the sequencing relations among products. The virtual life cycle or product tuples are defined in order to make product batch sizing very smooth and independent of demand time frames. When all consequent product tuples are merged, a realistic product make life cycle is obtained: a cycle which starts with just one set up, production over a period of time (might be greater than on demand bucket), and ends with an unloading operation.

6.7.5 A Reconfiguration Cycle or Configuration Bucket

A configuration bucket is the time at which a certain configuration is loaded into the system, system configuration. It should span a sufficient time to justify costs incurred, variable overheads that should add to product flexible costs, and hiring and firing expenses. Since reconfiguration process might accompany workforce level adjustments, a configuration should be loaded for at least once a month or its multiples. The longer the configuration-up period may span, the lower the contribution to flexible costs and consequently the higher the profitability could be. There are other considerations: unless there is a very stable demand, maintaining a certain configuration for long time means more inventory accumulation or lost sales possibilities. Determining which configuration to be loaded, configuration-up time, and configuration path, work-force attached at each configuration period over the planning horizon are identified as important decisions related to the management process of the supply-side of an RMS system.

6.7.6 P-Bucket or R-Bucket

The previous discussion of product make and configuration cycles is urgent for understanding configuration maps and their tight relations to operations management of RMS systems. The concepts discussed previously mould the following definitions:

R-Bucket: R stands for reconfiguration bucket; the concept of reconfiguration bucket is very important for determining product cost structure and evaluating system performance. Configuration up time is a critical decision in the RMS operations context.

R-bucket is the time frame that spans two consequent different configurations of an RMS system.

P-Bucket: P-bucket is very important to optimize batch sizes and to evaluate the exact costs of a certain product. Both R-Bucket and P-Bucket are complementary and extremely foundational in RMS operations cost accounting.

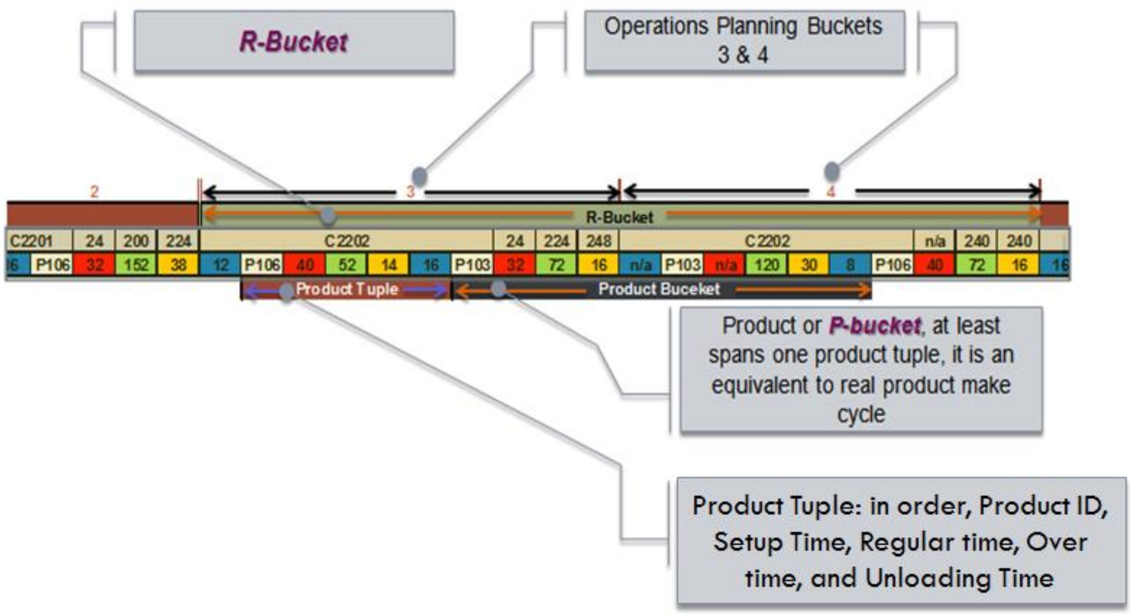


Figure 6-14: R & P Buckets

6.8 Progressive Modeling III and the ROP

This chapter shows so many new foundations related to the reconfiguration and operations planning problem: the data model, related issues, and planning foundations. All these topics can be listed under the first phase of the PM process, simply can be called the analytics. In the next chapter, the logic that governs is discussed. Advanced notation, hierarchical binaries, mathematical statements will be introduced as new PM advancements. Chapter 8 represents the last stage of PM: controlling the logic. The solution algorithm and the novel structured decision space are presented in details. In chapter 9 shows a case study, that epitomizes how PM managed to turn RMS into a real system and addresses its challenges. The component model of problem is the last part of the ROP analytics stage.

6.9 ROP: The Component Model

The components of ROP are depicted in Figure 6-15. The data model presented in this chapter is summed up in the Machinery & Product Makes component. This component encapsulates machinery (modules and configurations), workforce related data, and product makes data, section 6.4.1.1 can be reviewed for details. The Products component encapsulates all product manufacturing-independent data, section 6.4.1.2. The modeller defines the logic and ships it into the optimizer to get the best-synchronized system plans. At the end of chapter 8, the solution master algorithm will show the flow of logic among these component to create different alternatives of system plans and choosing the best of them as an output of the ROP.

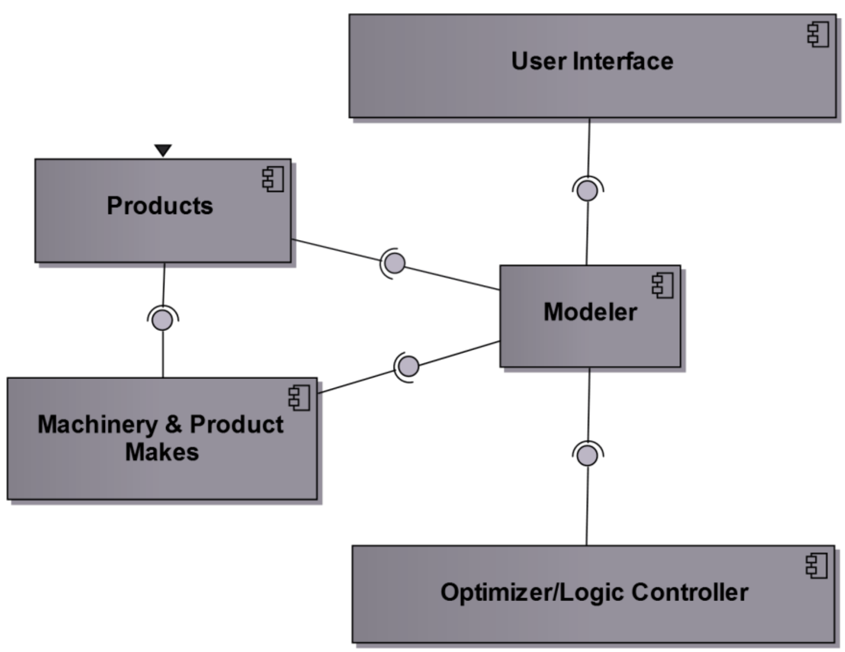


Figure 6-15: ROP Component Model

6.10 Summary

In this chapter, the Reconfiguration and Operations Planning problem was presented for the first time. A conceptual framework and data model were presented to set the foundations for further problem analysis and modeling. Some foundations, related issues, and basic planning definitions were also presented. The chapter concluded with the component model of ROP. In the next chapter, the logic that governs the reconfiguration and operations planning problem will be presented. More foundations will be elaborated whenever necessary in the upcoming three chapters.

Chapter 7 RECONFIGURATION AND OPERATIONS PLANNING PROBLEM: THE MATHEMATICAL STATEMENT

7.1 Introduction

Based on many foundations and analyses presented in chapter 6, the ROP problem was defined and the ROP component model culminated the first step of PM. The next step is to develop the logic that governs. Since the underlying system structure may be changed, modeling manufacturing operations in an RMS environment is different from their counterparts in other traditional operations in traditional manufacturing environments. Optimizing RMS operations should come second if changing the system structure could make the difference, i.e. swinging the capacity lever. In order to reconfigure an RMS system, market awareness of the operations that can be made by an RMS system to respond to its market demand is needed. A system reconfiguration operation might be omitted by just thinking of swinging one of its immediate levers—Inventory, overtime, subcontracting. In this chapter, a new mathematical model of reconfigurations and operation planning problem is presented. The model developed brings an assortment of mathematical models that is dependent on each other and materializes the foundations presented in chapter 6.

A novel advanced notation and tuplized nomenclature are presented in this chapter. The nomenclature is deployed over many tuples and every tuple define a collection of related data pieces (variables). Every tuple is listed in a separate table in order to make the ROP mathematical modeling better structured and more eloquent. The ROP modeling begins with discussing the system configuration modeling. Afterwards, the product operations modeling starts: products can be loaded (setup), produced either during regular or overtime hours or both, and then unloaded. Products manufacturing operations are concerned with sequencing operations and allocating time available among them. By analyzing demand and comparing it to the available supply, subcontracting decisions can be made if applicable. By identifying the current configuration, workforce levels could be adjusted at planning bucket boundaries. When demand and supply gets out of balance inventory/back orders build up and that should be under control. The objective of ROP is to put all these levers in tandem: the objective statement could include maximize the system profit; minimize inventories, magnify system responsiveness with best options, satisfy customers, and optimize system resources.

This chapter is organized as follows: the nomenclature is presented first in its newly developed tuplized format. Decision variables are described before discussing the ROP mini-problems and developing their mini-models. The assembly of these mini-models will lead to the definition of the concept of mathematical and objective statements that can describe the new system models that PM can define. The chapter will conclude by listing a concise ROP mathematical statement that represents the logic that governs the ROP problem.

7.2 Nomenclature

7.2.1 Advanced Notation

Before listing the nomenclature, the concept of advanced notation will be presented first. ROP is a compilation of many problems as already implied in chapter6. The problem has an enormous number of dynamic and interwoven variables and parameters. Letters and Greek symbols are neither enough for defining these data items nor enough for making them easy to grasp. Tuplizing the nomenclature has been developed as a solution, see Figure 7-1. Another problem that popped out while developing the ROP math model was the use of configurations or products that could be obsolete with time or can be added on the fly; this is considered a new problem characteristic to the operations management field. The concept of ID resolved that problem. ROP uses three basic indices for most variables to distinguish them: Product ID, Configuration ID, and Time Index. In addition, a basic symbol or acronym that could be shared among many variables and their distinguishing verbal acronyms are used to complete variables advanced notation symbols. Figure 7-2 gives an example.

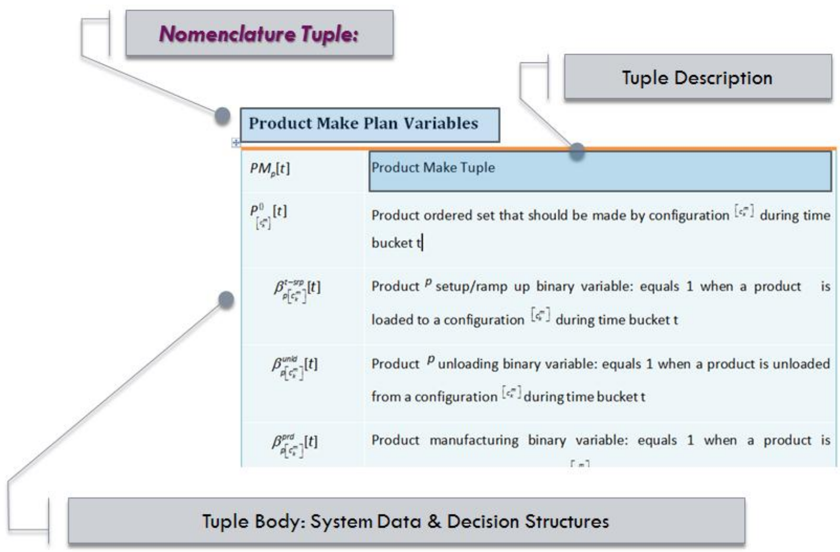


Figure 7-1: Tuplized Nomenclature

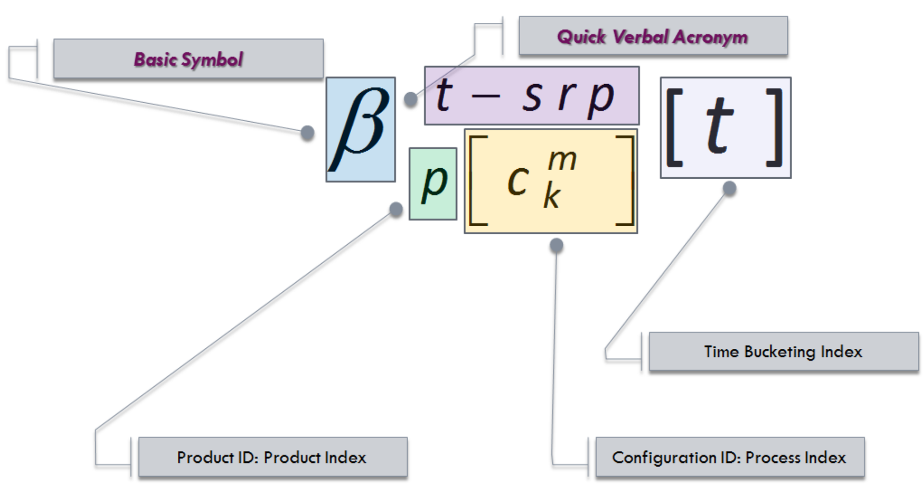


Figure 7-2: Advanced Notation Legend

7.2.2 Nomenclature Tuples

Numbers and IDS

p	Product ID, P101, P102, P103 etc.
m	Module ID, M1000, 2000, 3000 etc.
c_k^m	Configuration k loaded to module m ID, C1101,C2101, C3101 etc.
N_b	Number of planning buckets

Product Demand: Mix and Volumes

$D_p[t]$	Product *p* demand during a planning bucket *t*
N_p	Number of demanded products
P	a set of all product IDs that a manufacturing firm can make or supply to its markets i.e. product mix, $p \in P$
Pr_p	Price of product *p*

Product Supply: Parameters

PS_p	Product Supply Tuple
PS_p^{C-mtr}	Material cost of product p (\$/unit)
PS_p^{C-hld}	Holding cost of product p (\$/unit)
$PS_p^{C-sbcntrc}$	Subcontracting cost of product p (\$/unit)
$PS_p^{C-bkord}$	Backorder cost of product p (\$/unit)

Product Make Parameters

$PM_{p\left[c_k^m\right]}$	Configuration dependent products make tuple. It encompasses all product manufacturing dependent data.
$PM_{p\left[c_k^m\right]}^{t-srp}$	Set up/Ramp-up time of product p loaded to configuration c_k^m
$PM_{p\left[c_k^m\right]}^{t-unld}$	Unloading time of product p unloaded from configuration c_k^m
$PM_{p\left[c_k^m\right]}^{C-srp}$	Setup cost of product p loaded to configuration c_k^m
$PM_{p\left[c_k^m\right]}^{C-unld}$	Unloading cost of product p unloaded from configuration c_k^m
$PM_{p\left[c_k^m\right]}^{t-cycle}$	Cycle time of product p loaded to configuration c_k^m
$PM_{p\left[c_k^m\right]}^{thrpt}$	Throughput of product p loaded to configuration c_k^m
$PM_{p\left[c_k^m\right]}^{VC}$	Variable cost of product p loaded to configuration c_k^m (Fixed costs are already included in set/ramp ups and unloading costs)

Product Make Plan Variables

$PM_p[t]$	Product Make Tuple
$P_{\left[c_k^m\right]}^{()}[t]$	Product ordered set that should be made by configuration c_k^m during time bucket t
$\beta_{p\left[c_k^m\right]}^{t-srp}[t]$	Product p setup/ramp up binary variable: equals 1 when a product is loaded to a configuration c_k^m during time bucket t
$\beta_{p\left[c_k^m\right]}^{unld}[t]$	Product p unloading binary variable: equals 1 when a product is unloaded from a configuration c_k^m during time bucket t
$\beta_{p\left[c_k^m\right]}^{prd}[t]$	Product manufacturing binary variable: equals 1 when a product is manufactured by configuration c_k^m during time bucket t
$PMB_{p\left[c_k^m\right]}^{t-srp}[t]$	Set/ramp up time of product p made by configuration c_k^m during time bucket t

$PMB_{p\left[c_k^m\right]}^{t-unld}[t]$	Unloading time of product p if it has been made by configuration c_k^m during time bucket t
$PMB_{p\left[c_k^m\right]}^{R}[t]$	Regular quantity of product p made by configuration c_k^m during time bucket t
$PMB_{p\left[c_k^m\right]}^{O}[t]$	Overtime quantity of product p made by configuration c_k^m during time bucket t
$PMB_{p\left[c_k^m\right]}^{O\max}[t]$	Maximum overtime quantity of product p made by configuration c_k^m during time bucket t
$PMB_{p\left[c_k^m\right]}^{t-R}[t]$	Regular time allocated to produce product p during a planning bucket t when it is loaded to configuration c_k^m
$PMB_{p\left[c_k^m\right]}^{t-O}[t]$	Over time allocated to produce product p during a planning bucket t when it is loaded to configuration c_k^m

Product Supply: Mix and Volumes

$PSB_p[t]$	Product supply bucket: a tuple of product supply mix and volumes during a certain bucket t
$PSB_p^R[t]$	Total regular time supply (volume) of product p during time bucket t
$PSB_p^O[t]$	Total overtime supply (volume) of product p during time bucket t
$PSB_p^S[t]$	Total subcontracted quantity (if subcontracting is available) p during time bucket t
$PSB_p^I[t]$	Total Inventory quantity of product p during time bucket t
$PSB_p^B[t]$	Total backordering quantity of product p during time bucket t
$PSB_p^I[0]$	Initial Inventory quantity of product p prior to the current planning session
$PSB_p^B[0]$	Initial backordering quantity of product p prior to the current planning session

Module Workforce Plan

$W_m[t]$	Module Workforce bucket: a tuple of (Workforce level and their dependent hiring and firing values)
$W_m[t]$	Workforce level at module m during period *t*
$H_m[t]$	Workers hired at module m during period *t*
$F_m[t]$	Workforce fired at module m during period *t*

Configuration Parameters

$G_{\left[c_k^m\right]}$	Configuration tuple that represents all the data related to a certain configuration k loaded to a certain module m
$G^{W}_{\left[c_k^m\right]}$	Work force level attached to configuration c_k^m
$G^{t-Rcn}_{\left[c_q^m\right]\left[c_k^m\right]}$	Reconfiguration time from configuration *q* to configuration *k*
$G^{VC-Rcn}_{\left[c_q^m\right]\left[c_k^m\right]}$	Reconfiguration variable cost from configuration *q* to configuration *k*
$G^{FC-Rcn}_{\left[c_q^m\right]\left[c_k^m\right]}$	Reconfiguration fixed cost from configuration *q* to configuration *k*
$P_{\left[c_q^m\right]}$	Configuration product set that can be made while configuration c_k^m is loaded

Module Related Data

W_m	Module work force tuple
C_F^m	Work force firing cost of a worker has a skill-set standard needed by module *m*
C_H^m	Work force Hiring cost of a worker has a skill-set standard needed by module *m*
C_W^m	Regular work force hourly rate ($/hr)

C_o^m	Overtime hourly rate ($/hr)
CP_m	Configuration path of module m: N_b-tuple of configurations c_k^m indexed by bucket order $(1,2,3,...,N_b)$
$\beta_{CP^m}^{rcnfg}[t]$	Reconfiguration binary variable of configuration $CP_m[t]$
$\tau_{CP^m}^{rcnfg}[t]$	Reconfiguration time consumed to load configuration $CP_m[t]$ during time bucket *t*
$C_{CP^m}^{rcnfg}[t]$	Reconfiguration cost of module m during time bucket *t*
$C_{CP^m}^{srp}$	Setup and ramp costs of configuration path *m*
$C_{CP^m}^{unld}$	Unloading costs of configuration path *m*
$C_{CP^m}^{prd}$	Operation costs of configuration path *m*
$C_{CP^m}^{rcnfg}[t]$	Reconfiguration cost of module *m* during time bucket *t*
CMap	Configuration Map: Nm-tuple of configuration paths (CP_m s)

System Work Regulations

SWR	System Working Regulation Tuple (working days/month, hrs/shift, shift/day etc.)
$SWR^{WD}[t]$	Working Nb-Tuple: a sequence of period-work days, indexed by bucket number or order (1,2,..)
$SWR^{WH}[t]$	Working Nb-Tuple: a sequence of period-work days, indexed by bucket number or order (1,2,..)
$SWR^{OT}[t]$	Maximum overtime hours allowed per day: Nb-tuple indexed by bucket number or order
$SWR^{h/s}$	Number of working hours per shift
$SWR^{s/d}$	Number shifts per day

System Initial and Final States

B_{p0}	Outstanding back orders of product p at start of planning horizon (units)
I_0	Inventory of product p at start of planning horizon (units)
W_0	Workforce at the at the start of the planning horizon (man-day)
A_{kp}	Pre-planning system state constants, *k=1,2,…* of products *p*
$\mathfrak{M}_{kp}$	End-of-Planning System desired constants, *k=1,2,..* state of product *p*
$_{[^m C_k]}[0]$	Initial Configuration k loaded to module *m* (i.e. bucket 0 configuration)
$_{[^m C_k]p}[0]$	last product *p* loaded to configuration *k* loaded to module *m* (i.e. bucket 0 last product)

7.3 ROP Sub-Models

7.3.1 Reconfiguration Modeling

Spicer and Carlo (Spicer and Carlo 2007) presented a practical cost model to compute the reconfiguration cost between two scalable-RMS configurations. They considered their model the first to evaluate the reconfiguration costs. Their reconfiguration cost includes the cost of physical arrangement (labour cost) and the cost of lost capacity during system reconfiguration and ramp-up. In order to escape from the difficulty associated with reconfigurations costs and time estimations, Youssef and ElMaraghy (2006) and Son et al (Son 2000) developed other metrics to drive the configuration selection process. The work reported linked the demand to the system design process. Similar to Spicer and Carlo (Spicer and Carlo 2007), The system under study was highly scalable system. Liu et al (2006) introduced another cost-effective reconfiguration planning for multi-module-multi-product RMS's that best reflects the market demand changes. They defined the reconfiguration planning problem as the best reallocation of part families to production modules, and the best rebalancing of the whole system and each individual module to achieve minimum related cost and simultaneously satisfy the market demand.

The ROP first considers what is called configuration granularity. As already described in the data model, every module has its own configuration library. Developing new configurations is the responsibility of system designers, process planners, and any other concerned functional units or personnel. From ROP perspective, configurations are there and optimized to be ready to choose from in order to respond to market changes to match the fluctuating demand, either mix or volume or both, with the best mix of available resources. Configuration design is not a parameter in the configuration selection process. Throughput and product mix that can be made are the leading key parameters that make a certain configuration preferred over other configurations. Investment decisions, buying and selling modules/machines/material handling equipment, are system design issues and are irrelevant to operational decisions. Since the ROP encompass all production activities in an RMS environment within one model as will be shown later, there is no need to think of system's reconfiguration lost capacity or how it could be calculated. When system reconfiguration takes place, their associated costs add up to the other operation costs and the time consumed is subtracted from the time available for other operations. Therefore, any consequences of lost capacity are already taken care of regardless of the mix and the volumes of the products being produced. The new model for reconfiguration costs depends on classifying the reconfiguration costs as fixed costs and variable costs. The ramp up times and costs are product operations costs and will be discussed very shortly. The reconfiguration modeling is only concerned with the reconfiguration operations, timely decisions, time consumed, and costs incurred.

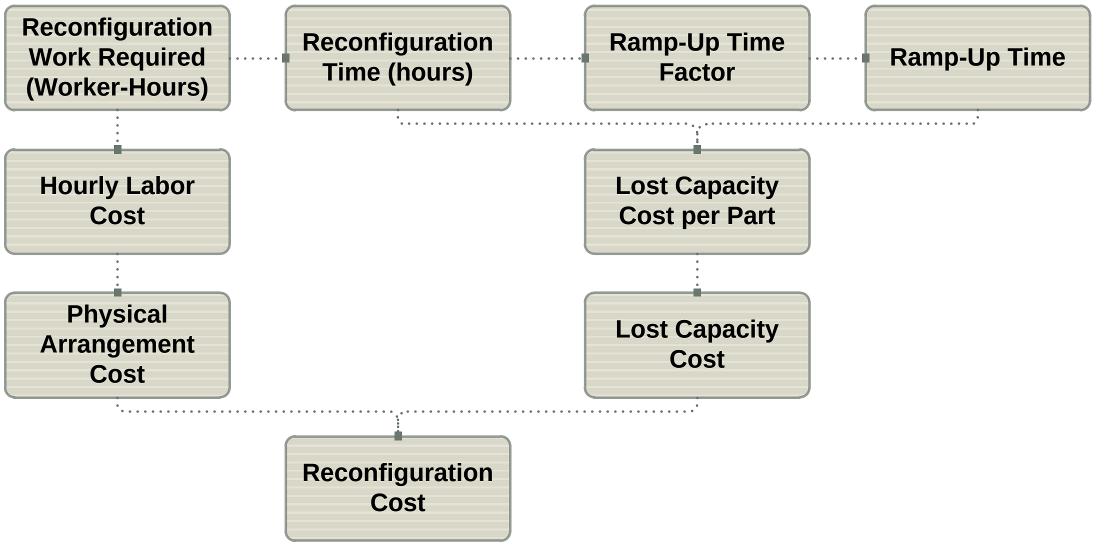


Figure 7-3: Reconfiguration Cost Structure Model Presented by Spicer and Carlo (2007)

In order to synchronize demand planning activities—assumed to be on a monthly basis—with reconfiguration planning process, the reconfiguration decisions are supposed to be taken at the beginning of demand planning buckets. A decision to maintain a configuration or replace it with another is determined by the binary $\beta_{CP^m}^{rcnfg}[t]$. It has a value of zero if the current configuration is the same one allocated to previous planning bucket; otherwise, it has the value 1 (Equation (7.1))

As already reported in chapter 6, an array (ordered set or tuple) of configuration IDs assigned defines a module configuration path. If the reconfigurable manufacturing system is composed of M modules, the set of configuration paths defines the System Configuration Map. Once configuration binaries are defined, reconfiguration times and costs can be evaluated using equation (7.2) and(7.3). According to previous and current configuration IDs, the corresponding time should be extracted from configuration data files.

The system reconfiguration total duration and total costs are equivalent to the total reconfiguration time and costs of its paths respectively. The reconfiguration cost has two main components fixed and variable. The fixed costs include any fixed costs regardless of the time consumed during the configuration process. If there is a module/machine/piece of equipment are only bought for certain configuration and resold once the configuration is unloaded, this should be considered a fixed cost; otherwise this should considered a capital budgeting decision and its context is anywhere else other than the ROP. The aforementioned rule is a basic cost accounting principal which unfortunately ignored by many RMS researchers. Adding a configuration to a module/system library is an investment decision and it should be evaluated during the system design/development process. Hiring costs of a quality specialist to supervise the reconfiguration process could add to reconfiguration cost. Reconfiguration variable costs are any cost that could be estimated on hourly basis, moving equipment might be leased based on hourly basis; a reconfiguration crew who might be helping the operating workforce might be hired on hourly basis. All the reconfiguration costs parameters are evaluated during the design process of any new configuration. Once done, the ROP is only concerned with what is fixed and what is variable. Operating labour force wages are estimated in another place described later.

Function templates are a great tool to define the low level of cost structure even though it is not applied here explicitly. The details are a configuration design concern as described earlier. The ramp up costs and unloading costs are another two operations that will be discussed in a later part of the model because both are product dependent.

$$\beta_{CP^m}^{rcnfg}[t] = \begin{cases} 0 & CP^m[t] = CP^m[t-1] \\ 1 & CP^m[t] \neq CP^m[t-1] \end{cases} \tag{7.1}$$

$$\tau_{CP^m}^{rcnfg}[t] = \begin{cases} G_{CP^m[t-1]CP^m[t]}^{t-Rcn} & \beta_{CP^m}^{rcnfg}[t] = 1 \\ 0 & otherwise \end{cases} \tag{7.2}$$

$$C_{CP^m}^{rcnfg}[t]=\beta_{CP^m}[t]\left(G_{[c_q^m][c_k^m]}^{VC-Rcn}G_{[c_q^m][c_k^m]}^{t-Rcn}+G_{[c_q^m][c_k^m]}^{FC-Rcn}\right),\ CP^m[t]=\left[c_q^m\right]\&CP^m[t-1]=\left[c_k^m\right] \quad \textbf{(7.3)}$$

$$C^{reconfiguration}=\sum_{m=1}^{N_M}\sum_{t=1}^{N_B}C_{CP^m}^{rcnfg}[t] \quad \textbf{(7.4)}$$

7.3.2 Product Make Modeling

7.3.2.1 Product Make Sequencing Decisions

Product make modeling is the next responsibility of the ROP problem. Every configuration defines its own product space. According to demand mix, a subset of such product space may be produced. The product make set is the intersection of product demand set (demand mix) and configuration product list, Equation(7.5). Once the product make set is identified, the product sequence could be chosen. Equation (7.6) defines the bucket product sequence. The size of product tuple could be of size 1 or include the entire elements of product make set, inequalities (7.7) and(7.8).

$$P_{[c_k^m]}^{\{\}}[t]=P\cap P_{[c_k^m]} \quad \textbf{(7.5)}$$

$$P_{[c_k^m]}^{()}[t]=\left(p:p\in P_{[c_k^m]}^{\{\}}[t]\right) \quad \textbf{(7.6)}$$

$$\left|P_{[c_k^m]}^{()}[t]\right|\leq\left|P_{[c_k^m]}^{\{\}}[t]\right| \quad \textbf{(7.7)}$$

$$\left|P_{[c_k^m]}^{()}[t]\right|\geq 1 \quad \textbf{(7.8)}$$

7.3.2.2 Product Set up Decisions

Once the product make sequences are identified, the next step is to determine the setup/ramp up decisions. In this study, the ramp up and set up terms are used interchangeably. Actually, the term setup should be preferred in planning contexts. A ramp up process could be considered a product development/system installation process. It could happen once during the configuration/product development cycles. The planning horizon should not include such periods. ROP is concerned with the steady state configuration/product loading and unloading processes. For every product belongs to the current configuration and within the current planning bucket there is a binary set up decision as described in the newly developed hierarchical binary (7.9). The value of set up binary is determined after checking many hierarchical preconditions. These conditions can be described verbally as follows and mathematically as shown in equation(7.9).

For every product belongs to the loaded configuration during the current bucket:

Rule 1: if a product p is not among the product tuples sequence, the set up binary value is assigned the value of zero.

Rule 2: if that product is not the first product in the product tuples sequence, the set up binary value is assigned the value of 1.

Rule 3: if the current planning bucket configuration is not the same of previous bucket, i.e. reconfiguration took place; the set up binary value is assigned the value of 1.

Rule 4: if the last product of previous bucket is not the same one of the first product of the current configuration, the setup value is assigned 1; otherwise, it is assigned the value zero—no setup is needed.

Rules 1-4 are recursively mutually exclusive rules. For example, rule 2 does not apply unless rule 1 fails and so on. Equation (7.9) is called a hierarchical binary equation. Recursive mutually exclusive rules and hierarchical binaries are major contributions to RMS science and are considered some of the new contributions of PM at the mathematical modeling part.

$$\exists p \in P_{\left[c_k^m\right]} :$$

$$\beta^{srp}_{p\left[c_k^m\right]}[t] = \begin{cases} 0 & p \notin P^{()}_{\left[c_k^m\right]}[t] \\ \begin{cases} 1 & p \neq First(P^{()}_{\left[c_k^m\right]}[t]) \\ \begin{cases} 1 & c_k^m \neq CP^m[t-1] \\ \begin{cases} 1 & p \neq Last(P^{()}_{\left[c_k^m\right]}[t-1]) \\ 0 & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} \tag{7.9}$$

The product set up process in an RMS is different from its peers of traditional manufacturing systems. The underlying system structure is changeable. For example, considering the scalable-RMS with single product, the product can be produced using many configurations. Once a reconfiguration is identified, the setup (ramp up) process begins; every configuration defines its setup time and cost associated. Once the setup binaries are identified, the set up times and costs can be evaluated using equations (7.10) and (7.11) respectively. The process is iterated for all product make sequences and for all planning buckets. The set up cost for every configuration path (module) is evaluated using equation (7.12) and for the whole system using equation(7.13).

$$\exists p \in P_{\left[c_k^m\right]} : \qquad PMB^{t-srp}_{p\left[c_k^m\right]}[t] = \beta^{srp}_{p\left[c_k^m\right]}[t] \times PM^{t-srp}_{p\left[c_k^m\right]} \tag{7.10}$$

$$\exists p \in P_{\left[c_k^m\right]} : \qquad PMB^{C-srp}_{p\left[c_k^m\right]}[t] = \beta^{srp}_{p\left[c_k^m\right]}[t] \times PM^{C-srp}_{p\left[c_k^m\right]} \tag{7.11}$$

$$C^{srp}_{CP^m} = \sum_{t=1}^{N_b} \sum_{\forall p \in P_{\left[c_k^m\right]}} PMB^{C-srp}_{p\left[c_k^m\right]}[t] \tag{7.12}$$

$$C^{Set/RampUp} = \sum_{m=1}^{N_m} C^{srp}_{CP^m} \tag{7.13}$$

7.3.2.3 Product Make Operations Modeling

ROP treats RMS as a typical manufacturing system, i.e. has a regular time and overtime operations. As a result, the product make operations are assumed to include both regular and overtime operations. For every planning bucket and for every product decided to be among the product tuples sequence, there is another production binary associated as described by equation (7.14). Once a product is decided to be among the planning bucket product make tuples sequence, the value of its binary should be equal 1. More elaborations will be described later in sections (7.3.2.5 & 7.3.2.6)

$$\exists p \in P_{\left[c_k^m\right]} : \qquad \beta^{prd}_{p[CP^m]}[t] = \begin{cases} 0 & p \notin P^{()}_{\left[c_k^m\right]}[t] \\ 1 & otherwise \end{cases} \tag{7.14}$$

7.3.2.4 Product Unloading Decisions

Product unloading decisions are introduced to decouple the product unloading operations from their cousins, i.e. system reconfigurations and product setups counterparts. During reconfiguration process, there is a product loaded needs to be unloaded first. Within a multi-product module, there are unloading operations that takes place during products changeovers as well. Separating the unloading process make the set up times, reconfiguration times, and unloading times all sequence independent. Similar to set up decisions, product unloading decisions are also defined using hierarchical binaries. The last product of the previous bucket is the key product in the decision hierarchy chain. For all product set elements of the current loaded configuration, the following rules are the ones that govern the unloading process:

Rule 1: If a product p is not among the current bucket product sequence, the unloading binary value is assigned the value of zero.

Rule 2: if that product is not the last product in the product sequence, the unloading binary value is assigned the value of 1; all the products other than the last must be unloaded.

Rule 3: if the current planning bucket is the last one, the product is kept on the system, i.e. its unloading binary has the value zero.

Rule 4: if the first product of the next bucket is not the same one of the last product of the current configuration, the unloading value is assigned the value of 1.

Rule 5: if the next bucket configuration is not the same as the current configuration, i.e. a reconfiguration takes place, the product has to be unloaded first. Equation (7.15) shows the mathematical representation for product unloading decision. Similar to set up binaries, the unloading ones are recursively mutually exclusive. For every applicable product, once a rule is fired a value is assigned to the unloading binary and the process terminates.

$$\exists p \in P_{[c_k^m]}:$$

$$\beta^{unld}_{p[c_k^m]}[t] = \begin{cases} 0 & p \notin P^{()}_{[c_k^m]}[t] \\ \begin{cases} 1 & p \neq Last(P^{()}_{[c_k^m]}[t]) \\ \begin{cases} 0 & t = N_b \\ \begin{cases} 1 & p \neq First(P^{()}_{[c_k^m]}[t+1]) \\ \begin{cases} 1 & CP^m[t] \neq CP^m[t+1] \\ 0 & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} \tag{7.15}$$

After deciding on the unloading operations, the time consumed for every product and the unloading associated cost can be evaluated. Equation (7.16) shows how the unloading time is evaluated for every bucket t. Equation (7.17) shows how the unloading cost can be estimated as well. The cost of all the unloading operations is evaluated for every production module by equation (7.17) and the cost of all the unloading operations throughout the system is estimated using equation(7.19).

$$\exists p \in P_{\left[c_k^m\right]}: \qquad PMB_{p\left[c_k^m\right]}^{t-unld}[t] = \beta_{p\left[c_k^m\right]}^{unld}[t] \times PM_{p\left[c_k^m\right]}^{t-unld} \tag{7.16}$$

$$\exists p \in P_{\left[c_k^m\right]}: \qquad PMB_{p\left[c_k^m\right]}^{C-unld}[t] = \beta_{p\left[c_k^m\right]}^{unld}[t] \times PM_{p\left[c_k^m\right]}^{C-unld} \tag{7.17}$$

$$C_{CP^m}^{unld} = \sum_{t=1}^{N_b} \sum_{\forall p \in P_{\left[c_k^m\right]}} PMB_{p\left[c_k^m\right]}^{C-unld}[t] \tag{7.18}$$

$$C^{Unloading} = \sum_{m=1}^{N_m} C_{CP^m}^{unld} \tag{7.19}$$

7.3.2.5 Regular Time Operations Scheduling

Regular time production operation scheduling is concerned with scheduling all the manufacturing operations in an RMS environment. So far, reconfiguration, setup, and unloading operations are discussed; all these operations are both configuration and product dependent. Every planning bucket has a number of working hours that are available for all manufacturing operations. A realistic calendar is consulted to evaluate the number of hours that are available for these manufacturing operations. Equation (7.20) shows the number of hours available during planning bucket t. some portions of the planning bucket time is consumed for preparing the system for production operations: these portions of time would be non-productive times; costs are incurred without having an immediate output. Earlier researchers referred to this time as the lost capacity time (Spicer and Carlo 2007). Only products listed in the product make sequence can have time slots for manufacturing purposes available after taking into consideration other time slices deducted for other system preparation operations. Equation (7.21) exclude all those products that do not belong to the current product sequence list from having a time slot. The planning bucket time available for product sequence is constrained by bucket time length and non-production operations. Constraint(s) (7.22) shows the time available for production operations after subtracting all other non-production operations time slots or slices. Equation (7.23) shows how the throughput is calculated using the corresponding cycle time. Using the throughput and the time slot constrained by equation(7.22), the regular time product volumes are estimated using equation(7.24). Once individual product volumes are identified, regular time production costs can be evaluated using equation(7.25) for every path. Equation (7.26) evaluates the production costs for the whole system for both the regular and overtime quantities produced. The overtime decision variables will be discussed immediately in the next section.

$$SWR^{WH}[t] = SWR^{wD}[t] \times SWR^{h/s} \times SWR^{s/d} \tag{7.20}$$

$$\exists p \in \left(P_{\left[c_k^m\right]} - P^{\{\}}_{\left[c_k^m\right]} \right): \qquad PMB^{t-R}_{p\left[c_k^m\right]}[t] = 0 \tag{7.21}$$

$$\sum_{\forall p \in P^{()}_{\left[c_k^m\right]}} PMB^{t-R}_{p\left[c_k^m\right]}[t] = DT[t] - \tau^{rcnfg}_{CP^m}[t] - \sum_{\forall p \in P_{\left[c_k^m\right]}} PMB^{t-srp}_{p\left[c_k^m\right]}[t] - \sum_{\forall p \in P_{\left[c_k^m\right]}} PMB^{t-unld}_{p\left[c_k^m\right]}[t] \tag{7.22}$$

$$PM^{thrpt}_{p\left[c_k^m\right]} = \frac{1}{PM^{t-cycle}_{p\left[c_k^m\right]}} \tag{7.23}$$

$$\exists p \in P_{\left[c_k^m\right]} : \quad PMB^{R}_{p\left[c_k^m\right]}[t] = PMB^{t-R}_{p\left[c_k^m\right]}[t] \times PM^{thrpt}_{p\left[c_k^m\right]} \tag{7.24}$$

$$C^{prd}_{CP^m}[t] = \sum_{\forall p \in P^{()}_{\left[c_k^m\right]}} PM^{VC}_{p\left[c_k^m\right]} \left(PMB^{R}_{p\left[c_k^m\right]}[t] + PMB^{O}_{p\left[c_k^m\right]}[t] \right) \tag{7.25}$$

$$C^{prd}_{CP^m} = \sum_{t=1}^{N_B} \left(PS^{C-mtr}_{p} + PM^{VC}_{p\left[c_k^m\right]} \right) \left(PMB^{R}_{p\left[c_k^m\right]}[t] + PMB^{O}_{p\left[c_k^m\right]}[t] \right) \tag{7.26}$$

7.3.2.6 Overtime Decisions and Scheduling

By checking the workforce regulation and bucket slice allocated for individual products, the maximum overtime for each product can be estimated as described by equation(7.27). Constraint (7.28) ensures that the total number of days assigned for overtime never exceeds the total limits, integrity constraint. Overtime is an integer value that occurs at the end of workdays. The cost of overtime production has been already embedded in equation(7.26).

$$\exists p \in P^{()}_{\left[c_k^m\right]}: \qquad PMB^{O\max}_{p\left[c_k^m\right]}[t] = Days\left(PMB^{t-R}_{p\left[c_k^m\right]}[t]\right) \times SWR^{OT}[t] \times PM^{thrpt}_{p\left[c_k^m\right]} \tag{7.27}$$

$$\sum_{\forall p \in P^{()}_{\left[c_k^m\right]}} Days\left(PMB^{t-R}_{p\left[c_k^m\right]}[t]\right) \leq DT[t] \tag{7.28}$$

$$\exists p \in P^{()}_{\left[c_k^m\right]}: \qquad PMB^{O}_{p\left[c_k^m\right]}[t] \leq PMB^{O\max}_{p\left[c_k^m\right]}[t] \tag{7.29}$$

7.3.3 Product Supply and Product Balance Equations

Since a product might be produced by more than one module, the total regular volume of a product p is calculated using equation(7.30). Total overtime is also estimated similarly using equation(7.31). Equation (7.32) evaluates the total product supply. In addition to regular and overtime quantities, some products might have substitutes and could be outsourced. Equation (7.33) balances demand, supply, and inventory.

$$PSB^{R}_{p}[t] = \sum_{\forall m} PMB^{R}_{pCP^m}[t] \tag{7.30}$$

$$PSB^{O}_{p}[t] = \sum_{\forall m} PMB^{O}_{pCP^m}[t] \tag{7.31}$$

$$\exists p \in P: \quad PSB_p[t] = PSB^{R}_{p}[t] + PSB^{O}_{p}[t] + PSB^{S}_{p}[t] \tag{7.32}$$

$$\exists p \in P: \quad D_p[t] + B_p[t] - B_p[t-1] = I_p[t-1] - I_p[t] + PSB_p[t] \tag{7.33}$$

7.3.3.1 System Envelop Constraints

System envelop constraints are concerned with the balanced state of the system over its planning horizon. Each product has an initial state of inventory/backordering and could have target values of both of them at the end of the planning horizon. In other words, inventory and backorders are product state variables. Equations (7.34) to (7.37) define product initial and target state variables. Equation (7.38) defines system envelop constraints for all products. System envelop constraints are highly valuable in chaining the decision space, which is a novel concept presented by PM to accelerate the solution process. Constraints (7.36) to (7.38) are all soft constraints.

$$B_p[0] = A_{1i} \qquad \forall Product\ p \in P \tag{7.34}$$

$$I_p[0] = A_{2i} \qquad \forall Product\ p \in P \tag{7.35}$$

$$B_p[N_p] = \mathfrak{M}_{1i} \qquad \forall Product\ p \in P \tag{7.36}$$

$$I_p[N_p] = \mathfrak{M}_{1i} \qquad \forall Product\ p \in P \tag{7.37}$$

$$\exists p \in P: \quad \sum_{t=1}^{t=N_b} PSB_p[t] + \left\{ I_p[0] - B_p[0] \right\} + \left\{ I_p[N_B] - B_p[N_B] \right\} = \sum_{t=1}^{t=N_b} D_p[t] \tag{7.38}$$

7.3.4 Workforce Modeling

The workforce and its relation to reconfigurable manufacturing is one of the most ignored aspects in the RMS literature. Reconfiguration process, especially in scalable RMS systems, should be accompanied by workforce adjustments. A module with capacities X, 1.3X, 1.7X, for example, cannot its operations using the same workforce size. The module workforce is supposed to be participating in the reconfiguration and executing ramp up/setup, manufacturing, and unloading activities. If there is additional workforce involved in the reconfiguration process, they should be modelled as part of fixed or variable costs associated with the reconfiguration modeling part. Equation(7.39) defines the workforce state of module m during bucket time t. Workforce levels are constant as long as a configuration g is maintained at the system. Equation (7.39) shows how the modular workforce variable is estimated. Equation (7.40) represents the workforce balance among buckets. Equation (7.41) shows the mutual exclusive relation between hiring and firing variables. Equation (7.42) evaluates the workforce level at the system level. Equation (7.43) represents the workforce balance at the system level. Equation (7.44) estimates the payroll and hiring and firing costs.

$$W_m[t] = G^W_{\left[c_k^m\right]}, \quad CP^m[t] = c_k^m \qquad \forall \text{module} \, m, \forall \text{bucket} \, t \tag{7.39}$$

$$W_m[t] = W_m[t-1] + H_m[t] - F_m[t] \qquad \forall \text{module} \, m, \forall \text{bucket} \, t \tag{7.40}$$

$$H_m[t] \times F_m[t] = 0 \qquad \forall \text{module} \, m, \forall \text{bucket} \, t \tag{7.41}$$

$$W[t] = \sum_{m=1}^{N_m} W_m[t] \qquad \forall bucket \, t \tag{7.42}$$

$$W[t] = W[t-1] + H[t] - F[t] \tag{7.43}$$

$$C^{WorkForce} = \sum_{m=1}^{N_M} \sum_{t=1}^{N_B} C_H^m H^m[t] + C_F^m F^m[t] + C_W^m W_m[t] SWR^{H/B}[t] \tag{7.44}$$

7.3.5 System level Cost Estimation

Equation (7.45) estimates all the costs related to machining activities. Equation (7.46) evaluates the inventory holding costs. Equation (7.47) represents the backordering cost. Since backordering cost is intangible one, it is separated as an individual objective. Backordering cost represents a good measure of the system-market relationship and customer satisfaction level. Equation (7.48) represents the subcontracting costs. Equation (7.49) estimates the total costs and Equation (7.50) represents the system revenue. The rule the governs the revenue term is that we cannot sell more than our demand, market constraint, and we cannot sell more than we can produce, i.e. system capacity constraint. Equation (7.51) shows the profitability of operations. Other system fixed costs such as building, hydro, white-collar workforce etc. are irrelevant to the operation planning costs. A profitability performance indicator is necessary in an RMS environment. As the ROP model reveals, the cost structure in an RMS environment is very complicated. The same product can be produced using many configurations with different workforce sizes. Not all direct costs are supposed to be fixed as long as there is a system reconfiguration. A profitability measure must be used instead of the traditional cost minimization for that reason.

$$C^{Machining} = C^{Unloading} + C^{Set/RampUp} + C^{prd} + C^{Overtime} + C^{reconfiguration} \tag{7.45}$$

$$C^{holding} = \sum_{\forall p \in P} \sum_{t=1}^{N_B} C_h I_p[t] \tag{7.46}$$

$$C^{Backorders} = \sum_{\forall p \in P} \sum_{t=1}^{N_B} C_b I_p[t] \tag{7.47}$$

$$C^{Subcontracting} == \sum_{\forall p \in P} \sum_{t=1}^{N_B} C_S S_p[t] \tag{7.48}$$

$$C^{Total\,Costs} = C^{Machining} + C^{holding} + C^{Subcontracting} + C^{WorkForce} \tag{7.49}$$

$$R^{revenue} = \sum_{\forall p \in P} \rho_p^{price} Min\{D_p[t], PSB_p[t]\} \tag{7.50}$$

$$P^{profit} = R^{revenue} - C^{Total\,Cost} \tag{7.51}$$

7.4 The ROP Mathematical Statement

When addressing large-scale problems or systems like the ROP, the classical "mathematical model" notion needs to be expanded. In this context, PM introduces a novel notion to replace mathematical models, mathematical statements. In order to make the logic that governs more smooth-tongued, it was presented earlier in a fragmented format (mini-mathematical models). The following listing summarizes the ROP novel mathematical statement.

Nomenclature:

The nomenclature has already been presented at section 7.2.2.

Decision Variables/Structures:

a. System Configuration: Configuration Selection, Configuration Paths, and System Configuration Maps

b. Operations Schedules (the “Whens” and the “How longs”): reconfiguration, set/ramp ups, unloading, regular and overtime production, r-buckets and p-buckets.

c. Product volumes and mixes: which products to produce or outsourced and in what quantities: product supply curves

d. Inventory and Backorders: Inventory and back order curves

Objectives Statement:

Reconfiguration and Operation Planning problem is formulated to optimize a growing list of objectives whether implicitly or explicitly:

1. Maximize profitability; profitability is the supreme objective of any manufacturing enterprise.

2. Maximize responsiveness: from ROP perspective, responsiveness can be achieved by many levers: reconfiguration, inventories, overtime, and subcontracting.

3. Maximize system efficiency: minimize inventories, optimize configuration selection, max configuration up time (r-bucket duration), optimize product batching (product setups, unloading, and change over times), and max workforce utilization.

All these objectives are interwoven as value drivers of the reconfigurable manufacturing systems.

Templates:

$$Min\,/\,Max\ financials\ Z_1 = g_1(machinery) + g_2(workforce) + g_3(Inventory) + g_4(product) \qquad \textbf{(7.52)}$$

$$Min\ \text{Inventory Investments}\ Z_2 = g_2(I_{it}, C_{Ii}) \qquad \textbf{(7.53)}$$

$$Min\ \text{Backorders}\ Z_3 = g_3(B_{it}, p_i) \qquad \textbf{(7.54)}$$

Objective Implementation:

Max Operations Profitability

$$\sum_{\forall p \in P} \rho_p^{price} Min\{D_p[t], PSB_p[t]\} - \left[\underbrace{\left(\begin{array}{l} \underbrace{\sum_{m=1}^{N_m} \sum_{t=1}^{N_b} \sum_{\forall p \in P_{\left[c_k^m\right]}} PMB_{p\left[c_k^m\right]}^{C-unld}[t]}_{Unloading} \\ \underbrace{\sum_{m=1}^{N_M} \sum_{t=1}^{N_B} C_{CP^m}^{rcnfg}[t]}_{\text{Re}\, configuraion} \\ \underbrace{\sum_{m=1}^{N_m} \sum_{t=1}^{N_b} \sum_{\forall p \in P_{\left[c_k^m\right]}} PMB_{p\left[c_k^m\right]}^{C-srp}[t]}_{Set/Ramp\, up} \\ \underbrace{\sum_{t=1}^{N_B} \left(PS_p^{C-mtr} + PM_{p\left[c_k^m\right]}^{VC} \right) \left(PMB_{p\left[c_k^m\right]}^{R}[t] + PMB_{p\left[c_k^m\right]}^{O}[t] \right)}_{Machining\,\&\,Materials} \end{array} \right)}_{Manufacturing\, Operations\, Costs} + \underbrace{\left(\begin{array}{l} \underbrace{\sum_{m=1}^{N_M} \sum_{t=1}^{N_B} C_H^m H^m[t] + C_F^m F^m[t] + C_W^m W_m[t] SWR^{H/B}[t]}_{Hiring,\, Firing,\, and\, Payroll} \\ \underbrace{\sum_{\forall p \in P} \sum_{t=1}^{N_B} PS_p^{C-hld} I_p[t]}_{Inventory\, holding\, costs} \\ \underbrace{\sum_{\forall p \in P} \sum_{t=1}^{N_B} PS_p^{C-sbcntrc} PSB_p^S[t]}_{Subcontracting} \end{array} \right)}_{Other\, Costs} \right] \qquad \textbf{(7.55)}$$

Minimize capital investment in inventory

$$Min\,Z_2 = \frac{1}{N_b}\left\{\sum_{\forall p}\sum_{t=1}^{N_b} Pr_p \max\left\{I_p[t],0\right\}\right\} \tag{7.56}$$

Minimize backorder

$$Min\,Z_3 = \frac{1}{N_b}\left\{\sum_{\forall p}\sum_{t=1}^{N_b} PS_p^{C-bkord} \max\left\{-I_p[t],0\right\}\right\} \tag{7.57}$$

Constraints:

$$P^{\{\}}_{\left[c_k^m\right]}[t] = P \cap P_{\left[c_k^m\right]} \tag{7.58}$$

$$P^{()}_{\left[c_k^m\right]}[t] = \left(p : p \in P_{\left[c_k^m\right]}\right) \tag{7.59}$$

$$\left|P^{()}_{\left[c_k^m\right]}[t]\right| \le \left|P^{\{\}}_{\left[c_k^m\right]}[t]\right| \tag{7.60}$$

$$\left|P^{()}_{\left[c_k^m\right]}[t]\right| \ge 1 \tag{7.61}$$

$\exists p \in P_{\left[c_k^m\right]}$:

$$\beta^{srp}_{p\left[c_k^m\right]}[t] = \begin{cases} 0 & p \notin P^{()}_{\left[c_k^m\right]}[t] \\ \begin{cases} 1 & p \ne First(P^{()}_{\left[c_k^m\right]}[t]) \\ \begin{cases} 1 & c_k^m \ne CP^m[t-1] \\ \begin{cases} 1 & p \ne Last(P^{()}_{\left[c_k^m\right]}[t-1]) \\ 0 & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} \tag{7.62}$$

$$\exists p \in P_{\left[c_k^m\right]} : \qquad PMB^{t-srp}_{p\left[c_k^m\right]}[t] = \beta^{srp}_{p\left[c_k^m\right]}[t] \times PM^{t-srp}_{p\left[c_k^m\right]} \tag{7.63}$$

$$\exists p \in P_{\left[c_k^m\right]}:$$

$$\beta^{unld}_{p[c_k^m]}[t] = \begin{cases} 0 & p \notin P^{()}_{\left[c_k^m\right]}[t] \\ \begin{cases} 1 & p \neq Last(P^{()}_{\left[c_k^m\right]}[t]) \\ \begin{cases} 0 & t = N_b \\ \begin{cases} 1 & p \neq First(P^{()}_{\left[c_k^m\right]}[t+1]) \\ \begin{cases} 1 & CP^m[t] \neq CP^m[t+1] \\ 0 & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} \tag{7.64}$$

$$\exists p \in P_{\left[c_k^m\right]}: \qquad PMB^{t-unld}_{p\left[c_k^m\right]}[t] = \beta^{unld}_{p\left[c_k^m\right]}[t] \times PM^{t-unld}_{p\left[c_k^m\right]} \tag{7.65}$$

$$\exists p \in P_{\left[c_k^m\right]}: \qquad \beta^{prd}_{p[CP^m]}[t] = \begin{cases} 0 & p \notin P^{()}_{\left[c_k^m\right]}[t] \\ 1 & otherwise \end{cases} \tag{7.66}$$

$$SWR^{WH}[t] = SWR^{wD}[t] \times SWR^{h/s} \times SWR^{s/d} \tag{7.67}$$

$$\exists p \in \left(P^{\{\}}_{\left[c_k^m\right]} - P_{\left[c_k^m\right]}\right): \qquad PMB^{t-R}_{p\left[c_k^m\right]}[t] = 0 \tag{7.68}$$

$$\sum_{\forall p \in P^{()}_{\left[c_k^m\right]}} PMB^{t-R}_{p\left[c_k^m\right]}[t] = DT[t] - \tau^{rcnfg}_{CP^m}[t] - \sum_{\forall p \in P_{\left[c_k^m\right]}} PMB^{t-srp}_{p\left[c_k^m\right]}[t] - \sum_{\forall p \in P_{\left[c_k^m\right]}} PMB^{t-unld}_{p\left[c_k^m\right]}[t] \tag{7.69}$$

$$PM^{thrpt}_{p\left[c_k^m\right]} = \frac{1}{PM^{t-cycle}_{p\left[c_k^m\right]}} \tag{7.70}$$

$$\exists p \in P_{\left[c_k^m\right]}: \quad PMB^{R}_{p\left[c_k^m\right]}[t] = PMB^{t-R}_{p\left[c_k^m\right]}[t] \times PM^{thrpt}_{p\left[c_k^m\right]} \tag{7.71}$$

$$PSB^R_p[t] = \sum_{\forall p} PMB^R_{pCP^m}[t] \tag{7.72}$$

$$PSB^O_p[t] = \sum_{\forall p} PMB^O_{pCP^m}[t] \tag{7.73}$$

$$\exists p \in P: \quad PSB_p[t] = PSB_p^R[t] + PSB_p^O[t] + PSB_p^S[t] \tag{7.74}$$

$$\exists p \in P: \quad I_p[t] - B_p[t] + D_p[t] = I_p[t-1] - B_p[t-1] + PSB_p[t] \tag{7.75}$$

$$B_p[0] = A_{1i} \qquad \forall Product\ p \in P \tag{7.76}$$

$$I_p[0] = A_{2i} \qquad \forall Product\ p \in P \tag{7.77}$$

$$B_p[N_p] = \mathfrak{M}_{1i} \qquad \forall Product\ p \in P \tag{7.78}$$

$$I_p[N_p] = \mathfrak{M}_{1i} \qquad \forall Product\ p \in P \tag{7.79}$$

$$\exists p \in P: \quad \sum_{t=1}^{t=N_b} PSB_p[t] + \left\{ I_p[0] - B_p[0] \right\} + \left\{ I_p[N_B] - B_p[N_B] \right\} = \sum_{t=1}^{t=N_b} D_p[t] \tag{7.80}$$

$$W_m[t] = G^W_{\left[c_k^m\right]}, \quad CP^m[t] = c_k^m \qquad \forall mdoule\, m, \forall bucket\, t \tag{7.81}$$

$$W_m[t] = W_m[t-1] + H_m[t] - F_m[t] \qquad \forall mdoule\, m, \forall bucket\, t \tag{7.82}$$

$$H_m[t] \times F_m[t] = 0 \qquad \forall mdoule\, m, \forall bucket\, t \tag{7.83}$$

$$W[t] = \sum_{m=1}^{N_m} W_m[t] \qquad \forall bucket\, t \tag{7.84}$$

$$W[t] = W[t-1] + H[t] - F[t] \tag{7.85}$$

Provided the time for any operation to be evaluated on hourly basis, all the decision variables are integer and positive.

7.5 Summary

In this chapter, the reconfiguration and operations planning problem mathematical statement was presented. Many sub-problems related to the ROP were analyzed one by one to illustrate the logic that governs every one of them. The mathematical statement that represents the ROP mini-models assembly was presented in a concise format at the end of the chapter. The ROP mathematical statement shows how progressive modeling can create a new class of large-scale mathematical models by compiling a group of tinier ones. In the next chapter, the solution algorithm for the ROP model presented in this chapter will be discussed.

Chapter 8 RECONFIGURATION AND OPERATIONS PLANNING PROBLEM: THE SOLUTION ALGORITHM

8.1 Introduction

In Chapter 7, an ROP mathematical statement was developed. In this chapter, a relatively lengthy solution algorithm will be presented. The solution algorithm counts a lot on the innovations brought by PM in earlier applications. A new concept of structured decision space will be presented; therefor, the blurred boundaries of the genospace/phenospace presented earlier in chapter 5 will disappear. The concept of encoding/decoding will be eliminated, couplers will be used wherever necessary, and state machines will be utilized as well. The algorithm has two major parts: initialization and recombination. The algorithm abides by the protocol described at chapter 5 (section 5.6.8). The problem will be solved in the multi-objective space using the same optimizer introduced previously in chapters 4 and 5. Chapter 9 introduces a detailed case study.

This chapter starts by describing the society of decision plans that ROP control: configuration maps or operations plans, product plans, workforce plans, inventory, backorders, and subcontracting plans. Every plan has its own logic that outlines how it could be created either from scratch or as an outcome of other plans. In this regard, plans could be classified as independent, semi-dependent, or dependent decision structures. An independent decision structure is the one that can stand alone by itself without further information needed form other structures to complete its definition. All the aforementioned plans are either semi-independent or dependent ones. The interdependence among these structures contributed to coin the term “chained decision space” for the first time. When addressing large-scale problems or systems such as ROP, decision variables and state variables could be grouped into unified decision structures. Accordingly, the search space is composed of many linked societies of these decision structures. In real systems environment, there should be arrays or complex structures of decisions. Taking an ROP as an example, demand plans spurs operation plans, which spur inventory and backorder plans. Subcontracting plans and workforce plans can coexist as well. The structured decision space is the natural evolution of the blurred genospace/phenospace boundaries presented in chapter 5, section 5.6.2. After defining all the ROP decision structures, the operators that could be applied individually to each one of them will be demonstrated. Finally, all the pieces will be weaved together to define the dynamics of the ROP solution. In chapter 9, an ROP case study will be introduced.

8.2 Setting Up the Search Space

In chapter 6, section 6.4.1, the dynamic data model that underlies the ROP problem was presented. Later in chapter 7, it was instrumental in defining the tuplized nomenclature of its mathematical statement. Configurations may be added or removed, workforce may be adjusted, process could be performing better or maybe worse, new products could join or could be removed. Modules and their configurations could be also updated. The first step before initializing the solution algorithm is to filter both the configuration and product spaces. Every configuration defines its product space, i.e. product make lists; any configuration that does not have any of its products belongs to demand mix should be excluded. Obsolete or not demanded products should be removed as well. Configurations and products filtering create a crisply defined search space, which accelerates the solution process as a result. Only candidate configurations and candidate product sets, i.e. feasible configurations and product sets, are available to the search process. In this study, an RMS is assumed to have multiple modules where every module has its list of configurations, and each configuration has its own set of product make lists. Once the demand is identified and the system files are parsed, all candidate configurations and product make lists are immediately filtered and the ROP optimization process could start by then.

8.3 The Structured Decision Space, the Society of Decision Plans, and the Solution Algorithm

As chapters 6 and 7 have revealed, the ROP problem is a compilation of many problems that need to be solved simultaneously in a very dynamic and evolvable environment. In fact, RMS brings a very a challenging environment where there is a multilayer optimization process that needs to be defined implicitly. There is a demand that can be met by many levers: capacity scalability/convertibility options, holding product inventories, allowing overtime, and maybe subcontracting. In order to address the complexity of the ROP and the tremendous number of decisions associated with it, the decision space is defined in terms of decision structures. Every solution point is a society of decision structures. A decision structure is an organized records of data (both decisions and state variables) that wired together to encompass a group of system/product/process/workforce planning variables. In this context, configuration maps, product make plans, supply, inventory, backorders, and subcontracting can be defined. These decisions are semi-independent as already mentioned in the introductory part of this chapter. Starting with a configuration map, other plans can be developed. Once all the decision structures are identified, a group of localized recombination operators are applied to individual decision structures in order to improve the current solutions. Some decision structures can possess several states. State machines are utilized to define different possible states for every structure. Some recombination operators can be applied according to these structures current states. All the recombination operators that can be applied to different ROP structures are presented in this chapter. The chapter concludes by the master algorithm that wires all the pieces together.

8.3.1 Configuration Maps Development

8.3.1.1 Step#1: Configuration Paths

As defined in chapter 6, a configuration or operations path defines a sequence of operations buckets. Every bucket holds many slots that encapsulate related information of both system and product operations. If the RMS under study has multiple modules, the set of configuration paths defines a system configuration map. Figure 8-1 shows the initial configuration map of a system composed of a couple of modules.

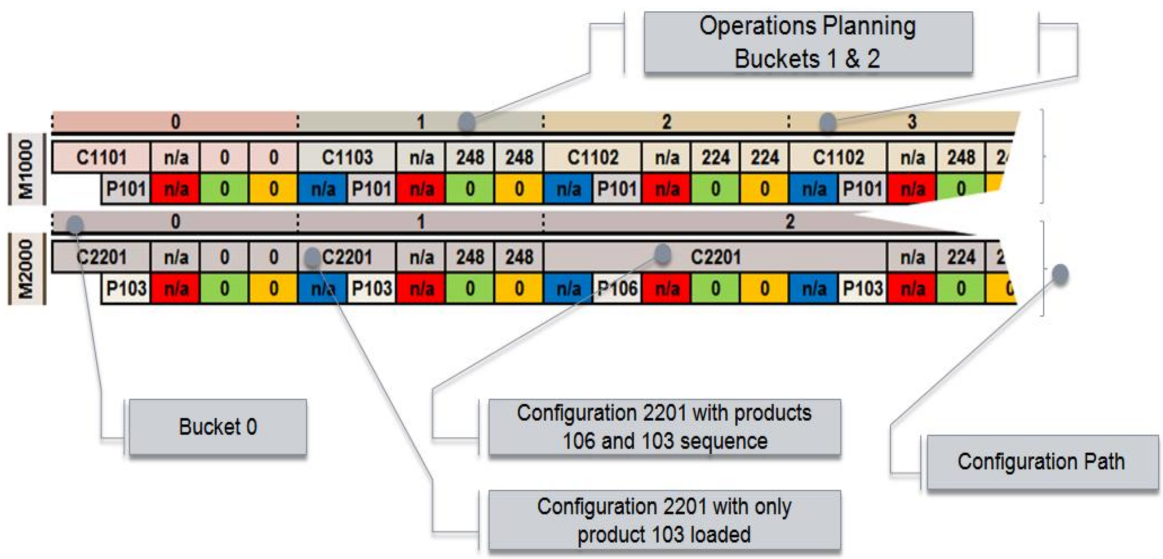


Figure 8-1: Configuration Maps Step# 1: Configuration and Product Sequences

At step#1 of the building process of a configuration map, configuration and products are loaded randomly for all the demand buckets. For every module, there is an independent configuration path. The first slot to be defined is the configuration ID slot. After consulting the candidate configurations list, a configuration ID can be selected. The next slot is updated directly by consulting the system calendar for the bucket duration. For example, February has a number of working days other than March; Once a configuration is selected as the active configuration for the current bucket, a product or a list of products (if the current module is a multi-product) are chosen to be loaded. In case of the current path belongs to a multi-product module, a dynamic list is chosen from the current configuration candidate products set. A crowding factor is chosen first, 80% for example, if the value of a random number is below this value, a product joins the product sequence; otherwise, it is excluded and so on. At least one product must be chosen during that process. The aforementioned operation defines how the implicit lot-sizing problem is managed. Every planning bucket defines five slots for every product: product ID, setup flag/time slot, regular operations time slot, overtime, unloading flag/time slot. Only product IDs slots are updated at this step. In Figure 8-1, Module 1000 is a single product module; therefore, every bucket contains only one bucket. Module 2000, a multiproduct module, bucket 2, for example, has two product tuples.

The pseudo code for stage 1 is described in pseudo code listing 8-1. "Bucket 0" is the last bucket of previous planning session. Bucket 0 is very important because reconfiguration flags (reconfiguration binaries) and product setup decisions are determined according to these values as will be described later in section 8.3.1.2. The only two important pieces of information that "bucket 0" holds are the previous configuration and product IDs. The unloading of bucket 0 last product depends on the next bucket state (configuration and product ID). That is why the copy process is necessary. Any other slots hold the value of zero just for the time being.

Pseudo Code Listing 8-1: Configuration Map Stage 1 Algorithm

```
For each module in RMS
   Initialize an empty configuration path
         Copy bucket 0
         For =1 to Number of planning buckets
         Define a new bucket
         Pick a candidate configuration
         Generate a product make set
         Define bucket length
```

```
        Next bucket
    Next bucket
  Next module
```

8.3.1.2 Step#2: Reconfiguration Flags and Times

Once all configuration IDs are identified, the reconfiguration flags or binaries can be determined according to configuration precedence relations. For better illustration purposes, the top row of any bucket has three additional slots to be updated. The first slot stores the reconfiguration time which implicitly means the reconfiguration binary has the value of 1. If there is no reconfiguration is needed the slot holds the value of "n/a" which implicitly means the reconfiguration time has the value of zero, i.e. no reconfiguration process during that bucket should take place. Once the configuration time is updated, the time available for product operations is ready to be reallocated, see Figure 8-2 for an illustration. Once the reconfiguration binaries and configuration times have been updated, the reconfiguration costs can be evaluated for all buckets, modules, and the system. The costs are extracted from their corresponding configuration objects, i.e. the members of configuration candidates of every module.

8.3.1.3 Step#3: Setup and Unloading Decisions

Both the setup/ramp up and unloading decisions are dependent on both configuration precedence and products precedence relations. The unloading algorithm is more complicated than the reconfiguration algorithm. *The unloading process of the last product is always set to be the first operation of the next bucket*. This interprets the shift that appears at the second row of every configuration path. Whenever there is a reconfiguration process, the product that already loaded to the system has to be unloaded first. The setup/unloading process differs if the module at hand is a single product or a multi-product one.

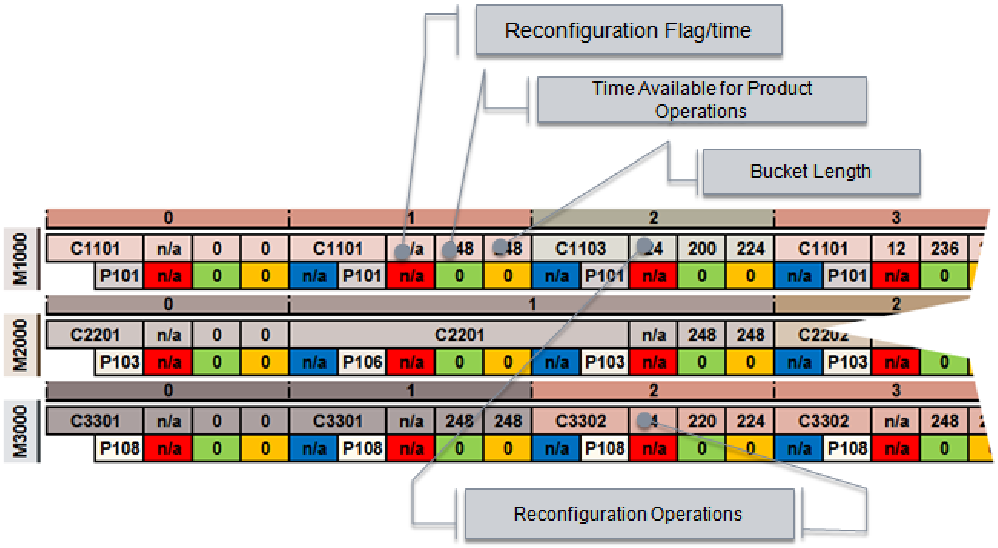


Figure 8-2: Configuration Map Setp#2: Updating Reconfiguration Flags and Times

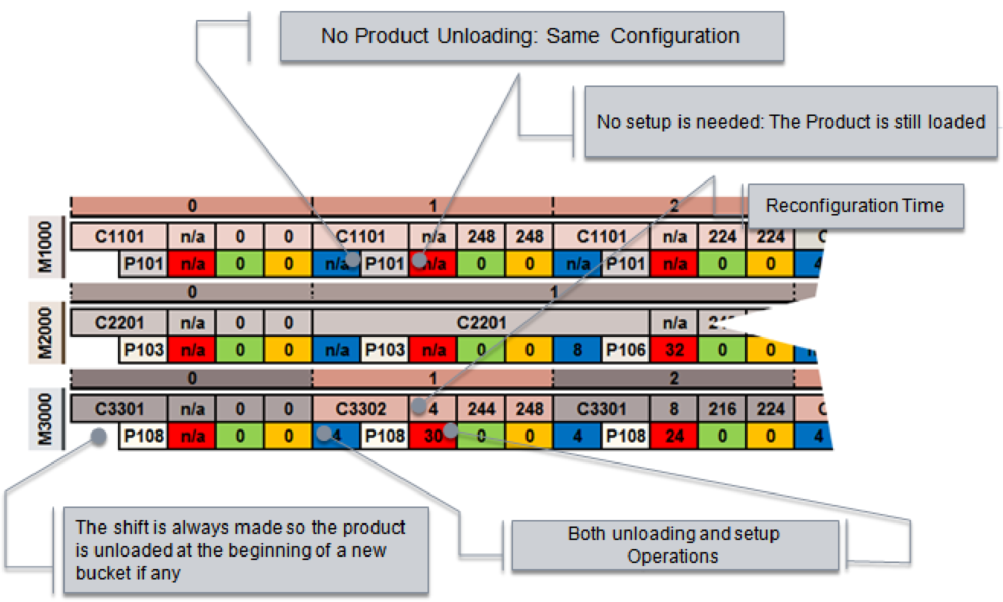


Figure 8-3: Configuration Map Step#3: Update the Setup and Unloading Flags

Even though the setup and unloading algorithms are a little bit lengthy, the logic that governs them is concisely encapsulated mathematically by the couple of hierarchical binaries described in the previous chapter and is repeated here, equations (7.9) and (7.15). An interested reader can consult the corresponding verbal rules in chapter 7, sections 7.3.2.2 and 7.3.2.4 respectively, to get a full description of how setup/unloading decisions are made. Once the setup/unloading decisions are made, their associated times can be updated. There are a couple of slots for set/up and unloading decisions and times used expressively to give all the details of the set up and unloading processes. The unloading slot holds the value of “n/a” which implicitly means the unloading time is zero and there is no unloading will take place. If an unloading process is needed, the unloading time slot holds the corresponding time value, which implicitly means the unloading decision is yes. Color-coded slots are used to make reading the configuration maps easier; the unloading slot has a blue-coloured slot, the setup has a red-coloured one; a product ID occupies a white slot, the regular and overtime occupy green and yellow slots respectively.

$$\exists p \in P_{\left[c_k^m\right]} :$$

$$\beta^{srp}_{p\left[c_k^m\right]}[t] = \begin{cases} 0 & p \notin P^{()}_{\left[c_k^m\right]}[t] \\ \begin{cases} 1 & p \neq First(P^{()}_{\left[c_k^m\right]}[t]) \\ \begin{cases} 1 & c_k^m \neq CP^m[t-1] \\ \begin{cases} 1 & p \neq Last(P^{()}_{\left[c_k^m\right]}[t-1]) \\ 0 & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} \tag{7.9}$$

$$\exists p \in P_{\left[c_k^m\right]} :$$

$$\beta^{unld}_{p\left[c_k^m\right]}[t] = \begin{cases} 0 & p \notin P^{()}_{\left[c_k^m\right]}[t] \\ \begin{cases} 1 & p \neq Last(P^{()}_{\left[c_k^m\right]}[t]) \\ \begin{cases} 0 & t = N_b \\ \begin{cases} 1 & p \neq First(P^{()}_{\left[c_k^m\right]}[t+1]) \\ \begin{cases} 1 & c_k^m \neq CP^m[t+1] \\ 0 & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} & otherwise \end{cases} \tag{7.15}$$

8.3.1.4 Step#4: Operation Scheduling

Once all reconfiguration, set up, unloading flags and their corresponding times are updated, the exact time available for production operations can be allocated. If the bucket has only one product, the remaining time will be allocated to that product; otherwise, i.e. the multiproduct case, the time available for operation is distributed using a random proportioning algorithm. The operation scheduling coupler is designed to make the individual production operations always ends at a full day. This has some implications on product switching process and makes overtime decisions much easier.

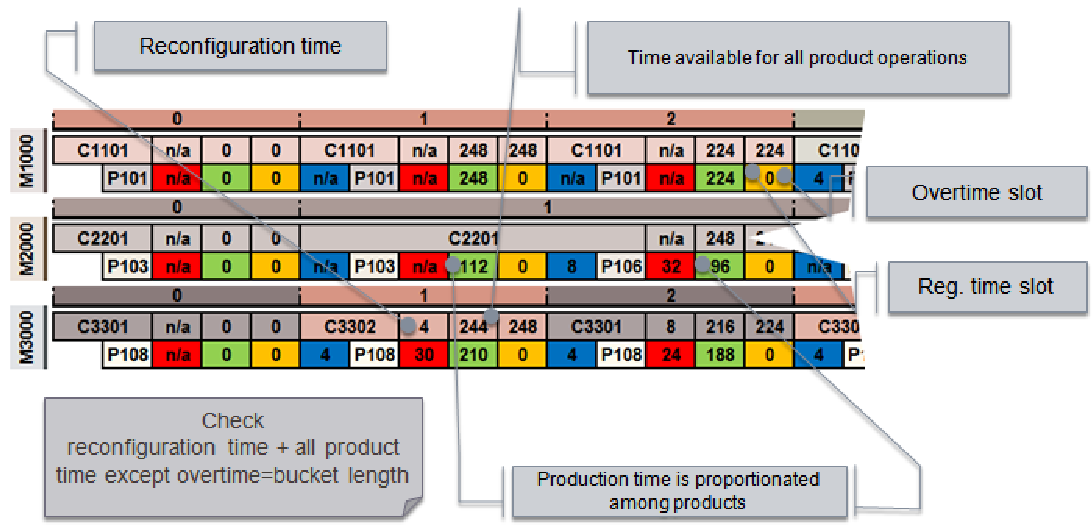


Figure 8-4: Configuration Map Step#4: Regular time production operation time allocation

8.3.2 Configuration map, next links and the missing slots

Until now, all the configuration map slots are updated except the overtime ones. As will be described later, product supply plans and product make plans are the ones that define the amount of time that should be allocated for overtime slots. Product Make plans define the time allocation constraints and product supply ones define what could be produced to meet the demand without violating the constraints defined by product make plans. Product Make plans are described in the next section.

8.3.3 Product Make Plans: Part I

In an RMS environment, system throughput and production costs are configuration dependent. The productivity rate became variable due to different configuration throughputs. Product make plans are developed in order to identify production time and quantities during regular and overtime periods. A certain product might be produced by more than one module. In order to determine the product supply later (see section 8.3.5), the product make plans are supposed to be identified first. A Product make plan is composed of N_b tuples of product regular and overtime production times and quantities where N_b is the number of planning buckets. If a product is not produced during a certain bucket, a nullable tuple is assigned (all slots are marked by n/a), see Figure 8-5.

In order to develop a product make plan, the corresponding regular time production is extracted from the system's configuration map. Using the product ID and by consulting the corresponding configuration data files for the productivity rate or throughput (1/cycle time), the product regular quantity is identified as a result. By checking the system work regulations and calendar, the maximum value of the overtime slot can be allocated as well. This maximum value is a constraint on the overtime that can be consumed by a certain product. Both the overtime time and the overtime quantity are decided during developing the product supply plans as will be described later in section 8.3.5.

Product::106 Module::2000								
Column1	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Reg Time	24	148	48	N/A	N/A	112	64	56
Reg Quantity	192	2220	720	N/A	N/A	1560	768	840
Max Over Tin	4	38	12	N/A	N/A	26	16	14
Max OT.Quai	48	570	180	N/A	N/A	390	192	210
Ov Time	4	38	12	N/A	N/A	26	16	14
OvQuantity	48	570	180	N/A	N/A	390	192	210

Figure 8-5: Product Make Plan

8.3.4 Workforce Plans

Once a system is scaled up/down, a workforce adjustment may be necessary. According to data model presented, every module has its own workforce plan. If the workforce members are allowed to be switched among system modules, the system workforce plan will have a meaning. All variables related to workforce decisions should be dependent ones. The workforce level is a configuration design parameter. With a certain reconfiguration process, workforce adjustments are assumed. Workforce plans could be determined for both modules and the system as the whole. Figure 8-6 shows a sample of both system and module workforce plans. For every module, there is a corresponding workforce plan. Configuration IDs are extracted from configuration maps first and the system data files are consulted to extract the corresponding workforce level information. Thereafter, the hiring and firing values are updated as a result.

In traditional manufacturing, when addressing the capacity problems, there are always two factors under consideration: machinery and workforce. Usually, one factor is considered a leading factor. In traditional cost accounting, the workforce was considered the leading factor. In RMS, at least from the ROP perspective, this basic tenet might need to be changed; configuration (i.e. machinery) rather than direct labour should be the leading factor. Figure 8-6 shows some samples of modules and system workforce plans.

Module 1000 Workforce Plan

	P. State	1	2	3	4	5	6	7	8
W	10	10	10	17	14	10	10	10	17
H		0	0	7	0	0	0	0	7
F		0	0	0	3	4	0	0	0

Module 2000 Workforce Plan

	P. State	1	2	3	4	5	6	7	8
W	12	12	12	16	16	16	16	16	16
H		0	0	4	0	0	0	0	0
F		0	0	0	0	0	0	0	0

Module 3000 Workforce Plan

	P. State	1	2	3	4	5	6	7	8
W	12	16	12	16	16	16	16	12	16
H		4	0	4	0	0	0	0	4
F		0	4	0	0	0	0	4	0

System Workforce Plan

	P. State	1	2	3	4	5	6	7	8
W	34	38	34	49	46	42	42	38	49
H		4	0	15	0	0	0	0	11
F		0	4	0	3	4	0	4	0

Figure 8-6: Sample RMS Module and System Workforce Plans

8.3.5 Product Supply Decisions and Plans

Once the product make plans have been updated, the immediate system levers (overtime, subcontracting, and inventory) should be ready to create the required balance between demand and supply. The system envelop constraints concepts presented at chapter seven are instrumental in determining the supply plans decisions. By consulting target inventory/customer service levels and checking the output of production operations, the supply and demand can be matched after consulting the remaining available system levers as well.

$$B_p[0] = A_{1i} \qquad \forall\, Product\ p \in P \tag{7.34}$$

$$I_p[0] = A_{2i} \qquad \forall\, Product\ p \in P \tag{7.35}$$

$$B_p[N_p] = \mathfrak{M}_{1i} \qquad \forall\, Product\ p \in P \tag{7.36}$$

$$I_p[N_p] = \mathfrak{M}_{1i} \qquad \forall\, Product\ p \in P \tag{7.37}$$

$$\exists p \in P: \quad \sum_{t=1}^{t=N_b} PSB_p[t] + \left\{ I_p[0] - B_p[0] \right\} + \left\{ I_p[N_B] - B_p[N_B] \right\} = \sum_{t=1}^{t=N_b} D_p[t] \tag{7.38}$$

Using these equations 7.34 to 7.38 the following algorithm is applied:

For each product in product mix

Step 1: if a product demand is less than its total regular supply, the product plan state is marked as "*Regular Plan*"; regular production is enough; escape to the next product.

Step 2: if the product demand is greater than its total supply taking into consideration all the overtime quantities that can be produced and subtracted if any, all the overtime slots should be occupied with the maximum overtime quantities specified by product make plans part I (section 8.3.3). In addition, if subcontracting is allowed, the maximum subcontracting quantity allowed is distributed all over the planning horizon. A subcontracting coupler takes care of this part. The product plan state is marked as "*Subcontracting Ceiling"* (if subcontracting is allowed) or "*Overtime Ceiling*" (if the subcontracting is not allowed). Escape to the next product.

Set 3: if subcontracting is allowed and the demand is greater than all the manufacturing capability (both overtime and regular time) but **less** than the total supply, subcontracting will be needed. A subcontracting coupler decides on the quantities needed and product plan state is marked as "*Subcontracting below Ceiling*".

Step 4: if the demand is greater than the regular time and less than the maximum quantity provided by both regular and overtime, overtime only will be needed. An overtime coupler is utilized to decide on the overtime quantities. The plan state is marked as "*Overtime below Ceiling.*"

Similar to hierarchical binaries the product supply rules are recursively mutually exclusive and the algorithm terminates to the next product once one of these rules is fired until all product plans are updated. Figure 8-7 illustrates an updated product supply plan.

Product 101 Plan

	P. State	1	2	3	4	5	6	7	8
D		2500	4000	2000	1500	6000	3000	1500	4000
Rmax		1584	2136	1224	1520	1984	1068	1248	2160
R		1584	2136	1224	1520	1984	1068	1248	2160
Omax		400	552	312	384	496	276	312	552
O		400	552	312	384	496	276	312	552
S		1022	1631	2449	1224	612	816	612	1633
I	0	0	786	610	2238	675	0	183	0
B	2200	1694	1375	1021	5	670	1510	838	493

Figure 8-7: an Updated Product Supply Plan

8.3.6 Product Make Plans: Part II

The last missing piece of information related to product make plans is identifying the overtime slots. Using the product supply plans, this part is done backwardly. If the product at hand is produced by just one module, the overtime periods, and quantities are updated after consulting the product supply plans. If that product can be produced by more than one module, an overtime distribution coupler may be utilized to allocate overtime among eligible modules.

Product::106 Module::2000

Column1	1	2	3	4	5	6	7	8
Reg Time	24	148	48	N/A	N/A	112	64	56
Reg Quantity	192	2220	720	N/A	N/A	1560	768	840
Max Over Tin	4	38	12	N/A	N/A	26	16	14
Max OT.Quai	48	570	180	N/A	N/A	390	192	210
Ov Time	4	38	12	N/A	N/A	26	16	14
OvQuantity	48	570	180	N/A	N/A	390	192	210

Figure 8-8: an Updated Product Make Plan

8.3.7 Inventory and Back/orders

After deciding on all product supply decisions, both inventory and backorders records can be identified iteratively for all products and for all buckets.

Product 101 Plan

	P. State	1	2	3	4	5	6	7	8
D		2500	4000	2000	1500	6000	3000	1500	4000
Rmax		1584	2136	1224	1520	1984	1068	1248	2160
R		1584	2136	1224	1520	1984	1068	1248	2160
Omax		400	552	312	384	496	276	312	552
O		400	552	312	384	496	276	312	552
S		1022	1631	2449	1224	612	816	612	1633
I	0	0	786	610	2238	675	0	183	0
B	2200	1694	1375	1021	5	670	1510	838	493

Figure 8-9: A Complete Product Plan

8.4 Objectives Evaluation:

8.4.1 Max Operations Profitability

The objectives utilized in this study are profit (maximized), inventory investment (minimized), and backordering (minimized). The cost structure of production operations environment is non-linear. Same product can be produced using many cost values. Profit margins rather than costs play a supreme leading factor in determining which product supply mix and volumes is the best. The profit function (7.55) developed in chapter 7 is utilized for this purpose.

$$\sum_{\forall p\in P}\rho_p^{price} Min\{D_p[t], PSB_p[t]\} - \left[\underbrace{\left(\begin{array}{l} \underbrace{\sum_{m=1}^{N_m}\sum_{t=1}^{N_b}\sum_{\forall p\in P_{[c_k^m]}} PMB_{p[c_k^m]}^{C-unld}[t]}_{Unloading} \\ \underbrace{\sum_{m=1}^{N_M}\sum_{t=1}^{N_B} C_{CP^m}^{rcnfg}[t]}_{\text{Re}configuraion} \\ \underbrace{\sum_{m=1}^{N_m}\sum_{t=1}^{N_b}\sum_{\forall p\in P_{[c_k^m]}} PMB_{p[c_k^m]}^{C-srp}[t]}_{Set/Ramp\,up} \\ \underbrace{\sum_{t=1}^{N_B}\left(PS_p^{C-mtr} + PM_{p[c_k^m]}^{VC}\right)\left(PMB_{p[c_k^m]}^{R}[t] + PMB_{p[c_k^m]}^{O}[t]\right)}_{Machining\,\&\,Materials} \end{array} \right)}_{Manufacturing\,Operations\,Costs} + \underbrace{\left(\begin{array}{l} \underbrace{\sum_{m=1}^{N_M}\sum_{t=1}^{N_B} C_H^m H^m[t] + C_F^m F^m[t] + C_W^m W_m[t] SWR^{H/B}[t]}_{Hiring,\,Firing,\,and\,Payroll} \\ \underbrace{\sum_{\forall p\in P}\sum_{t=1}^{N_B} PS_p^{C-hld} I_p[t]}_{Inventory\,holding\,costs} \\ \underbrace{\sum_{\forall p\in P}\sum_{t=1}^{N_B} PS_p^{C-sbcntrc} PSB_p^S[t]}_{Subcontracting} \end{array} \right)}_{Other\,Costs} \right] \quad \textbf{(7.55)}$$

8.4.2 Minimize capital investment in inventory

The inventory investment is to be minimized in order to promote the lean practice and maximize the bottom line financials, which spurs better cash flow and consequently better market value of the manufacturing firm itself.

$$Min\,Z_3 = \frac{1}{N_b}\left\{\sum_{\forall p}\sum_{t=1}^{N_b} PS_p^{C-bkord} \max\left\{-I_p[t],0\right\}\right\} \tag{7.56}$$

It is very important to notice that all the objectives reported here are sample implementations. PM allows anything to be developed further or redefined as necessary.

8.4.3 Minimize backorders

Since the backordering objective is not a tangible cost, backordering objective is not treated as a cost term. There might be a manufacturing policy or strategy to achieve only a certain level of customer service (blew 100%). Even though backordering costs were usually incorporated in cost functions in some traditional production planning models, this not the case for ROP.

$$Min\,Z_2 = \frac{1}{N_b}\left\{\sum_{\forall p}\sum_{t=1}^{N_b} Pr_p \max\left\{I_p[t],0\right\}\right\} \tag{7.57}$$

8.5 Initialization Algorithm

Now that all the pieces of the puzzle are in shape, the initialization algorithm can be summarized as follows: the algorithm developed is a population based. ROP solution algorithm has its roots in genetic algorithm so the initialization is similar to GA one as described in code listing 8.2. A group of localized cross over and mutation operators are utilized for recombination purposes.

Pseudo Code Listing 8-2: ROP Initialization Algorithm

```
Create an empty population (pop)
For i=1 to PopSize
    Create all decision structures (configurations maps, all product
    plans, and workforce plans)
    Assemble all these data structures into a composite structure
    called Plan.
    Evaluate all the objectives and store their values in an
    objectives vector
    Attach these plans and their objectives to an individual
    Add an Individual to pop
Next i
```

8.6 Recombination Operators

As already described, an ROP plan is an assembly of different decision structures. Recombination process takes place whenever possible to search for better solutions. Since all their decisions are independent ones, configuration maps take the lion share of these operators. Except for product overtime and subcontracting decisions, all other variables are dependent ones and they cannot be changed by themselves. After a recombination operation happens, a localized update might just be needed. For example, when one path is a subject for a mutation operation, only that path undergoes an updating process. Nevertheless, the global evaluation of objectives, system level evaluation, is always executed. For example, every path has its cost terms of reconfiguration and production operations. Once this value is updated, the total sum for all the configuration map members is updated. In configuration maps, whenever a binary decision changes, a reconfiguration decision for example, the other time values (flexible ones only, regular and overtime values) are just massaged to get an updated values. This is extremely important if we want to maintain the good schema developed throughout the solution process.

8.6.1 Cross Over Operators

The cross over operators could take place at two levels: map level and path level. The next couple of sections describe the details.

8.6.1.1 Map Operators

In map operators, a single point cross over is just used because the ROP might only define a handful of modules, which is a very small number to be considered for a double point cross over. A crossing point is chosen according to the number of modules of the system. The crossing points divide the configuration map into two configuration path groups (smaller configuration maps) and then these groups are swapped to form a new couple of children. Figure 8-10 illustrates the process.

a) Before cross over

b) After Cross Over

Figure 8-10: Map Cross Over—Module 3000 path has been selected for cross over.

8.6.1.2 Path Operators

The path cross over operators are bucket level operators. Both single point and double point cross over operators can be applied. When the cross over operators takes place, both configuration and product sequencing decisions are updated according to their new precedence relations. As for the regular time slots they are recalculated in a way that make them very close to their original values. If the set up operation is decided, for example, the set up time needed is cannibalized from other products operations. Similarly, an added time can be redistributed if a set up operation is omitted. The goal is to maintain the time proportions as close as possible to their original values. This would contribute a lot in not losing good schema or good solution out of the cross over operation process. Only a couple of identical paths are allowed to go for crossed over operation. Figure 8-11 shows an example of the double point cross over operator. Configuration paths of Module 3000 are swapped. The cross over bucket range is bucket 3 to bucket 6 (double point). It is important to notice that in bucket 3 of Module 3000 path, the reconfiguration process was discarded (the new bucket configuration ID slot holds the same configuration ID value). The time gained is injected in the production operations.

a) Path Double Point Before Cross Over

b) Path Double Point Cross Over Operator

Figure 8-11: A Double Point Path Cross Over: M3000 path is chosen, and buckets 3-6 are the subject of the cross over operation.

8.6.2 Mutation Operators

Mutation operators are applied to both configuration maps, product plans (overtime and subtracting if their states are below their ceiling values). Subcontracting ceiling plans can be mutated as well. The following subsections describe the details.

8.6.2.1 Configuration Map

8.6.2.1.1 Bucket Operators

Configuration Flip Operator

In the configuration flip operator, a path and then a bucket of this path is chosen randomly. The configuration slot is replaced with another configuration from the module candidate configuration list.

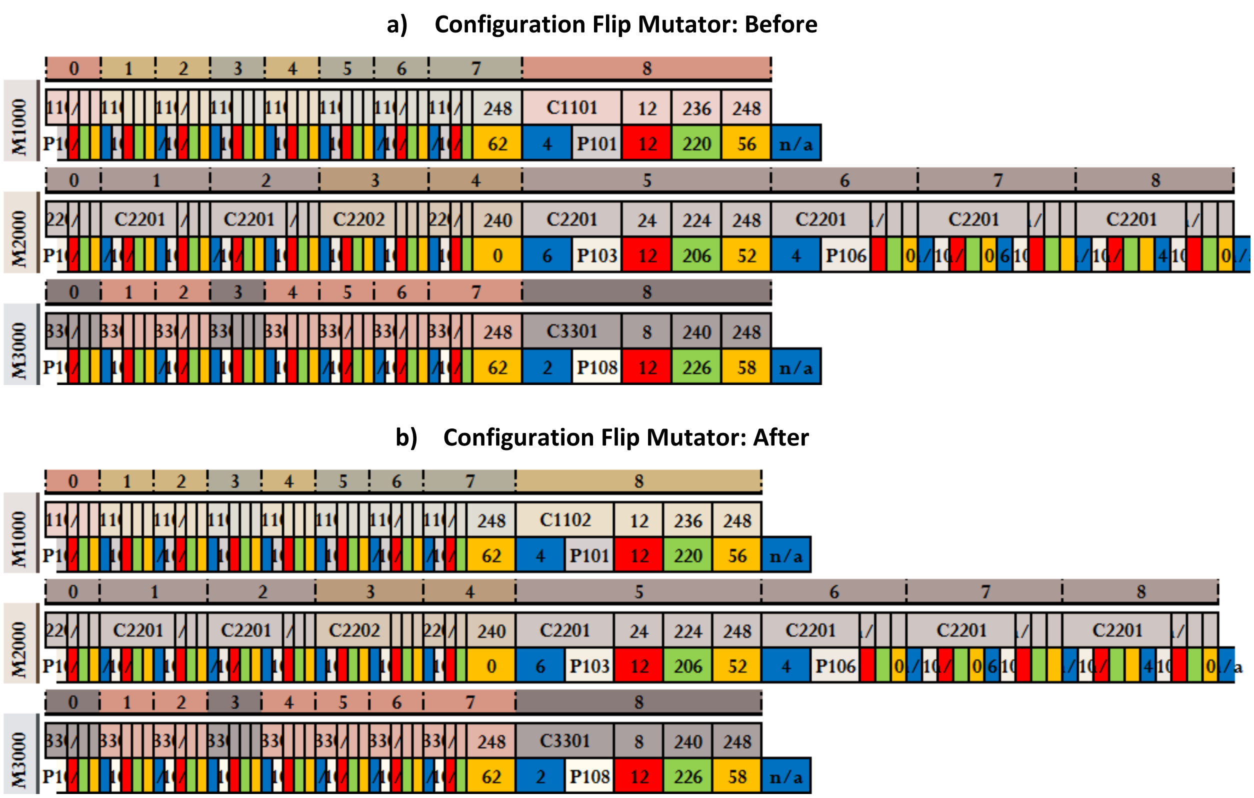


Figure 8-12: Configuration Flip Mutator: Module 1000, Bucket 8 configuration flipped from C1101 to C1102

Bucket Sequence Operator

The bucket sequence operator is a cardinal operator that has subordinates. The underlying operators are Swap, Insert, and Inverse. In Swap, a couple of buckets are swapped. In Insert, a bucket is cut from one place, all other buckets next to the cut point location are shifted one move, and the cut bucket is inserted in the insertion place. In Inverse, a sequence of buckets is inverted.

Multi Product Path Operators (Product Tuples Operators)

If the module at hand is a multi-product, the following operators can be applied: Product Tuples Add and Drop, Product Tuples Sequence, Product Operations Time Slices. In Product Tuples Add and Drop operators, a certain product is added or dropped from the product sequence list. Drop operators cannot be applied unless the bucket contains at least a couple of product tuples. As for Product Tuples Sequence, the subordinate operators: Insert, Inverse, and Swap could be applied to the product tuples as one block. Time slices allocated for regular time can be also mutated using Swap, Inverse, and Insert. In addition, a canalization operator can be applied as well, i.e. a time slice is subcontracted form one product regular time and added to the other's.

8.6.2.1.2 Overtime Operators

Overtime operators are only applied to product plans. If a product overtime state is below its ceiling, overtime operators can be applied to get a better distribution of overtime hours over the planning horizon. This operator is applied to product make plans. Overtime can be swapped, inserted, inverted, or even cannibalized provided that the overtime ceiling constraint is not violated.

8.6.2.1.3 Subcontracting Plans Operators

If a product allows subcontracting and regardless of its ceiling values, the following operators could be applied: subcontracting slots can be swapped, inserted, inverted, and cannibalized. Unlike overtime operators, there are no constraints to be checked.

8.7 ROP: The ROP Master Algorithm

After the long discussion of both the initialization and recombination mechanisms utilized in order to solve and optimize the ROP, the time has come to wire all the parts together:

Step1: The problem data files, system and market data (product mix and volumes) are read and filtered to define only candidate configuration and candidate products. This will lead to a better memory footprint and less computational power. The user interface component is responsible for this part. In this study, the entire problem data is hard coded, which could be enough for research purposes. Once the data is read, it becomes available and ready to be populated to other components: modeller, system machinery and product makes, products, and optimizers.

Step2: the solution process starts from the modeller where the solution process can be sparked by system user and the modeller starts to trigger the optimization algorithm in the optimizer. The modeller implements the interface called "IGenerator" presented in chapter 5 and takes charge of the initialization and recombination algorithms. The modellers also take care of the creating and updating the society of problem decision structures and the evaluation of their objectives. The modeller attaches itself with its entire internal component to the optimizer, which controls the solution process.

Step3: once the *Modeler* activates the optimization process, the optimizer takes hold of everything. Through the "*IGenerator*", which describes the communication protocol between the modeller and the optimizer, the optimizer is now hooked to both initialization and recombination algorithms. The optimizer executes the initialization process, performs selection and Pareto-front updates (SPEA2 implementation), and trigger the recombination process. The process is iterated for a certain number of generations, i.e. stopping criterion, and the results are printed on screen or saved to a disk for a later analysis or post processing.

The algorithm described throughout this chapter was implemented using the C# language 3.0 and the .NET Framework 3.5. The results are shipped to Excel that acts as a COM server to show the results.

The last master algorithm shows that the solution strategy developed is almost the same used for MMAPP even if the ROP cannot be compared to the MMAPP in the previous chapter 5. This a very good example of how Software Engineering, CBSE to be specific, can lead us to thought reuse. CBSE epitomizes itself as a technical enabler of CMPC frameworks.

8.8 Summary

This chapter introduced the solution algorithm for the ROP problem. The chapter started by showing how the decision space structures defined and evaluated one by one. The next chapter will introduce the case study developed to test both ROP model and its solution algorithm.

Chapter 9 RECONFIGURATION AND OPERATIONS PLANNING PROBLEM: CASE STUDY, RESULTS, AND POST ANALYSIS

9.1 Introduction

Over the last three chapters, the ROP problem has been presented. In this chapter, a case study with sample results is introduced. The case study is designed to make the discussion of the ROP problem as generic as possible. Progressive Modeling was developed to deal with the lack of crisp perception of what is a reconfigurable manufacturing process. Thinking in terms of Progressive Modeling not only makes the modeling process of RMS easier but also unleashes the potential to develop RMS large-scale system models and better realistic case studies. In this chapter, an ROP case study is developed and presented. Results are demonstrated and post analyzed. The chapter concludes the ROP problem and discusses the new potential of Progressive Modeling.

9.2 Case Study

In this case study, four products are assumed to be produced by three reconfigurable modules. These four parts could be four parts of an engine piston block assembly, different gears, etc. From the ROP perspective, the type and the design of products themselves are irrelevant, all what is needed is the data defined by the data model presented in chapter six, which has no assumptions related to product type or design.

In this case study, the RMS is supposed to be composed of three different modules. The first (M1000) is designed to be a single product scalable-RMS module; just one product can be manufactured using three different configurations. The second module (M2000) is designed to be a functional-RMS (multi-product scalable-RMS) in which a multiple products can be produced using a couple of alternative configurations. The third module (M3000) is the one that is designed to disseminate the system evolve-ability. As already described in chapter six, every configuration defines its product space with different operations parameters. Product P108 is supposed to be a replacement of product P107 that became obsolete. When the filtering process starts, see chapter eight for details, the filtered product list will show just one product because the other product will be discarded and the module M3000 will be treated as if it were just a single product scalable module. The following subsections report all the detailed case study data with some descriptions wherever necessary.

9.2.1 Demand and Product Data

The demand horizon is assumed to span 8 months, starting January and ending August. Product demand portfolio includes products P101, P103, P106, P108 respectively. Every product has its own demand curve as already reported by Table 9-1. In Table 9-2, more related data is reported about initial inventory (Io), backorders (Bo), unit holding cost (Ch), backordering cost (Cb), materials cost (Cm), and subcontracting cost if any (Cs).

Table 9-1: Products Demand

Product Number	Product ID	1	2	3	4	5	6	7	8
1	101	2500	4000	2000	1500	6000	3000	1500	4000
2	103	3000	2000	4500	3800	4200	5000	4500	3000
3	106	1500	1200	1000	2500	2000	1400	1700	1200
4	108	3000	2400	1500	1500	3400	2000	1100	2100

Table 9-2: Products other Information

	ID	Io (Units)	Bo (Units)	Ch ($/Unit)	Cb ($/Unit)	Cm ($/Unit)	Cs ($/Unit)
1	101	0	2200	0.5	40	20	35
2	103	1300	0	0.6	27	17	N/A
3	106	0	2200	0.5	40	20	N/A
4	108	0	1000	0.7	30	19	N/A

9.2.2 System domains, Configuration Domain, Product Domains

Every module has its own ID and defines its configuration domain. Similarly, every configuration defines its product domain. As a convention, every module ID starts with a capital M, every configuration ID starts with a capital C, and every product ID starts with a capital letter P. The concept of IDs captures the dynamically structured manufacturing environment that the RMS brought to the manufacturing world: new modules can be added or removed from RMS system. New configurations can be added to/dropped from a certain RMS module library, new configurations can be developed (process development), some machines or even machine modules may be added or removed. Products can be further developed. i.e. product features can be added or removed, another material can be used etc. All the analytics, modeling, algorithms, and now the case study have taken care of all these system dynamics. RMS systems define system domain (module-level data), process domain (configurations), and product domain (process capability). The remaining part of the case study data reflects all the data needed to be defined.

RMS: MODULES, CONFIGURATIONS, WORKFORCE AND WORK REGULATIONS DATA

Every module is presented by a data a block, Module M1000 may be taken as example to describe the data related. M1000 defines three different configurations (C1101, C1102, and C1103). Every configuration has its own workforce level attached (Table 9-3). Every configuration defines its cousins of configurations that can be reconfigured from.

Every configuration defines its workforce, reconfiguration data, and product make parameters (product list). All reconfiguration costs are either fixed costs or function of reconfiguration time (variable costs) or both. Every effort was spent to have a case study with a high level of integrity and meaningful figures. Every configuration defines a process space for its products (different unloading time, setup time, cycle time etc.). Variable costs are function of reconfiguration time. This part epitomizes the importance of function templates developed at the early days of PM. If for any reason this not the right way to estimate the reconfiguration costs, another implementation can be replaced. The last part of data of the configuration block is the product make data, i.e. product make domain. Every product defines product set up cost, set up time, and a cycle time (1/throughput). Similar to set up process, an unloading process defines both unloading time and cost. As reported in chapter 6, a product may be unloaded for two reasons: the first takes place when a reconfiguration process is executed and the other takes place when the configuration loaded allows more than one product to be produced, i.e. multi-product module. The last piece of data related to product make data is the variable machining cost.

The following blocks of data, modules and configuration blocks, report the detailed data used to describe the RMS system under study.

MODULE 1000

::Configuration 1101::::::::::::::::::::::::::::::::::::::

Table 9-3: C1101 Workforce

Item	Units	Value
WF Level	Man/Period	10

Table 9-4: C1101 Reconfiguration Data

#	ID	Time (Hrs)	Variable Cost ($/hr)	Fixed Cost ($)
1	1102	8	400	1600
2	1103	12	400	2400

Table 9-5: C1101 Product List

Product Make Parameter	Value
ID	101
Ramp/Set up Cost ($)	6000
Ramp/Set up Time (Hrs)	12
Unloading Cost ($)	1000
Unloading Time (Hrs)	2
Cycle Time (min)	10
V. Mach Cost ($/Unit)	0.7

::Configuration 1102::::::::::::::::::::::::::::::::::::::

Table 9-6: C1102 Configuration Workforce

Item	Units	Value
WF Level	Man/Period	14

Table 9-7: C1102 Reconfiguration Data

#	ID	Time (Hrs)	Variable Cost ($/hr)	Fixed Cost ($)
1	1101	16	300	4800
2	1103	12	300	3600

Table 9-8: C1102 Product List

Product Make Parameter	Value
ID	101
Ramp/Set up Cost ($)	10000
Ramp/Set up Time (Hrs)	15
Unloading Cost ($)	1500
Unloading Time (Hrs)	3
Cycle Time (min)	7.5
V. Mach Cost ($/Unit)	0.55

::Configuration 1103::::::::::::::::::::::::::::::::::::::

Table 9-9: C1103 Workforce

Item	Units	Value
WF Level	Man/Period	17

Table 9-10: C1103 Reconfiguration Data

#	ID	Time (Hrs)	Variable Cost ($/hr)	Fixed Cost ($)
1	1101	24	300	7200
2	1102	16	300	4800

Table 9-11: C1103 Product List

Product Make Parameter	Value
ID	101
Ramp/Set up Cost ($)	15000
Ramp/Set up Time (Hrs)	20
Unloading Cost ($)	3000
Unloading Time (Hrs)	4
Cycle Time (min)	5
V. Mach Cost ($/Unit)	0.45

Module 2000

:::Configuration 2201:::

Table 9-12: C2201 Workforce

Item	Units	Value
WF Level	Man/Period	12

Table 9-13: C2201 Reconfiguration Data

#	ID	Time (Hrs)	Variable Cost ($/hr)	Fixed Cost ($)
1	2202	24	300	7200
2	2203	48	300	14400

Table 9-14: C2201 Product List

Product Make Parameter	1	2
ID	103	106
Ramp/Set up Cost ($)	12000	15000
Ramp/Set up Time (Hrs)	12	16
Unloading Cost ($)	3000	3500
Unloading Time (Hrs)	4	6
Cycle Time (min)	10	5
V. Mach Cost ($/Unit)	1.2	0.8

:::Configuration 2202:::

Table 9-15: Configuration Workforce

Item	Units	Value
WF Level	Man/Period	16

Table 9-16: Reconfiguration Data

#	ID	Time (Hrs)	Variable Cost ($/hr)	Fixed Cost ($)
1	2201	24	300	7200
2	2203	48	300	14400

Table 9-17: Product List

Product Make Parameter	1	2
ID	103	106
Ramp/Set up Cost ($)	16000	20000
Ramp/Set up Time (Hrs)	16	20
Unloading Cost ($)	2000	4500
Unloading Time (Hrs)	3	6
Cycle Time (min)	8	4
V. Mach Cost ($/Unit)	1	0.75

Module 3000

::Configuration 3301:::

Table 9-18: Configuration Workforce

Item	Units	Value
WF Level	Man/Period	12

Table 9-19: Reconfiguration Data

#	ID	Time (Hrs)	Variable Cost ($/hr)	Fixed Cost ($)
1	3302	8	300	2400
2	3303	16	300	4800

Table 9-20: Product List

Product Make Parameter	1	2
ID	105	108
Ramp/Set up Cost ($)	8000	10000
Ramp/Set up Time (Hrs)	12	12
Unloading Cost ($)	1200	1200
Unloading Time (Hrs)	2	2
Cycle Time (min)	12	12
V. Mach Cost ($/Unit)	0.75	0.8

::Configuration 3302:::

Table 9-21: Configuration Workforce

Item	Units	Value
WF Level	Man/Period	16

Table 9-22: Reconfiguration Data

#	ID	Time (Hrs)	Variable Cost ($/hr)	Fixed Cost ($)
1	3301	4	300	1200
2	3303	8	300	2400

Table 9-23: Product List

Product Make Parameter	1	2
ID	105	108
Ramp/Set up Cost ($)	9000	12000
Ramp/Set up Time (Hrs)	16	14
Unloading Cost ($)	1500	1500
Unloading Time (Hrs)	3	2
Cycle Time (min)	10	10
V. Mach Cost ($/Unit)	0.7	0.75

::Configuration 3303:::::::::::::::::::::::::::::::::::::::

Table 9-24: C3302 Workforce

Item	Units	Value
WF Level	Man/Period	16

Table 9-25: Reconfiguration Data

#	ID	Time (Hrs)	Variable Cost ($/hr)	Fixed Cost ($)
1	3301	6	300	1200
2	3302	10	300	1800

Table 9-26: C3302 Product List

Product Make Parameter	1	2
ID	105	108
Ramp/Set up Cost ($)	14000	14000
Ramp/Set up Time (Hrs)	18	15
Unloading Cost ($)	1600	1400
Unloading Time (Hrs)	3	2
Cycle Time (min)	8	8
V. Mach Cost ($/Unit)	0.65	0.7

9.2.3 System Initial State

The system planning process takes place on a rolling planning horizon basis. If the system data are updated by a new configuration, a new product, or any other major data change, the planning process should be retriggered immediately. Every RMS system defines its own initial state: current configuration loaded for every module, and current product loaded for every configuration. All the subsequent reconfiguration and product loading/unloading decisions are dependent on these values. Table 9-27 reports the RMS initial states, i.e. initial configuration and product IDs for every module. The initial state defines the Bucket 0 for every configuration path.

Table 9-27: RMS Initial State

Mod ID	Configuration ID	Prod ID
1000	1101	101
2000	2201	103
3000	3301	108

9.2.4 Other System Data

Other system data include data related to workface, workforce regulations, and system calendar. ROP is supported by a real calendar that reports exactly the number of working days and the maximum allowed overtime rates. Unlike the APP assumption of equal interval planning buckets, ROP assumes a real calendar because the manufacturing operations are much more complicated than their counterparts in other traditional manufacturing systems. In addition, that made the planning operations as real as possible.

Table 9-28: Workforce Financial Info

Parameter	Units	Value
CF	$/Worker	2000
CH	$/Worker	1500
Cr	$/hr	12
Co	$/hr	18

Table 9-29: Work Regulations: Simple Info

Parameter	Value
Hours/Shift	8
Shift/Day	1

Table 9-30: Work Regulations: Bucket Based Info

	1	2	3	4	5	6	7	8
Days/Bucket	31	28	31	30	31	30	31	31
Max Overtime Hrs/Day	2	2	2	2	2	2	2	2

9.2.5 Optimization Data

The ROP uses the same optimizer that accompanied both the APP and MMAPP applications. The parameters used are illustrated in table 6-31: Alpha is the archive size which is defined as the maximum number the non-dominated solutions maintained at every generation. Mu is the number of parents chosen for reproduction. Lambda is the number of children to join the archive in order to get a better archive in the next generation. Appendix A may be consulted for SPEA2 details.

Table 9-31: Genetic Parameters

Parameter	Value
# of Generations	1000
Alpha	100
Mu	30
Lambda	30

9.3 Results

Like all others case problem presented by this research, ROP is a multi-objective problem, which maintains the balanced system performance perspective presented by CMPC systems. The following table illustrate both a 5-member and a 10-member Pareto front results; the word short stands for having only the values of the objectives. The objectives under study are maximizing profits, minimizing backorders, and minimizing inventory management. Unlike other traditional manufacturing systems, the cost structure of RMS is relatively complicated. Even though it could be considered a compilation of many stepwise and piece-linear parts, the final cost curve could be anything. That was planned for when PM early developed to address any non-linearity that could be an outcome of problem analysis. The post analysis section of this chapter shows the implications of having such kind of non-linearity and the unpredictability of RMS cost function. Minimizing costs may lead to erroneous decisions; profit margins are different form configuration to configuration and from product batch to another. The backorders is intangible objective; it is separated from the financial objective for that reason. The backordering cost could be a good measure for lost sales; however, with PM any other measure could be used with ease. Minimizing inventory investment promotes lean practice and improves bottom line financials. Tying profits, inventory, and backorders together as the driving forces of the search process creates balanced solutions that should be available to decision makers to help them choose the best comprising solution among the best that they can already have. The workforce objective has been omitted because the workforce level is assumed a system design parameter. Every configuration joins a module configuration library define its own optimum workforce in addition to other optimum process parameters. Workforce is only adjusted with the configuration selection process.

During the solution process the number of Pareto front members is maintained to 100; however, this could be unrealistic when presented to a decision maker or a group of decision makers. Short lists of just 5 or 10 could be sufficient. The same truncation algorithm that reduces the number of Pareto front members in case they exceed their desired limit is reused to get a 5 and 10 members Pareto lists reported in Table 9-32 and Table 9-33. For every solution point described in the aforementioned tables, there are corresponding operations plans, product plans, and workforce plans. The remaining list of figures, tables, and charts reports these data for just the first solution point presented. ROP defines many planning structures as already described in chapter 8.

Table 9-32: Pareto Font Short (5 members)

	Financials[Max]	Backorders[Min]	Inventory[Min]
1	806048.9	142590	489979
2	833030.7	779900	285950
3	945472.8	40000	1322726
4	1335506.9	826040	942620
5	1281464.9	1249440	303140

Table 9-33: Pareto Font Short (10 members)

	Financials[Max]	Backorders[Min]	Inventory[Min]
1	1226020.8	285170	736220
2	945661.95	46550	778665
3	1188128.1	438000	340220
4	806048.9	142590	489979
5	1311982.4	806690	495420
6	833030.7	779900	285950
7	1312011.4	633760	1492130
8	945472.8	40000	1322726
9	1335506.9	826040	942620
10	1281464.9	1249440	303140

9.3.1 Configuration Maps

The first planning structure of the ROP problem/system is the operations plans defined by configuration maps. Configuration maps encapsulated all the manufacturing operation of RMS. Figure 9-1 shows a complete configuration map corresponding to the first Pareto front point.

9.3.2 Product Make Plans

Product Make Plans define the individual product operations made by the system. Table 9-34 to Table 9-37 report the product make plans for the first Pareto member.

Table 9-34: Product::101 [Module::1000] Product Make Plan

	1	2	3	4	5	6	7	8
Reg Time	248	224	248	240	215	217	248	248
Reg Quantity	2976	1344	1488	1440	1720	1302	1488	1488
Max Over Time	62	56	62	60	54	56	62	62
Max Ov Quantity	744	336	372	360	432	336	372	372
Ov Time	62	56	62	60	54	56	62	62
Ov Quantity	744	336	372	360	432	336	372	372

Table 9-35: Product::103 [Module::2000] Product Make Plan

	1	2	3	4	5	6	7	8
Reg Time	N/A	114	209	120	202	112	110	204
Reg Quantity	N/A	795	1254	720	1515	840	612	1530
Max Over Time	N/A	28	54	30	52	28	26	52
Max Ov Quantity	N/A	210	324	180	390	210	156	390
Ov Time	N/A	28	54	30	52	28	26	52
Ov Quantity	N/A	210	324	180	390	210	156	390

Table 9-36: Product::106 [Module::2000] Product Make Plan

	1	2	3	4	5	6	7	8
Reg Time	228	38	N/A	100	N/A	105	74	105
Reg Quantity	2736	570	N/A	1104	N/A	1455	888	1455
Max Over Time	58	10	N/A	24	N/A	26	20	26
Max Ov Quantity	696	150	N/A	288	N/A	390	240	390
Ov Time	0	0	N/A	0	N/A	0	0	0
Ov Quantity	107	36	N/A	18	N/A	111	11	9

Table 9-37: Product::108 [Module::3000] Product Make Plan

	1	2	3	4	5	6	7	8
Reg Time	225	224	248	240	248	240	248	224
Reg Quantity	1687	1680	1860	1800	1860	1800	1860	1344
Max Over Time	58	56	62	60	62	60	62	56
Max Ov Quantity	435	420	465	450	465	450	465	336
Ov Time	58	56	62	60	62	60	62	56
Ov Quantity	435	420	465	450	465	450	465	336

	0					1					2					3					4				
M1000	C1101		n/a	0	0	C1101		n/a	248	248	C1101		n/a	224	224	C1101		n/a	248	248	C1101		n/a	240	240
		P101	n/a	0	0	n/a	P101	n/a	248	62	n/a	P101	n/a	224	56	n/a	P101	n/a	248	62	n/a	P101	n/a	240	60
	0					1					2										3				
M2000	C2201		n/a	0	0	C2201		n/a	248	248	C2202							24	200	224	C2201		24	224	248
		P103	n/a	0	0	4	P106	16	228	0	6	P106	20	38	0	6	P103	16	114	28	3	P103	12	209	54
	0					1					2					3					4				
M3000	C3301		n/a	0	0	C3303		6	242	248	C3303		n/a	224	224	C3303		n/a	248	248	C3303		n/a	240	240
		P108	n/a	0	0	2	P108	15	225	58	n/a	P108	n/a	224	56	n/a	P108	n/a	248	62	n/a	P108	n/a	240	60

	5					6					7					8									
M1000	C1102		16	232	248	C1101		8	232	240	C1101		n/a	248	248	C1101		n/a	248	248					
	2	P101	15	215	54	3	P101	12	217	56	n/a	P101	n/a	248	62	n/a	P101	n/a	248	62	n/a				
	4										5					6									
M2000	C2201							n/a	240	240	C2202		24	224	248	C2202							n/a	240	240
	n/a	P103	n/a	120	30	4	P106	16	100	0	6	P103	16	202	52	n/a	P103	n/a	112	28	3	P106	20	105	0
	5					6					7					8									
M3000	C3303		n/a	248	248	C3303		n/a	240	240	C3303		n/a	248	248	C3302		8	240	248					
	n/a	P108	n/a	248	62	n/a	P108	n/a	240	60	n/a	P108	n/a	248	62	2	P108	14	224	56	n/a				

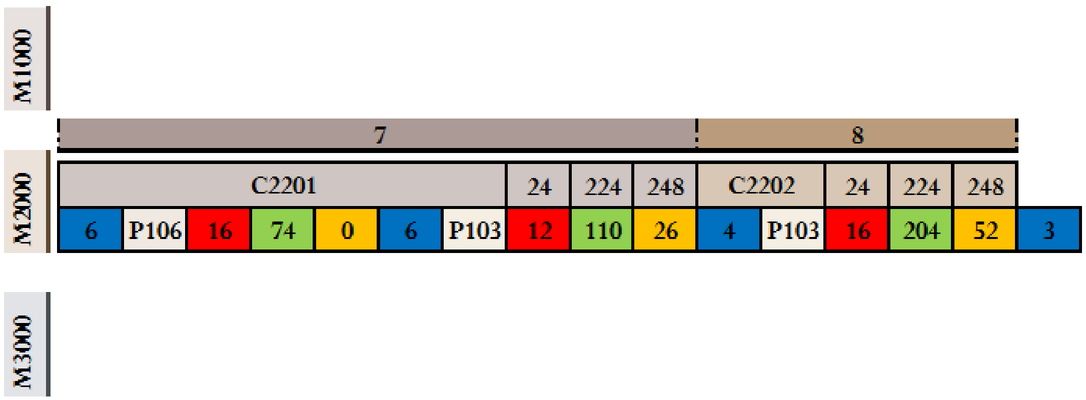


Figure 9-1: Configuration Maps of the first Pareto front member (Profit =806048.9)

9.3.3 Product Plans

Product Plans aggregates product supply plans by system operations, subcontracting plans if any, and Inventory and backordering plans. Table 9-38 to Table 9-41 report the results for the first Pareto member.

Table 9-38: Product 101 Plan

	P. State	1	2	3	4	5	6	7	8
D		2500	4000	2000	1500	6000	3000	1500	4000
Rmax		2976	1344	1488	1440	1720	1302	1488	1488
R		2976	1344	1488	1440	1720	1302	1488	1488
Omax		744	336	372	360	432	336	372	372
O		744	336	372	360	432	336	372	372
S		1020	1633	816	612	2449	1224	612	1633
I	0	40	0	29	941	0	0	376	0
B	2200	0	647	0	0	458	596	0	131

Table 9-39: Product 103 Plan

	P. State	1	2	3	4	5	6	7	8
D		1500	1000	1500	1100	1700	1500	1000	1000
Rmax		0	795	1254	720	1515	840	612	1530
R		0	795	1254	720	1515	840	612	1530
Omax		0	210	324	180	390	210	156	390
O		0	210	324	180	390	210	156	390
S		0	0	0	0	0	0	0	0
I	1300	0	0	0	0	0	0	0	126
B	0	200	195	117	317	112	562	794	0

Table 9-40: Product 106 Plan

	P. State	1	2	3	4	5	6	7	8
D		500	600	700	600	800	700	700	900
Rmax		2736	570	0	1104	0	1455	888	1455
R		2736	570	0	1104	0	1455	888	1455
Omax		696	150	0	288	0	390	240	390
O		107	36	0	18	0	111	11	9
S		0	0	0	0	0	0	0	0
I	0	143	149	0	0	0	37	236	800
B	2200	0	0	551	29	829	0	0	0

Table 9-41: Product 108 Plan

	P. State	1	2	3	4	5	6	7	8
D		3000	2400	1500	1500	3400	2000	1100	2100
Rmax		1687	1680	1860	1800	1860	1800	1860	1344
R		1687	1680	1860	1800	1860	1800	1860	1344
Omax		435	420	465	450	465	450	465	336
O		435	420	465	450	465	450	465	336
S		0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0
B	1000	1878	2178	1353	603	1678	1428	203	623

9.3.4 Work force Plans

Table 9-42-Table 9-44 report the workforce plans for the first Pareto member of individual modules and Table 9-45 reports the workforce plan of the RMS as a whole.

Table 9-42: Module 1000 Workforce Plan

	P. State	1	2	3	4	5	6	7	8
W	10	10	10	10	10	14	10	10	10
H		0	0	0	0	4	0	0	0
F		0	0	0	0	0	4	0	0

Table 9-43: Module 2000 Workforce Plan

	P. State	1	2	3	4	5	6	7	8
W	12	12	16	12	12	16	16	12	16
H		0	4	0	0	4	0	0	4
F		0	0	4	0	0	0	4	0

Table 9-44: Module 3000 Workforce Plan

	P. State	1	2	3	4	5	6	7	8
W	12	16	16	16	16	16	16	16	14
H		4	0	0	0	0	0	0	0
F		0	0	0	0	0	0	0	2

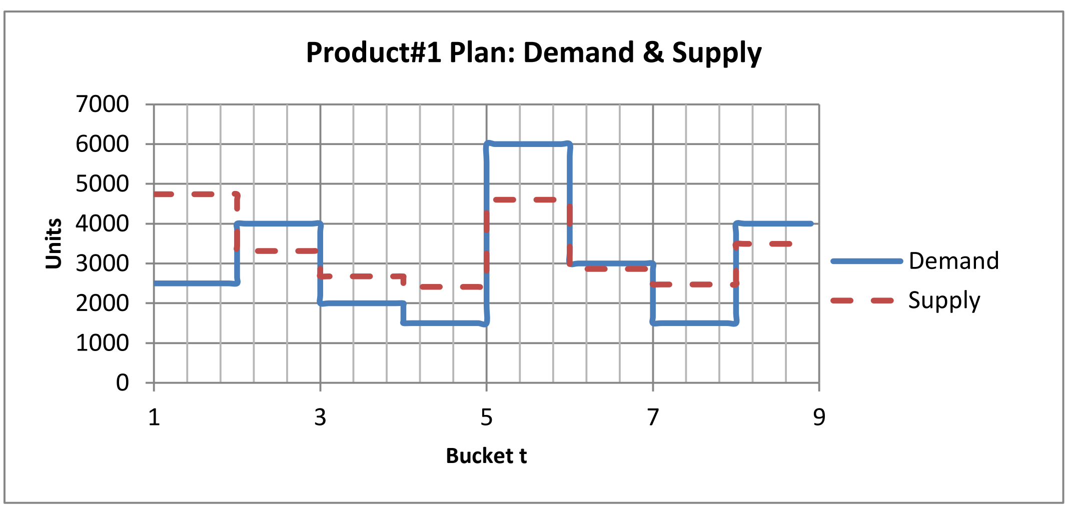
Table 9-45: System Workforce Plan

	P. State	1	2	3	4	5	6	7	8
W	34	38	42	38	38	46	42	38	40
H		4	4	0	0	8	0	0	2
F		0	0	4	0	0	4	4	0

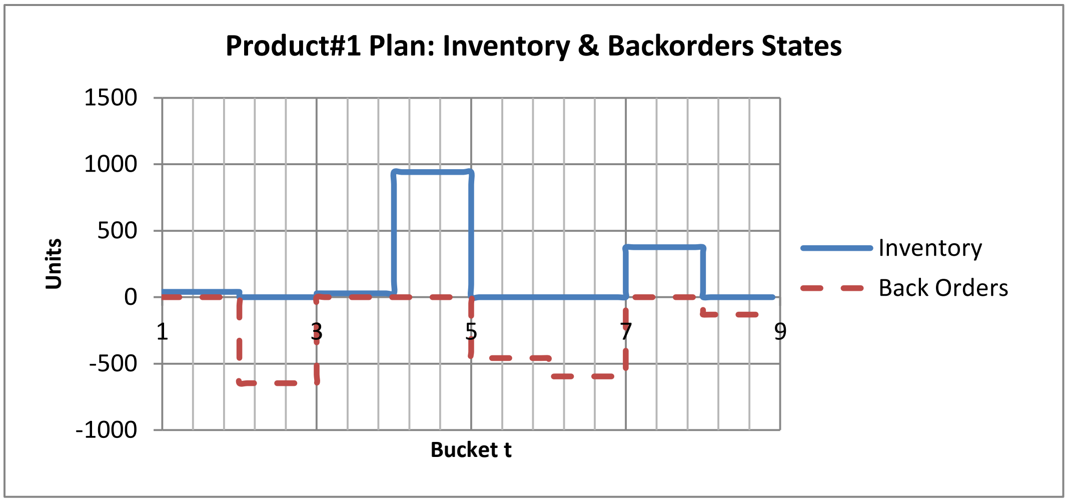
9.3.5 Planning Charts: Demand and Supply, Inventory and Back Orders, and Workforce Adjustments

Demand and supply, and inventory and back orders charts are quick snapshots of the individual planning activities that result from the decisions made by configurations maps, matching the demand and supply, and checking the available system levers. Results reported earlier give the exact figures; charts show the behaviour and the dynamics of the competing demand and supply. Planning charts may reveal major problems that should be planned for instantly; it can show that demand and supply for a certain product are in tandem; it can show there is a huge amount of inventory or backorders are expected over the planning horizon. Charts ring the bells for the decision makers if the current system resources are not enough to create the balance that is needed to respond to unmet demand. Charts given for the first Pareto member show some of the idiosyncrasies that may need further analyses of decision makers. Now a well-developed logic is in service to help to create a highly educated decisions.

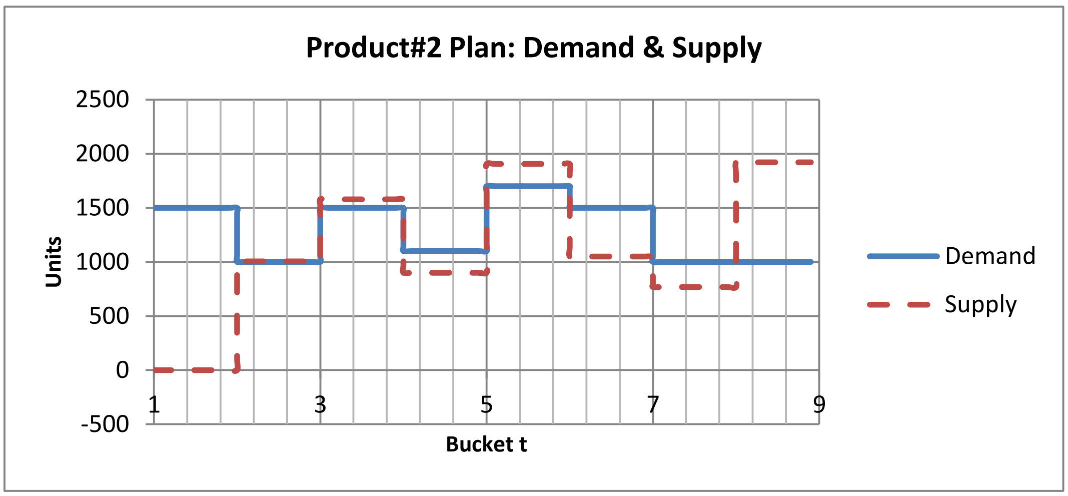
(a)



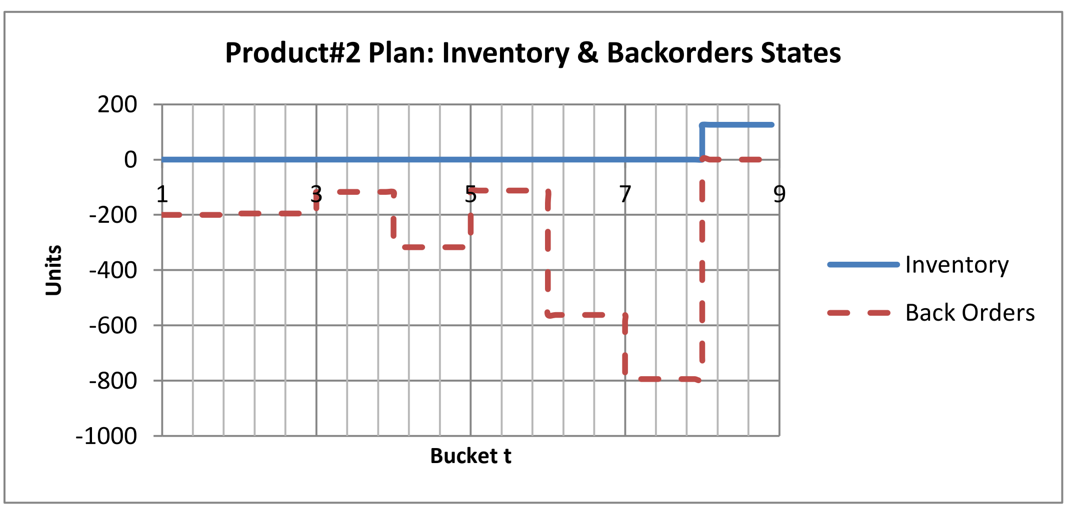
(b)


Product#1 Plan: Inventory & Backorders States
1500
1000
500
0
-500
-1000
Units
1
3
5
7
9
Bucket t
Inventory
Back Orders

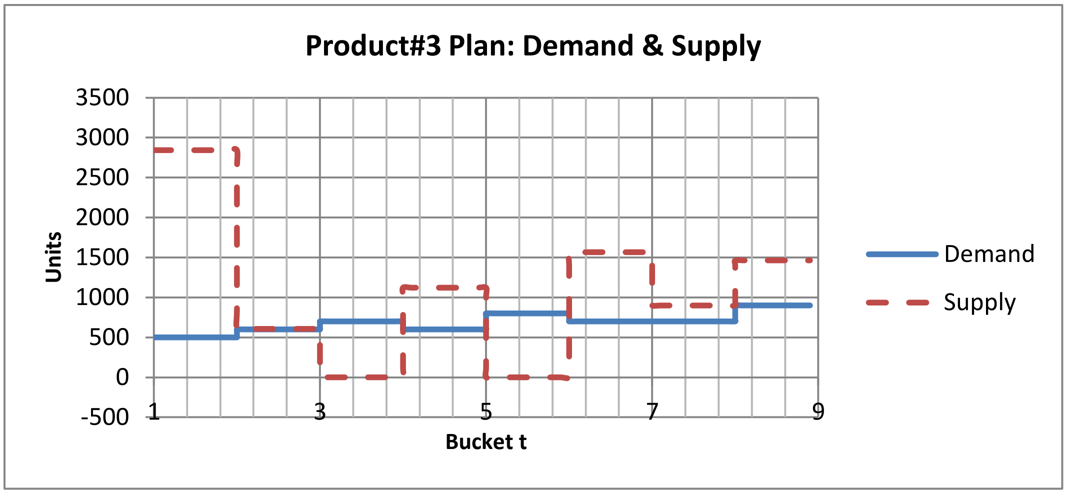
(c)


Product#2 Plan: Demand & Supply
2500
2000
1500
1000
500
0
-500
Units
1
3
5
7
9
Bucket t
Demand
Supply

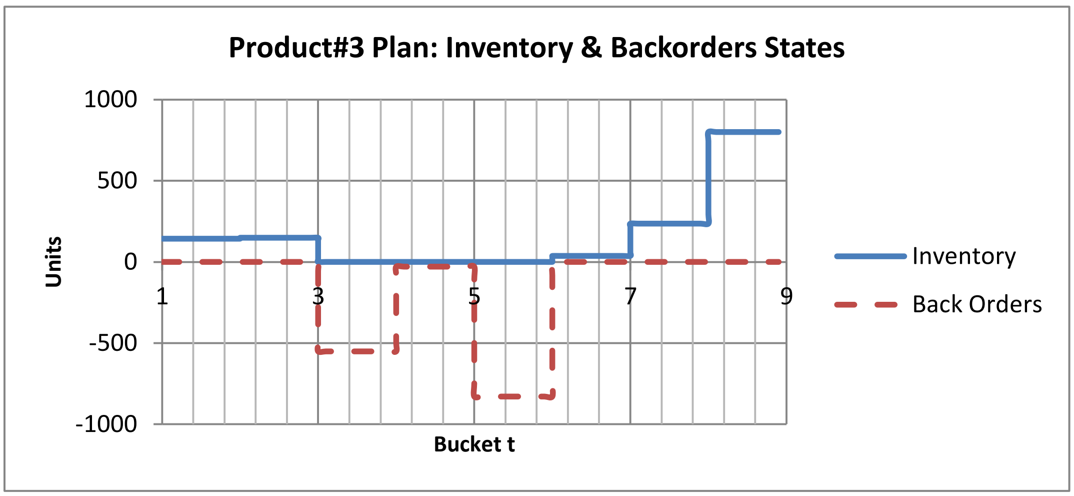
(d)


Product#2 Plan: Inventory & Backorders States
200
0
-200
-400
-600
-800
-1000
Units
1
3
5
7
9
Bucket t
Inventory
Back Orders

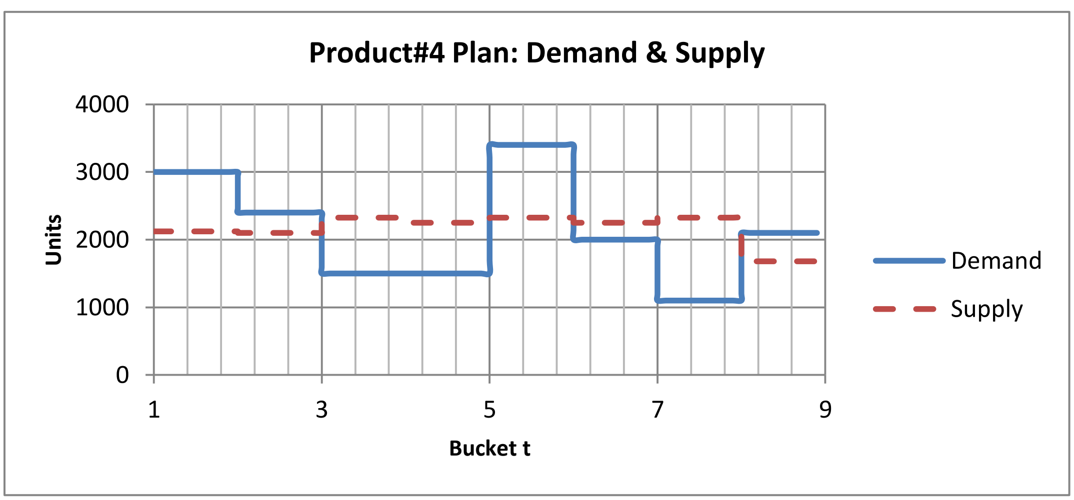
(e)


Product#3 Plan: Demand & Supply
Units
3500
3000
2500
2000
1500
1000
500
0
-500
1
3
5
7
9
Bucket t
Demand
Supply

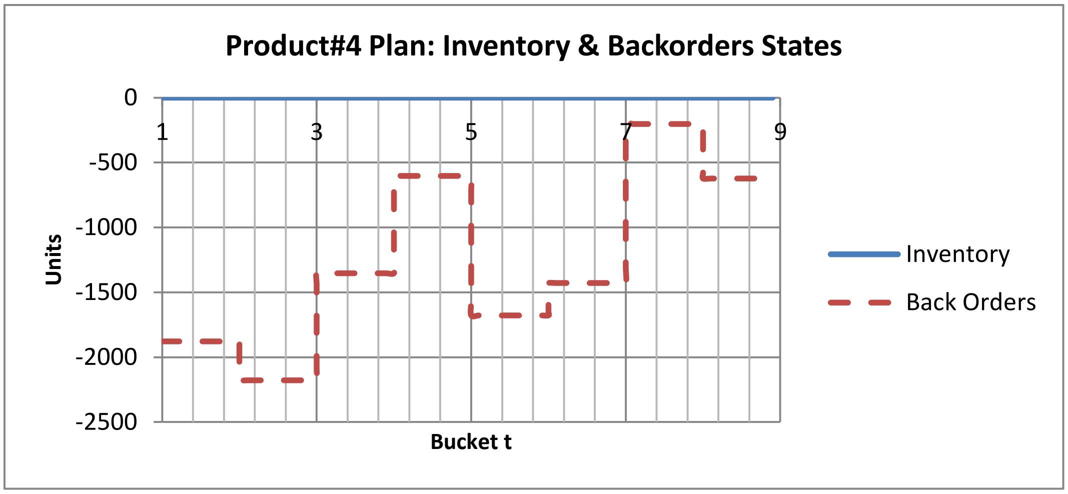
(f)


Product#3 Plan: Inventory & Backorders States
Units
1000
500
0
-500
-1000
1
3
5
7
9
Bucket t
Inventory
Back Orders

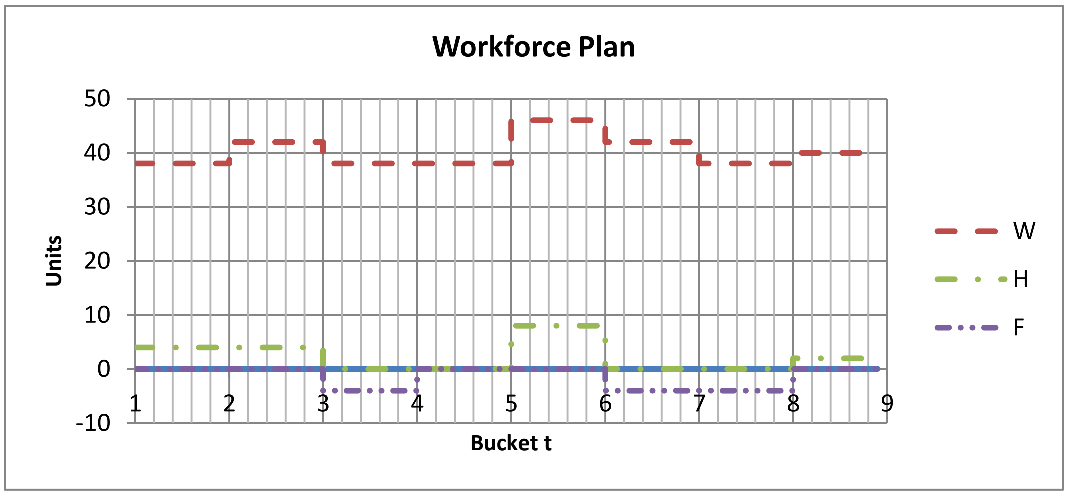
(g)


Product#4 Plan: Demand & Supply
Units
4000
3000
2000
1000
0
1
3
5
7
9
Bucket t
Demand
Supply

(h)



(i)



(j)

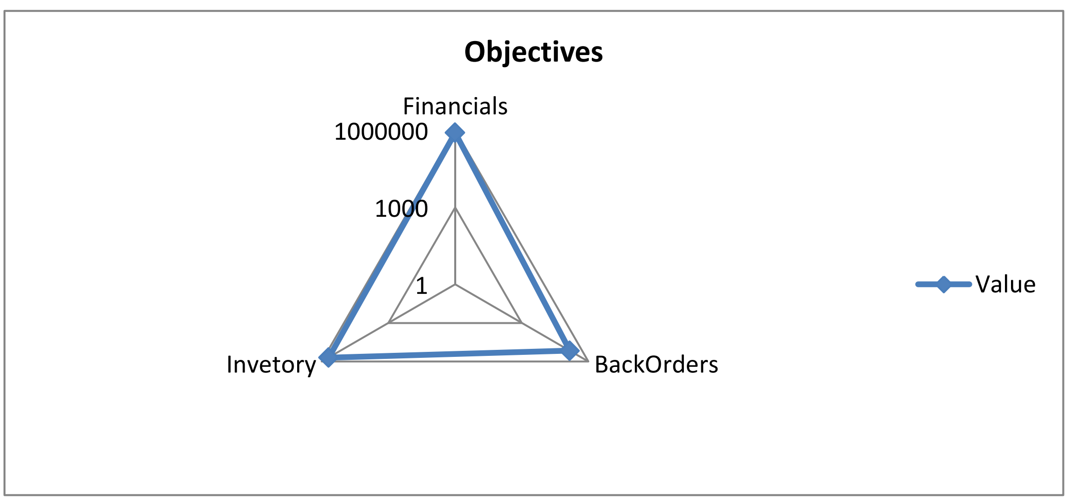


Figure 9-2: ROP Charts: (a-h) Product supply and demand/Inventory Backorders Plans, (i) workforce plans, and (j) Objectives values radar chart

9.3.6 Objectives Evaluation in Details

Table 9-46 to table 9-48 describe the operations cost evaluations. All the tables are banded according to configuration buckets defined. As already reported in chapter 7, operations costs include reconfiguration costs, product set up/unloading costs, production activities costs (both regular and overtime), materials costs, machining costs, and payroll costs. Many cost ratios can be developed, for example, the reconfiguration costs for module 1000 is equivalent to 3.4%, operations costs is 93.3%, and workforce costs 3.3%.

Even though reconfigurable manufacturing systems are very promising technology and can provide the capacity needed when needed, the new system brings a long list of challenges. RMS, as already reported in many places in this dissertation, was the first catalyst to find a new modeling approach that can capture the new demanding and immature manufacturing process. This case study and its results prove that the mission is now a success. With PM, RMS became real and now it can be treated as if it were a well-established system.

A product cost in the new environment cannot be exactly estimated unless the r-bucket is defined first. Within the context of an r-bucket, the cost of reconfigurations can be allocated to manufactured products and the exact profit contribution margin can evaluated. Estimating the product costs (cost/unit) is one of out of box findings brought by the R-bucket. All R-buckets are banded in the product make cost tables. The p-bucket is concerned with product selection, batch size, and overtime quantities. p-buckets can be traced using the same cost tables or the original configuration maps.

9.3.6.1 Product Make Costs (Operating Costs)

Table 9-46: Path 1000 Product Make Costs

R.Bucket #	Config ID	Time	Cost	P.Bucket #	Slot #	Prouduct #	Cm	Cr	Co	Set-up		Reg. Prod				Ov. Prod			Unloading		Cost	WorkForce					
										Time	Cost	Time	Quant	V. M/C Cost	R. Cost	Time	Quant	O. Cost	Time	Cost		CH	CF	H	F	T.Cost	
1	C1101	N/A	N/A	0	1	101	20	12	18	N/A	N/A	0	0	0.7	0	0	0	0	N/A	N/A	0	1500	2000	N/A	N/A	N/A	
				1	2	101	20	12	18	N/A	N/A	248	2976	0.45	63835.2	62	744	16330.8	N/A	N/A	80166	1500	2000	0	0	0	
				2	3	101	20	12	18	N/A	N/A	224	1344	0.7	30508.8	56	336	7963.2	N/A	N/A	38472	1500	2000	0	0	0	
				3	4	101	20	12	18	N/A	N/A	248	1488	0.7	33777.6	62	372	8816.4	N/A	N/A	42594	1500	2000	0	0	0	
				4	5	101	20	12	18	N/A	N/A	240	1440	0.7	32688	60	360	8532	2	1000	42220	1500	2000	0	0	0	
2	C1102	16	9600	5	1	101	20	12	18	15	10000	215	1720	0.55	37926	54	432	9849.6	3	1500	59275.6	1500	2000	4	0	6000	
3	C1101	8	4800	6	1	101	20	12	18	12	6000	217	1302	0.7	29555.4	56	336	7963.2	N/A	N/A	43518.6	1500	2000	0	4	8000	
				7	2	101	20	12	18	N/A	N/A	248	1488	0.7	33777.6	62	372	8816.4	N/A	N/A	42594	1500	2000	0	0	0	
				8	3	101	20	12	18	N/A	N/A	248	1488	0.7	33777.6	62	372	8816.4	N/A	N/A	42594	1500	2000	0	0	0	
Totals			14400																		391434.2					14000	419834.2

Table 9-47: Path 3000 Product Make Cost

R.Bucket #	Config ID	Time	Cost	P.Bucket #	Slot #	Prouduct #	Cm	Cr	Co	Set-up		Reg. Prod				Ov. Prod			Unloading		Cost	WorkForce					
										Time	Cost	Time	Quant	V. M/C Cost	R. Cost	Time	Quant	O. Cost	Time	Cost		CH	CF	H	F	T.Cost	
1	C3301	N/A	N/A	0	1	108	19	12	18	N/A	N/A	0	0	0.8	0	0	0	0	2	1200	1200	1500	2000	N/A	N/A	N/A	
2	C3303	6	3600	1	1	108	19	12	18	15	14000	225	1687	0.7	35933.9	58	435	9613.5	N/A	N/A	59547.4	1500	2000	4	0	6000	
				2	2	108	19	12	18	N/A	N/A	224	1680	0.7	35784	56	420	9282	N/A	N/A	45066	1500	2000	0	0	0	
				3	3	108	19	12	18	N/A	N/A	248	1860	0.7	39618	62	465	10276.5	N/A	N/A	49894.5	1500	2000	0	0	0	
				4	4	108	19	12	18	N/A	N/A	240	1800	0.7	38340	60	450	9945	N/A	N/A	48285	1500	2000	0	0	0	
				5	5	108	19	12	18	N/A	N/A	248	1860	0.7	39618	62	465	10276.5	N/A	N/A	49894.5	1500	2000	0	0	0	
				6	6	108	19	12	18	N/A	N/A	240	1800	0.7	38340	60	450	9945	N/A	N/A	48285	1500	2000	0	0	0	
				7	7	108	19	12	18	N/A	N/A	248	1860	0.7	39618	62	465	10276.5	2	1400	51294.5	1500	2000	0	0	0	
3	C3302	8	4800	8	1	108	19	12	18	14	12000	224	1344	0.75	29232	56	336	7644	N/A	N/A	48876	1500	2000	0	2	4000	
Totals			8400																		402342.9					10000	420742.9

Table 9-48: Path 2000 Product Make Cost

R.Bucket #	Config ID	Time	Cost	P.Bucket #	Slot #	Prouduct #	Cm	Cr	Co	Set-up		Reg. Prod				Ov. Prod			Unloading		Cost	WorkForce					
										Time	Cost	Time	Quant	V. M/C Cost	R. Cost	Time	Quant	O. Cost	Time	Cost		CH	CF	H	F	T.Cost	
1	C2201	N/A	N/A	0	1	103	17	12	18	N/A	N/A	0	0	1.2	0	0	0	0	4	3000	3000	1500	2000	N/A	N/A	N/A	
				1	2	106	20	12	18	16	15000	228	2736	0.8	59644.8	0	94	1955.2	6	3500	80100	1500	2000	0	0	0	
2	C2202	24	14400	2	1	106	20	12	18	20	20000	38	570	0.75	12283.5	0	20	415	6	4500	75160.5	1500	2000	4	0	6000	
						103	17	12	18	16	16000	114	795	1	15678	28	210	4284	3	2000							
3	C2201	24	14400	3	1	103	17	12	18	12	12000	209	1254	1.2	25330.8	54	324	6868.8	N/A	N/A	44199.6	1500	2000	0	4	8000	
				4	2	103	17	12	18	N/A	N/A	120	720	1.2	14544	30	180	3816	4	3000	64834.4	1500	2000	0	0	0	
						106	20	12	18	16	15000	100	1104	0.8	24163.2	0	39	811.2	6	3500							
4	C2202	24	14400	5	1	103	17	12	18	16	16000	202	1515	1	29694	52	390	7956	N/A	N/A	53650	1500	2000	4	0	6000	
				6	2	103	17	12	18	N/A	N/A	112	840	1	16464	28	210	4284	3	2000	79799	1500	2000	0	0	0	
						106	20	12	18	20	20000	105	1455	0.75	31451.25	0	53	1099.75	6	4500							
5	C2201	24	14400	7	1	106	20	12	18	16	15000	74	888	0.8	19358.4	0	33	686.4	6	3500	69310.4	1500	2000	0	4	8000	
						103	17	12	18	12	12000	110	612	1.2	12458.4	26	156	3307.2	4	3000							
6	C2202	24	14400	8	1	103	17	12	18	16	16000	204	1530	1	29988	52	390	7956	3	2000	55944	1500	2000	4	0	6000	
Totals			72000																		525997.9					34000	631997.9

9.3.6.2 Other Objective Evaluations

Table 9-49: Product::101 Objective Evaluation

	1	2	3	4	5	6	7	8	Total
S	1020	1633	816	612	2449	1224	612	1633	9999
Subcontracting Cost	35700	57155	28560	21420	85715	42840	21420	57155	349965
I	40	0	29	941	0	0	376	0	1386
Holding Cost	20	0	14.5	470.5	0	0	188	0	693
Inventory Investment	2000	0	1450	47050	0	0	18800	0	69300
B	0	647	0	0	458	596	0	131	1832
Backordering Objectives	0	25880	0	0	18320	23840	0	5240	73280

Table 9-50: Product::103 Objective Evaluation

	1	2	3	4	5	6	7	8	Total
S	0	0	0	0	0	0	0	0	0
Subcontracting Cost	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	126	126
Holding Cost	0	0	0	0	0	0	0	75.6	75.6
Inventory Investment	0	0	0	0	0	0	0	5040	5040
B	200	195	117	317	112	562	794	0	2297
Backordering Objectives	5400	5265	3159	8559	3024	15174	21438	0	62019

Table 9-51: Product::106 Objective Evaluation

	1	2	3	4	5	6	7	8	Total
S	0	0	0	0	0	0	0	0	0
Subcontracting Cost	0	0	0	0	0	0	0	0	0
I	143	149	0	0	0	37	236	800	1365
Holding Cost	71.5	74.5	0	0	0	18.5	118	400	682.5
Inventory Investment	7150	7450	0	0	0	1850	11800	40000	68250
B	0	0	551	29	829	0	0	0	1409
Backordering Objectives	0	0	22040	1160	33160	0	0	0	56360

Table 9-52: Product::108 Objective Evaluation

	1	2	3	4	5	6	7	8	Total
S	0	0	0	0	0	0	0	0	0
Subcontracting Cost	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0
Holding Cost	0	0	0	0	0	0	0	0	0
Inventory Investment	0	0	0	0	0	0	0	0	0
B	1878	2178	1353	603	1678	1428	203	623	9944
Backordering Objectives	56340	65340	40590	18090	50340	42840	6090	18690	298320

9.4 Conclusions

PM brought RMS from just a system that promises responsiveness via capacity options to a real system with a full range of levers such as capacity, inventory, overtime, and subcontracting. Many innovations have been brought in order to address the new challenges posed by the new technology. ROP demonstrates what PM can contribute to the modeling and development of RMS systems. ROP is a typical example of the new generation of large-scale problems that PM can bring to the Operations management field.

9.5 Summary

In this chapter, a case study was presented to illustrate the foundations, models developed, solution algorithms throughout the last three chapters by an example. Results showed that PM brought RMS to reality, made it change ready, made it work within the latest PM managerial sphere, the optimized tandem (balanced objectives and well-orientated system levers). From ROP perspective, the system, the product, the process, and the people are now in tandem.

Chapter 10 SUMMARY, INSIGHTS, AND CONTRIBUTIONS

10.1 Introduction

The last couple of decades have witnessed a level of fast-paced development of new ideas, products, manufacturing technologies, manufacturing practices, customer expectations, and civilization movements as it has never been before. Change became the intrinsic characteristic that is addressed everywhere. How to deal with change, how to manage it, how to bind to it, how to steer it, how to create value out of it were the early questions at the early days of this research. The early objective was to develop a manufacturing planning and control system for reconfigurable manufacturing systems. The first initiative was to create an evolvable loosely coupled MPC framework that is able to catch the pace with the underlying evolving system. With such a system, market, system, products, processes, and workforce are all in the state of change. The term “Change Ready MPC Systems” was coined, Component Based Software Engineering was introduced as enabling technology, and some characteristics were presented to define the dynamics that control CMPCs behaviour. In order to limit the scope to just one problem or CMPC component and to make its internal logic change ready, the aggregate production planning (APP) problem was chosen as a case problem. In fact, studying the APP in the RMS context was the original research project of this research. Unfortunately, the problem suffers severely from the lack of applicability in the industrial domain; hundreds of academic papers and there is no realistic application. The problem epitomized what was articulated by this research as the academic-industrial gap. Sometimes, we create the problems that we are able to solve not the problem that the world asks us to solve; instead of answering the question asked, we develop our own question and answer it. Since that time, a new vision of developing a modeling approach that reduces the gap between the idealistics of the academia and the pragmatics of the industry was determined. The objective was to develop a modeling approach that capture the future even if it is hard to quantify or grasp; a modeling approach that is evolving by nature and gives us the time to understand, implement, review, and change our thought again and so on. Function templates and systemizing problems were the early gadgets. The single product aggregate production planning was the first application. The next step was to work on the solution algorithms and make them evolvable especially if the problem at hand is highly constrained. Several concepts were developed to make progressive algorithms a super-set of already existing algorithms with a basic rule, to break those algorithms rules themselves if necessary. Finally, the time has come to develop a model for the aggregate production planning in an RMS environment. Unfortunately, the system itself can change its structure, which was considered an extra production planning lever, i.e. system reconfiguration. Influenced by many philosophies and advancements accompanied CMPC and PM developments, the new problem of Reconfiguration and Operations Planning was introduced. Both the problem scope and size are unprecedented in the RMS literature. ROP managed to address an armada of challenges posed by the new RMS technology. The ROP became the greatest ad for the new product: Progressive Modeling. Instead of developing a model for an APP problem in the RMS context, this research ended up by a new CMPC framework, a novel modeling approach, and three different application problems.

This chapter should be short by nature; some research insights are reported first, major research contributions and achievements are highlighted, Implementation tools used are summarized, and finally the directions for Progressive Modeling development is highlighted.

10.2 Research insights

During the long journey of this research, some insights came to the surface that could be described here

- In some domains, there is a gap between the academic and industrial worlds. In the industry, they want it practical, simple, very quick, and creates value. In the academia, they want it sophisticated and reflect that they are better-educated and problem solvers. As academicians it is an opportunity to prove we are well-educated and as for the industrialists a value can be created here.
- So many disciplines of knowledge have been developed over the years; the multidisciplinary research unleashes a new scientific regime that can define a new world of possibilities; linking disciplines in a synergistic way is not an easy task, links need to be identified or created if necessary. PM is founded on an army of synergized multidisciplinary tools that can address a new set of world problems.
- When it comes to an engineering problem it should be a cost problem, this what the literature mostly reports. In industry the bottom line financials such as net profit and return on assets, satisfied customers, strong value chain, sustainable system stability and so on is what makes corporate value. The system perspective and holistic approaches became necessary. A new paradigm in the academic literature has to start over. Progressive Modeling brings the notion of balance. Systems are counting on many levers; they can be orientated in so many ways without losing neither the focus nor the direction. As long as all the levers are in a balanced state, it would never be only a matter of a cost objective.
- Reconfigurable Manufacturing literature has suffered from so many problems due to the immaturity of the new technology. It is hard to define the logic and it is harder to test that logic. Everything became dynamic and willing to change.

Change Ready MPC systems and Progressive Modeling are developed in order to serve the RMS and now they are ready for RMS and others.

10.3 Research Contributions and Achievements

Throughout this research, many contributions and achievements have been made at many hierarchical levels. Under the umbrella of each of them, many contributions and achievements have been made.

10.3.1 Change Ready MPC systems

CMPC is a new vision of how to make manufacturing planning and control systems ready for change that may be created via development or mitigated as a threat originated from within or the surrounding environment.

Contribution and Achievements:

1. **Component Based Software Engineering (CBSE)** has been introduced as a technological enabler and architectural tool of CMPC systems and components. CBSE concepts inspired function templates and have some implications on master solution algorithms that used in solving MMAPP and ROP. In addition, CBSE has a great impact on systemizing problems, analyzing them, and developing larger models and their associated algorithms.
2. **Change Drivers** have an impact on how developed models and algorithms could be designed to be ready for their anticipated and unanticipated changes in their working environments. The outcome is more agile, more applicable, evolvable, and sustainable models.
3. **CMPC Characteristics:** a well-revised set of change ready MPC characteristics have been introduced to define new MPC frameworks that have been endorsed by many innovations of the novel modeling approach (PM) as well. Now, a new generation of distinguished systems and models are change ready by nature in terms of architecture, logic, embedded algorithms, and development.

10.3.2 Progressive Modeling

PM is an innovative multidisciplinary modeling approach that has been developed to better model industrial problems in a practical and modern way without losing the scientific rigor. Several concepts introduced in terms of analyzing industrial problems and dividing them into smaller manageable ones.

Contribution and Achievements

1. **PM brings a new generation of large-scale industrial problems:** from analysis where problems can be divided into smaller and manageable ones, passing by the novel mathematical statements, and ending by the novel chained search space, a new generation of large-scale industrial problems can be defined.

2. **The PM Process** formalizes the problem modeling, makes it more flexible, more generic, forward-looking, and progressive. PM process is about analytics, logic, and alternatives that we can select form. The process has an utmost generality level. It can be applied to both small and large problems regardless of tools used and domains of application.

3. **Componentizing Problems** contribute to simplifying problem solution process and facilitating better model development process.

4. **The Notion of Balance** has a direct impact on problems developed: all problems have been defined as multi-objective problems. The notion has extended to define the system levers that drive the system performance. The ROP epitomizes the notion of balance both in its two-dimensional definition. By applying this notion and using PM advancements, better operations management systems can be developed and optimized.

5. **Propagation of Modularity:** PM brings modularity to problems analysis and solutions in order to capture the intricacies and complexities of real world problems. Many innovation gadgets like componentizing problems, structuring

search space, separation of concerns and others that made a progressive modeling a reality owes a lot to such a modular thinking approach.

6. **Function Templates:** function templates brought futuristic perspective to mathematical models. The reason to resort to defining function templates is either the lack of knowledge or the lack of understanding. many implementations could be defined, tested, further developed, and updated. Function templates are intrinsic for the evolvability of mathematical models.
7. **Mathematical Statements:** When systems are modeled, many cascaded or interconnected problems can arise. Now, a compilation of mathematical models that encompasses a whole system including its components and its links to the surrounding environment can be defined.
8. **Advanced Notation** brings the concept of IDs to industrial problems, defines a more lucid symbolic system to define them, and is instrumental in developing large-scale mathematical models.
9. **Tuplezid Nomenclatures:** PM defines a new generation of large-scale problems where smaller ones can be defined. Tuplized nomenclature makes advanced notation symbols grouped to define smaller problem contexts. Advanced notation and tuplezid nomenclatures are basic building blocks in defining the novel mathematical statements.
10. **Data models:** when addressing new problems like the ROP, a data model is needed. Data models are foundational in defining mathematical statements.
11. **Separation of Concerns:** Separations of concerns epitomized in modularized components, separating demand management from operations management, utilized nomenclature, and structured decision space. Separation of concerns guided the modeling process of many case problems presented by this research.
12. **Model Deployment** models are deployed into manageable chunks of logic and assembled of smaller pieces of logic or mini-models. ROP expanded the notion to

encompass the nomenclature. modular logic is another form of separation of concerns.

13. **Coupler and Progressive Algorithms:** In a chained search space, a coupler is something like grey elliptical rectangles of a flow chart. They are capsules of micro-heuristics that could be developed further and make those algorithms more progressive and more efficient. Another use of couplers is model development; making micro changes at mathematical models lead to just changing the corresponding couplers.
14. **Structured Search Space:** large-scale problems may define multitudes of both decision and state variables. Structuring these variables is needed in order to manage them and consequently improve both the efficiency and effectiveness of the search process.
15. **Chained Search Space** is defined for the novel structured search space where a society of decision structures can be connected via system constraints.
16. **Incomplete Chromosome Definitions** was an earlier advancement that brought the chained search space.
17. **Introducing State Machines in the Search Space** is instrumental in the recombination process of the chained search space.
18. **Component-Based Master Algorithms are** interface-based solution algorithms that are compiled of an array of algorithms. Optimizers and modellers are instrumental in decoupling the decision space from the objective space.
19. **EMO:** the Evolutionary Multi-objective Optimization was introduced as a tool to optimize system performance via keeping a record of well-balanced system performance measures or objectives. The approach eliminates the need of pre-processing of data and guides the solution(s) search process into specific prejudiced areas.

20. **System Envelop Constraints:** Systems envelop constraints are instrumental in creating a feasible chained search space. Utilizing them and using state machines define a very fast search process in the chained search space.

21. **Hierarchical Binaries create new generation of hierarchically structured problems**. The ROP in a multi-product environment is a typical example of a hierarchical sequencing problem in the Operations Management field. Sequencing configurations(r-buckets) and then sequencing products on the top of these configurations (p-buckets).

10.3.3 Applications

10.3.3.1 Aggregate Production Planning

Aggregate production planning problem has been presented in a couple of variants to demonstrate how the new concepts developed in this research can add to a better understanding, analysis, modeling, and encapsulating a competitive advantage at the developed MPC models. Numerical examples given are used to validate the new approaches and concepts and show how it can outperform their counterparts in terms of modeling quality and efficient performance.

Contribution and Achievements

1. **Systemized APP models:** PM redefined both APP problem variants from system perspective, they became forward looking, and now they are much easier to adapt and act as pragmatic tools in the production planning field.

2. **Best industrial practices:** PM linked APP objectives to the best industrial practices. Both agile, lean, best system financials and system stability have been linked to problem objectives. A novel workforce changeability objective was introduced in order to imitate the best industrial practice in hiring and firing people in lump sums.

3. **Componentized Models** were presented for both problem variants.

4. **A common mistake related to APP set up decisions** was fixed; the set-up decisions and times of different products were estimated regardless of products sequence as long as there is a quantity produced during a planning bucket. In this study, all the setup decisions and times are accurately estimated.
5. **Function templates** were presented and implemented for the first time.
6. **System Envelop Constraints** were introduced to MMAPP formulation for the first time.
7. **Novel progressive algorithms** were presented (EMO, Incomplete chromosome definitions, couplers, state machines).

10.3.3.2 Reconfiguration and Operations Planning Problem

Contribution and Achievements

1. **A New Problem:** The scope and purpose of the reconfiguration and operations planning problem is unprecedented in both RMS and operations management literature. A problem like ROP could have never existed without developing Progressive Modeling first.
2. **New Foundations:** many foundations related to reconfigurable manufacturing science were presented for the first time: 1) Product Make Life cycle 2) Reconfiguration Life Cycle 3) the R-bucket 4) The P-bucket 5) Operations bucket 6) Configuration path (operations bucket version) 7) Configuration maps 8) Product related plans in an RMS environment: product supply, product operation, inventory, and backorders.
3. **New Holistic Manufacturing Model:** For the first time, a holistic manufacturing model that captures many intricacies of reconfigurable manufacturing process was presented. The ROP closed a missing loop in analyzing and understanding RMS operations.

4. **Promising Modeling Technology:** Progressive Modeling brings a new modeling technology able to define many problems and revise many models developed in the RMS literature. The cost models developed will play a pivotal role in justifying the economic feasibility of RMS as an alternative technology. Further potentials also can be extended to system design and system performance studies.

5. **A Novel ROP Mathematical Statement** was presented to define the logic behind the ROP. The model developed is unprecedented neither in scope nor in size. ROP model proves that the immediate capacity lever still plays a pivotal role in the next generation manufacturing technology. The ROP model shows how all these all levers can be orientated in many ways to produce better and economically justified responsive solutions.

6. **An integrated planning system in RMS environment:** ROP is an integrated operations management system in an RMS environment. RMS has been treated as a real system and many intricacies related to its amorphous process have been pinpointed.

7. **Seminal Solution Algorithms:** the ROP solution algorithm reflects the state-of-the-art of progressive modeling. Configurations paths, configuration maps, product related plans, workforce plans are ROP decisions structures. Many algorithms and operators have been developed to create, recombine, and optimize many decision structures in order to optimize the system performance.

8. **Novel ROP Objective Statement:** the system performance is now presented with the novel PM objectives statement, where both implicit and explicit objectives can be defined. No metric is needed. With PM, RMS became real systems.

9. **Case Study:** The case study is comprehensive and was an almost realistic test bench of the ROP problem and its associated logic and solution approach. The case study data and just one solution-point results were described in a separate

whole chapter, a typical example of the computational power that PM can bring to both optimization and operations management fields.

10. **ROP and PM Modeling and Computational Power:** ROP encompass a group of NP-hard problems: a configuration sequencing problem (NP-hard), product sequencing problem (NP-hard), scheduling problem (NP-hard), a multi-objective problem (NP-hard), lot sizing problem, an implicit goal programming (system envelop constraints), etc; non linearity assumption has been made about decision variables and all of them are integer/binaries; structured search space, chained one, couplers, state machines, carefully designed hierarchical binaries proves both the modeling and the computational power of PM as a novel modeling approach which make it the master contribution of this research.

The aforementioned contributions list just summaries major research contributions and achievements made by this dissertation. The interested reader can refer to many concepts, innovative PM gadgets, and ROP foundations in their original locations for further details.

10.4 Implementation

All the code related to problems presented in this research, APP, MMAPP, and ROP, has been implemented using the C# programming language version 3.0 and the .NET framework 3.5. The code and the logic behind were designed using a mix of component-based programming and object oriented programming principles. All the problem data have been hard coded, and all the results have been reported to Microsoft Excel, which acted as a COM server and a charting engine to report and illustrate results. In order to test the quality of results obtained, the code developed has been instrumented by so many assertion statements to make sure that there is no single constraint is violated. In separate testing sessions, all calculations have been simulated and sent to Excel step-by-step as a different approach to double-check all of them. Every application problem has its own modeller component. Both the APP and MMAPP share the basic or extended versions of Workforce and Product components. ROP has its separate RMS component that encapsulates the entire data model objects described in chapter 6. All the problem applications share the same optimizer component that encapsulates SPEA2 as the alternatives selection algorithm. The latest machine used to run the code has an Intel® Core™ i7 - 740QM, 1.73GHz, and 6GB of RAM. The average run time of 5 consecutive runs are 1:23 and 1:41, and 3.21 min:sec for APP, MMAPP, and ROP respectively. These computational times are estimated for 1000 generations, 100 individuals population size, and 30% of portents are recombined during every generation. By removing all assertion statements reported earlier, all these times should have lower values. Robustness is traded for efficiency throughout all applications developed in order to make sure that all the results are correct and reproducible.

10.5 The Future of Progressive Modeling

Usually, dissertations end up by discussing the future research of the problem at hand considering the problem addressed was solved in earlier chapters. With Progressive Modeling, the solution we get could be better whether by improving the solution algorithm or by improving our understanding. Progressive Modeling is created to solve many problems with whatever the challenges that we might encounter. It is the common answer that came before many questions; what if the answer is not that satisfactory, it should be developed further; that is why it was called Progressive Modeling from the very beginning. The future of Progressive Modeling is very simple—to stay progressive. The notion of optimized tandem and large-scale applications are the major areas for the next PM advancements.

REFERENCES

Abbasi, M. and M. Houshmand (2009). "Production planning of reconfigurable manufacturing systems with stochastic demands using tabu search." International Journal of Manufacturing Technology and Management **17**: 125-148.

Abdi, M. R. and A. W. Labib (2003). "A design strategy for Reconfigurable Manufacturing Systems (RMSs) using Analytical Hierarchical Process (AHP): a case study." International Journal of Production Research **41**: 2273-2299.

Abdi, M. R. and A. W. Labib (2004). "Feasibility study of the tactical design justification for reconfigurable manufacturing systems using the fuzzy analytical hierarchical process." International Journal of Production Research **42**: 3055-3076.

Abdi, M. R. and A. W. Labib (2004). "Grouping and selecting products: The design key of Reconfigurable Manufacturing Systems (RMSs)." International Journal of Production Research **42**: 521-546.

Aiping, L. and L. Chao (2009). Reconfigurable Manufacturing System Modeling Method based on object-oriented technology. 2009 IEEE International Conference on Mechatronics and Automation, 9-12 Aug. 2009, Piscataway, NJ, USA, IEEE.

Akinc, U. and G. M. Roodman (1986). "A new approach to aggregate production planning." IIE Transactions **18**: 88-94.

Armacost, R. L., R. J. Penlesky, et al. (1990). "Avoiding problems inherent in spreadsheet-based simulation models: an aggregate planning application." Production and Inventory Management Journal **31**: 62-68.

Artiba, A. and E. H. Aghezzaf (1997). "An architecture of a multi-model system for planning and scheduling." International Journal of Computer Integrated Manufacturing **10**(5): 380-393.

Barman, S. and E. E. Burch (1989). "The production switching heuristic: A practical revision." International Journal of Production Research **27**: 1863-1875.

Baykasoglu, A. (2001). "MOAPPS 1.0: aggregate production planning using the multiple-objective tabu search." International Journal of Production Research **39**(16): 3685-3702.

Beckmann, M. J. (1961). "Production smoothing and inventory control." Operations Research **9**: 456-467.

Bergstrom, G. L. and B. E. Smith (1970). "Multi-item production planning - An extension of the HMMS rule." Managemet Science **16**: 614-629.

Berry, W. L. and T. Hill (1992). "Linking systems to strategy." International Journal of Operations & Production Management **12(10)**: 3-15.

Bhattacharya, A. K. and J. L. Coleman (1994). "Linking manufacturing strategy to the design of a customized hybrid production control system." Computer-Integrated Manufacturing Systems **7**: 134-141.

Bi, Z. M., S. Y. T. Lang, et al. (2008). "Reconfigurable manufacturing systems: The state of the art." International Journal of Production Research **46**: 967-992.

Bishop, G. T. (1957). "On a problem of production scheduling." Operations Research **5**: 97-108.

Booch, G., J. Rumbaugh, et al. (2004). The Unified Modeling Language User Guide, Addison Wesley Professional.

Bowman, E. H. (1956). Production scheduling by the transportation method of linear programming. Operations Research. **4**.

Box, D. (1998). Essential COM, Addison-Wesley Professional.

Brandimarte, P., M. Rigodanza, et al. (2000). "Conceptual modeling of an object-oriented scheduling architecture based on the shifting bottleneck procedure." IIE Transactions **32**(10): 921-929.

Brookshea, G. (1989). Theory of Computation: Formal Languages, Automata, and Complexity, Addison Wesley

Bruccoleri, M. M., M. Amico, et al. (2003). "Distributed intelligent control of exceptions in reconfigurable manufacturing systems." International Journal of Production Research **41**(7): 393-412.

Buffa, E. S. and W. H. Taubert (1972). Production-Inventory Systems: Planning and Control. Homewood, IL, Irwin.

Chang, A. M., P. K. Kannan, et al. (1990). Design of an object-oriented system for manufacturing planning and control. Troy, NY, USA, IEEE Comput. Soc. Press: 2-8.

Chang, R. H. and C. M. Jones (1970). "Production and workforce scheduling extensions." AIIE Transactions **2**: 326-333.

Chappel, D. (2006). Understanding .NET, Addison-Wesley Professional.

Charnes, A., W. W. Cooper, et al. (1955). "A model for optimizing production by reference to cost surrogates." Econometrica **23**: 307-323.

Chopra, S. and P. Meindl (2007). Supply Chain Management, Pearson, Prentice Hall

Damon, W. W. and R. Schramm (1972). "Simultaneous decision model for production, marketing and finance." Managemet Science **19**: 161-172.

Deb, K., S. Agrawal, et al. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, Berlin, Germany, Springer-Verlag.

Deb, K., A. Pratap, et al. (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE Transactions on Evolutionary Computation **6**(2): 182-197.

Devedzic, V. and D. Radovic (1999). "A framework for building intelligent manufacturing systems." IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) **29**(3): 422-439.

Ebert, R. J. (1976). "Aggregate planning with learning curve productivity." Management Science **23**: 171 - 182.

ElMaraghy, H. A. (2005). "Flexible and reconfigurable manufacturing systems paradigms." International Journal of Flexible Manufacturing Systems **17**(4): 261-276.

ElMaraghy, H. A. (2009). Changeable and reconfigurable manufacturing systems, Springer-Verlag.

ElMaraghy, H. A. (2009). Changing and evolving products and systems – models and enablers. Changeable and reconfigurable manufacturing systems. H. A. ElMaraghy, Springer-Verlag**:** 25–45.

Fetter, R. B. (1962). "A Linear Programming model for long range capacity planning." Management Science **8**: 327-334.

Fonseca, C. M. and P. J. Fleming (1993). Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, San Mateo, CA, USA, Morgan Kaufmann.

Freibeit, T. I. (2004). Reliability and Productivity of Reconfigurable Manufacturing Systems.

Galbraith, J. R. (1969). "Solving production smoothing problems." Management Science **15**: 665-674.

Ganesh, K. and M. Punniyamoorthy (2005). "Optimization of continuous-time production planning using hybrid genetic algorithms-simulated annealing." International Journal of Advanced Manufacturing Technology **26**(1-2): 148-154.

Gaver, D. P. (1961). "Operating characteristics of a simple production, inventory-control model." Operations Research **9**: 635-649.

Goodman, D. A. (1973). "A new approach to scheduling aggregate production and work force." AIIE Transactions **5**: 135-141.

Goodman, D. A. (1974). "A goal programming approach to aggregate planning of production and work force." Management Science 20: 1569-1575.

Groff, G. K. and J. F. Muth (1972). Operations Management: Analysis for Decisions, Irwin, Homewood, IL.

Groover, P. M. (2000). Automation, Production Systems, and Computer-Integrated Manufacturing, Prentice Hall.

Grubbstrom, R. W. and J. Olhager (1997). "Productivity and flexibility: Fundamental relations between two major properties and performance measures of the production system." International Journal of Production Economics **52**(1-2): 73-82.

Hax, A. C. and D. Candea (1984). Production and Inventory Management, Prentice-Hall, Englewood Cliffs. NJ.

Heineman, G. T. and W. T. Councill (2001). Component-Based Software Engineering: Putting the Pieces, Addison Wesley.

Hiebeler, R., T. B. Kelly, et al. (1998). "Best Practices: Building Your Customer-Focused Solutions."

Holt, C. C., F. Modigliani, et al. (1956). "Derivation of a linear decision rule for production and employment." Management Science **2**: 159-177.

Holt, C. C., F. Modigliani, et al. (1955). A Linear Decision Rule for Production and Employment Scheduling. Management Science. **2** 1-30.

Hooke, R. and T. A. Jeeves (1961). "Direct search solution of numerical and statistical problems." Journal of the Association for Computing Machinery **8**: 212-229.

Hutchins, D. C. (1999). Just in time, Gower.

Hwang, H. and C. N. Cha (1995). "An improved version of the production switching heuristic for the aggregate production planning problem." International Journal of Production Research **33**: 2567-2577.

Ismail, M. A. and H. ElMaraghy (2009). "Progressive modeling-An enabler of dynamic changes in production planning." CIRP Annals - Manufacturing Technology **58**(1): 407-412.

Ismail, M. A. and H. A. ElMaraghy (2009). Component Oriented Design of Change-Ready MPC Systems. Changeable and Reconfigurable Manufacturing Systems. H. A. ElMaraghy, Springer-Verlag 213-226.

Johnson, R. (2002). Expert One-on-One J2EE Design and Development, Wrox.

Kao, E. P. C. (1979). "A multi-product dynamic lot-size model with individual and joint set-up costs." Operations Research **27**: 279-289.

Khoshnevis, B. and P. M. Wolfe (1983). "An aggregate production planning model incorporating dynamic productivity I: Model development." IIE Transactions **15**(2): 111-118.

Khoshnevis, B. and P. M. Wolfe (1983). "An aggregate production planning model incorporating dynamic productivity II: Solution methodology and analysis." IIE Transactions **15**(4): 283-291.

Klein, M. (1961). "On production smoothing." Management Science **7**: 286-293.

Koren, Y. (2003). "Reconfigurable manufacturing systems." Journal of the Society of Instrument and Control Engineers **42**(7): 572-582.

Koren, Y., U. Heisel, et al. (1999). "Reconfigurable manufacturing systems." CIRP Annals - Manufacturing Technology **48**(2): 527-540.

Kuzgunkaya, O. and H. A. ElMaraghy (2007). "Economic and strategic perspectives on investing in RMS and FMS." International Journal of Flexible Manufacturing Systems **19**: 217-246.

Landers, R. G., B. K. Min, et al. (2001). "Reconfigurable machine tools." CIRP Annals - Manufacturing Technology **50**: 269-274.

Laurent, G. (1976). "A note on range branch programming: Introducing a satisficing range in an L.P." Management Science **22**: 713-716.

Lee, S. M. and L. J. Moore (1974). "A practical approach to production scheduling." Journal of Production and Inventory Management **15**: 79-92.

Lee, W. B. and B. M. Khumawala (1974). "Simulation Testing of Aggregate Production Planning Models in Implementation Methodology." Management Science **20**: 903-911.

Leung, S. C. H. and S. S. W. Chan (2009). "A goal programming model for aggregate production planning with resource utilization constraint." Computers and Industrial Engineering **56**: 1053-1064.

Leung, S. C. H., W. Yue, et al. (2003). "Multi-site aggregate production planning with multiple objectives: a goal programming approach." Production Planning and Control **14**: 425-436.

Liao, Q., K. Tham Chen, et al. (2001). A Web-based material requirements planning integrated application, Los Alamitos, CA, USA, IEEE Comput. Soc.

Liu, J., W. Wang, et al. (2006). Optimal part family and production module planning for reconfigurable manufacturing systems. International Conference on Manufacturing Science and Engineering, MSEC 2006, October 8, 2006 - October 11, 2006, Ypsilanti, MI, United states, American Society of Mechanical Engineers.

Lockett, A. G. and A. P. Muhlemann (1978). "A problem of aggregate scileduling an application of goal programming." Intemational Journal of Production Research **16**: 127-135.

Manne, A. S. (1957). "A note on the Modigliani-Hohn production smoothing model." Managemem Science **3**: 371-379.

Manne, A. S. (1958). "Programming of economic lot sizes." Management Science **4**: 115-135.

Masud, A. S. M. and C. L. Hwang (1980). "Aggregate Production Planning Model and Application of Multiple Objective Decsion Methods." International Journal of Production Research **18**(6): 741-752.

Mazzola, J. B., A. W. Neebe, et al. (1998). "Multiproduct production planning in the presence of work-force learning." European Journal of Operational Research **106**(2-3): 336-356.

Mehrabi, M., G. Ulsoy, et al. (2000). "Reconfigurable manufacturing systems: Key to future manufacturing." Journal of Intelligent Manufacturing **11**: 403-419.

Meij, J. T. (1980). "Separable programming as a solution methodology for aggregate production planning." International Journal of Production Research **18**: 233-243.

Mellichamp, J. M. and R. M. Love (1978). "Production switching heuristics for the aggregate planning problem." Management Science **24**: 1242-1251.

Metaxiotis, K., D. Askounis, et al. (2001). "An object-oriented analysis and design of a model for production planning and control in industry." International Journal of Advanced Manufacturing Technology **18**(9): 657-664.

Monfared, M. A. S. and J. B. Yang (2007). "Design of integrated manufacturing planning, scheduling and control systems: A new framework for automation." International Journal of Advanced Manufacturing Technology **33**(5-6): 545-559.

Nam, S. J. and R. Logendran (1992). "Aggregate production planning – a survey of models and methodologies." European Journal of Operational Research **61**: 255-272.

Newson, E. P. (1975). "Multi-item lot size scheduling by heuristic. Part 1: With fixed resources." Management Science **21**: 1186-1193.

Newson, E. P. (1975). "Multi-item lot size scheduling by heuristic. Part 2: With variable resources." Management Science **21**: 1194-1203.

Ohiro, T., Myreshka, Morikawa, K. and Takahashi, K. (2003). A Stochastic Model for Deciding an Optimal Production Order and its Corresponding Configuration in a Reconfigurable

Manufacturing System with Multiple Product Groups. The CIRP 2nd International Conference on Reconfigurable Manufacturing, Michigan.

Olhager, J. (2003). "Guest editorial: Linking manufacturing strategy and production planning and control." Production Planning and Control **14**(6): 485-486.

Olhager, J. and J. Wikner (2000). "Production planning and control tools." Production Planning and Control **11**(3): 210-222.

Oliff, M. D. and G. K. Leong (1987). "A discrete production switching rule for aggregate planning." Decision Sciences **18**: 582-597.

Pels, H. J. (2006). "Classification hierarchies for product data modelling." Production Planning and Control **17**(4): 367-377.

Phruksaphanrat, B., A. Ohsato, et al. (2006). A comment on the formulation of an aggregate production planning problem, Bangkok, Thailand, Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States.

Pritchard, J. (1999). COM and CORBA Side by Side: Architectures, Strategies, and Implementations, Addison-Wesley Professional.

Rakes, T. R., L. S. Franz, et al. (1984). "Aggregate production planning using chance-constrained goal programming." International Journal of Production Research **22**: 673-684.

Rondeau, P. J. and L. A. Litteral (2001). "Evolution of manufacturing planning and control systems: from reorder point to enterprise resource planning." Production and Inventory Management Journal **42**(2): 1-7.

Schild, A. (1959). "On inventory, production and employment scheduling." Management Science **5**: 157-167.

Setchi, R. M. and N. Lagos (2004). Reconfigurability and reconfigurable manufacturing systems - State-of-the-art review, Berlin, Germany, Institute of Electrical and Electronics Engineers Computer Society.

Shan, F., L. X. Li, et al. (2001). "An object-oriented intelligent design tool to aid the design of manufacturing systems." Knowledge-Based Systems **14**(5-6): 225-232.

Shobrys, D. E. and O. C. White (2000). Planning, scheduling and control systems: Why can they not work together, Keystone, CO, USA, Elsevier Ltd.

Silver, E. A. (1967). "A tutorial on production smoothing and work force balancing." Operations Research **15**: 985-1010.

Silver, E. A. (1972). "Medium range aggregate production planning: State of the art." Journal of Production and Inventory Management **13**: 15-38.

Silver, E. A. and R. Peterson (1985). Decision Systems for Inventory Management and Production Planning, Wiley, New York.

Singhal, K. A. V. (1989). "Cost and shortage trade-offs in aggregate production planning." Decision Science **20**: 158-164.

Slama, D., J. Garbis, et al. (1999). Enterprise CORBA, Prentice Hall.

Sock Hwa, C. and C. A. Snyder (2000). "ERP adoption: a technological evolution approach." International Journal of Agile Management Systems **2**(1): 24-32.

Sommerville (2004). Software Engineering, Addison Wesley.

Son, S. Y. (2000). Design Principles and Methodologies for Reconfigurable Machining Systems. Engineering Research Center. Ann Arbor, MI, Ph.D. Dissertation, The University of Michigan.

Spicer, J. P. (2002). A Design Methodology for Scalable Machining Systems. Engineering Research Center. Ann Arbor, MI, Ph.D. Dissertation, University of Michigan.

Spicer, J. P. and H. J. Carlo (2006). Simultaneous scalable-reconfigurable manufacturing system design and inventory control policy decision making. International Conference on Manufacturing Science and Engineering, MSEC 2006, October 8, 2006 - October 11, 2006, Ypsilanti, MI, United states, American Society of Mechanical Engineers.

Spicer, P. and H. J. Carlo (2007). "Integrating reconfiguration cost into the design of multi-period scalable reconfigurable manufacturing systems." Journal of Manufacturing Science and Engineering, Transactions of the ASME **129**: 202-210.

Spicer, P., Y. Koren, et al. (2002). "Design principles for machining system configurations." CIRP Annals - Manufacturing Technology **51**: 275-280.

Srinivas, N. and K. Deb (1993). "Multiobjective optimization using nondominated sorting in genetic algorithms." Evolutionary Computation **2**(3): 221-248.

Stockton, D. J. and L. Quinn (1995). "Aggregate production planning using genetic algorithms." 201-209.

Taubert, W. H. (1968). "A search decision rule for the aggregate scheduling problem." Management Science **14**: 343-359.

Techawiboonwong, A. and P. Yenradee (2003). "Aggregate production planning with workforce transferring plan for multiple product types." Production Planning and Control **14**(5): 447-458.

Tsai, T. and R. Sato (2004). "A UML model of agile production planning and control system." Computers in Industry **53**(2): 133-152.

Tsu Ta, T. and T. O. Boucher (2002). "An architecture for scheduling and control in flexible manufacturing systems using distributed objects." IEEE Transactions on Robotics and Automation **18**(4): 452-462.

Vergin, R. C. (1966). "Production scheduling under seasonal demand." Journal of industrial Engineering **17**: 260-266.

Vollman, Berry, et al. (2005). Manufacturing Planning and Control for Supply Chaing Management, McGraw-Hill.

Wache, I. (1998). "Object-oriented modelling of material flow controls." Journal of Materials Processing Technology **76**(1-3): 227-232.

Wacker, J. G. and M. Hanson (1997). "Some practical advice to manufacturing managers: Empirical results from the Global Manufacturing Research Group." Production and Inventory Management Journal **38**(3): 64-71.

Wang, A. J. A., K. Qian, et al. (2005). Component-Oriented Programming.

Wang, R. C. and H. H. Fang (2001). "Aggregate production planning with multiple objectives in a fuzzy environment." European Journal of Operational Research **133**(3): 521-536.

Weindahl, H. P., H. A. ElMaraghy, et al. (2007). "Changeable Manufacturing - Classification, Design and Operation." CIRP Annals - Manufacturing Technology **56**(2): 783-809.

Welam, U. P. (1976). "Comments on goal programming for aggregate planning." Management Science **22**: 708-712.

Wu, B. (2000). Manufacturing and Supply Systems Management: A Unified Framework of Systems Design and Operation, Springer.

Wu, B. (2001). "Strategy analysis and system design within an overall framework of manufacturing system management." International Journal of Computer Integrated Manufacturing **14**(3): 319-341.

Xiaobo, Z., W. Jiancai, et al. (2000). "A stochastic model of a reconfigurable manufacturing system. Part 1: A framework." International Journal of Production Research **38**(Compendex): 2273-2285.

Xiaobo, Z., J. Wang, et al. (2000). "A stochastic model of a reconfigurable manufacturing system. Part 2: Optimal configurations." International Journal of Production Research **38**(Compendex): 2829-2842.

Youssef, A. M. A. and H. A. ElMaraghy (2006). "Assessment of manufacturing systems reconfiguration smoothness." International Journal of Advanced Manufacturing Technology **30**(Compendex): 174-193.

Youssef, A. M. A. and H. A. ElMaraghy (2007). "Optimal configuration selection for reconfigurable manufacturing systems." International Journal of Flexible Manufacturing Systems **19**(Copyright 2008, The Institution of Engineering and Technology): 67-106.

Zhang, D. and H. C. Zhang (1999). "Simulation study of an object-oriented integration testbed for process planning and production scheduling." International Journal of Flexible Manufacturing Systems **11**(1): 19-35.

Zhao, X., W. Jiancai, et al. (2001). "A stochastic model of a reconfigurable manufacturing system. 4. Performance measure." International Journal of Production Research **39**(Copyright 2001, IEE): 1113-1126.

Zhao, X., J. Wang, et al. (2001). "A stochastic model of a reconfigurable manufacturing system. Part 3: Optimal selection policy." International Journal of Production Research **39**(Compendex): 747-758.

Zitzler and Thiele (1999). "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach." IEEE Transactions on Evolutionary Computation.

Zitzler, E., M. Laummans, et al. (2001). SPEA2, Improving the Strength Pareto Evolutionary Algorithm, http://www.tik.ee.ethz.ch/aroma.

APPENDIX A: SPEA 2 ALGORITHM

SPEA2 was proposed by Zitzler and Thiele (2001) as an improvement of SPEA. The overall algorithm can be presented in the following steps:

Input: N (population size)

$\overline{\mathrm{N}}$ (archive size)

T (maximum number of generations)

Output: A (non-dominated set)

Step 1: Initialization: Generate an initial population P_0 and create the empty archive (external set) $\overline{\mathrm{P}_0} = \phi$, Set t = 0.

Step 2: Fitness assignment: Calculate fitness values of individuals in P_t and $\overline{P}_t$ .Each individual i in the archive $\overline{P}_t$ and the population P_t is assigned a strength value $S(i)$, representing the number of solutions it dominates

$$S(i) = \left|\{ j \mid j \in P_t + \overline{P}_t \wedge i \succ j\}\right|$$

Where $|.|$ denotes the cardinality of the set, + stands for multiset union and the symbol $\wedge$ corresponds to the Pareto dominance relation. On the basis of *S* value, the raw fitness $R(i)$ of an individual *i* is calculated

$$R(i) = \sum_{j \in P_t + \overline{P}_t \wedge i \succ j} S(j)$$

That is the raw fitness is determined by the strengths of its denominators in both archive and population, as opposed to SPEA where only archive members. In addition, density information is incorporated to discriminate between individuals having identical raw fitness values. The density estimation technique in SPEA2 is an adaptation of the kth

nearest neighbour method, where the density at any point is a decreasing function of distance to the kth nearest data point. Here the inverse of distance to the kth neighbour is used as the density estimate. For each individual *i* the distances (in objective space) to all individuals *j* in archive and population are calculated and stored in a list. After sorting the list in increasing order, the kth element gives the distance sought, denoted as $\sigma_i^k, k = \sqrt{N + \overline{N}}$ is used as a common setting. Now the density $D(i)$ could be evaluated as follows

$$D(i) = \frac{1}{\sigma_i^k + 2}$$

In the denominator, two is added to ensure that its value is greater than zero and that $D(i) < 1$. Finally, adding $D(i)$ to the raw fitness value if an individual i gives its fitness

$$F(i) = R(i) + D(i)$$

Step 3: Environmental selection: Copy all non-dominated individuals in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. Now there are three possible scenarios:

1. If $\left|\overline{P}_{t+1} = \overline{\mathrm{N}}\right|$, the environmental selection step is complete.

2. If $\left|\overline{P}_{t+1} > \overline{\mathrm{N}}\right|$, then reduce $\overline{P}_{t+1}$ by means of the *truncation operator*. This operator iteratively removes individuals from $\overline{P}_{t+1}$ until $\left|\overline{P}_{t+1} = \overline{\mathrm{N}}\right|$. At each iteration, the individual which has the minimum distance to another individual is chosen to be removed; ties are broken by considering the second smallest distance and so on.

3. If $\left|\overline{P}_{t+1} < \overline{\mathrm{N}}\right|$, then fill $\overline{P}_{t+1}$ with dominated individuals in P_t and $\overline{P}_t$. This can be implemented by sorting the multiset $P_t + \overline{P}_t$ according to the fitness values and copy the first $\overline{\mathrm{N}} - \left|\overline{P}_{t+1}\right|$ individuals i with *F(i)* $\geq$ 1 from the resulting ordered list to $\overline{P}_{t+1}$.

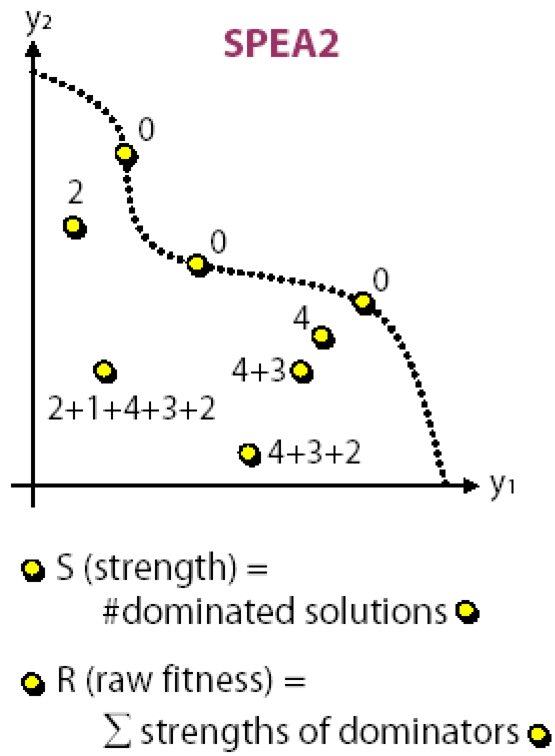


Figure A-1: SPEA2 Raw Fitness Evaluations (Zitzler, Laummans et al. 2002)

Step 4: Termination: If t $\geq$ T or another stopping criterion is satisfied then set A to the set of decision vectors represented by the non-dominated individuals in $\overline{P}_{t+1}$ and stop.

Step 5: Mating selection: Perform binary tournament selection with replacement on $\overline{P}_{t+1}$ in order to fill the mating pool.

Step 6: Recombination: Apply cross over and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter (t = t + 1) and go to Step 2.

APPENDIX B: COMPONENT MODELS AND COMPONENT DIAGRAMS

Problem component model is a seminal part of the first stage of Progressive Modeling. All component models presented in this research are presented using the highest levels of abstraction without describing the details in order to generalize the concepts developed. This appendix describes some formal definition of UML components and notation used.

UML (Booch, Rumbaugh et al. 2004) defines an interface as a collection of operations that specify a service that is provided by or requested from a class or component. A component is a replaceable part of a system that conforms to and provides the realization of a set of interfaces. The relationship between component and interface is important. All the most common component-based operating system facilities (such as COM+, CORBA, and Enterprise Java Beans) use interfaces as the glue that binds components together. An interface that a component realizes is called a provided interface, meaning an interface that the component provides as a service to other components. A component may declare many provided interfaces. The interface that a component uses is called a required interface, meaning an interface that the component conforms to when requesting services from other components. A component may conform to many required interfaces. Also, a component may both provide and require interfaces. As Figure B-1 indicates, a component is shown as a rectangle with a small two-pronged icon in its upper right corner. The name of the component appears in the rectangle. An interested reader is advised to review the UML user manual written by the three amigos who founded and developed UML, Booch, Rumbaugh, and Jacobson (2004).

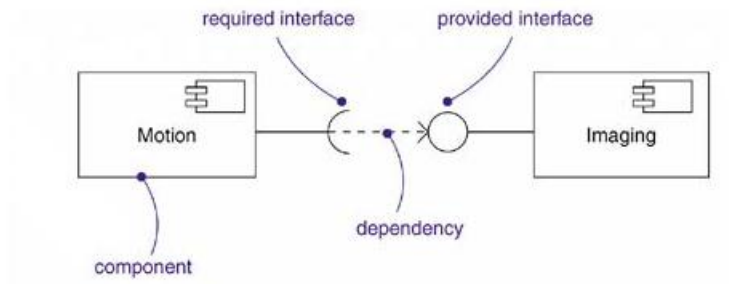


Figure B-1: Component Diagram Basic elements UML notation (Booch, Rumbaugh et al. 2004)

VITA AUCTORIS

Born for an Egyptian entrepreneur, Mohamed Abdel-Wahab Ismail had the chance to have hands on experience in the business world since he was a 12-years-old child. In 1993, he joined the Faculty of Engineering, Cairo University. He got his bachelor of science from the Mechanical Design and Production Department. In 1998, he joined the faculty of Computers and Information, Cairo University, as a graduate assistant; in 2004, he was promoted as assistant lecturer later. He earned a Master of Science in Industrial Engineering from Cairo University in 2004 and a Master of Business Administration (MBA) in Finance and International Business in 2005. In 2006, he joined the University of Windsor as a PhD student at the Industrial and Manufacturing Systems Engineering (IMSE) Department and as a Research Assistant as well in the Intelligent Manufacturing Center (IMS). From 1998 to 2005, he held many industrial positions as a Research and Development Engineer, Operations and Purchasing Engineer, and Operations Manager. Mohamed has also keen interest in software development and algorithms. He was also a true fan of mathematics since he was primary school student. Capitalizing on the aforementioned well-assorted industrial and academic experience, he founded and developed both Change ready MPC Systems and Progressive Modeling as a part of his PhD research.