AN AUTOMATED MACHINE SELECTION SYSTEM WITH FUZZY LOGIC FOR THE PLASTIC INJECTION INDUSTRY

SIRINUT SUWANNASRI

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR
THE DEGREE OF MASTER OF ENGINEERING
(INDUSTRIAL ENGINEERING)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2014**

Thesis
entitled

AN AUTOMATED MACHINE SELECTION SYSTEM WITH FUZZY LOGIC FOR THE PLASTIC INJECTION INDUSTRY

……………….…………..…..………
Ms. Sirinut Suwannasri
Candidate

……………………….……..………
Lect. Ronnachai Sirovetnukul,
Ph.D. (Industrial Engineering)
Major advisor

……………………….……..………
Lect. Kiattisak Sritrakulchai,
Ph.D. (Mechanical Engineering)
Co-advisor

……………………….……..………
Lect. Mongkol Thianwiboon,
Ph.D. (Mechanical Engineering)
Co-advisor

……………………….……..……
Assoc.Prof. Sombat Thanawan,
Ph.D., Acting Dean
Faculty of Graduate Studies
Mahidol University

……………………….……..………
Asst.Prof. Thanakorn Naenna,
Ph.D. (Engineering Science)
Program Director
Master of Engineering Program in
Industrial Engineering
Faculty of Engineering
Mahidol University

Thesis
entitled

AN AUTOMATED MACHINE SELECTION SYSTEM WITH FUZZY LOGIC FOR THE PLASTIC INJECTION INDUSTRY

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Engineering (Industrial Engineering)
on
March 24, 2014

……………….…………..….………
Ms. Sirinut Suwannasri
Candidate

……………….…………..….………
Lect. Jiraprabha Kimsunthorn,
Ph.D. (Mechanical Engineering)
Chair

……………………………..…..………
Lect. Ronnachai Sirovetnukul,
Ph.D. (Industrial Engineering)
Member

……………………………..…..………
Lect. Mongkol Thianwiboon,
Ph.D. (Mechanical Engineering)
Member

……………………………..…..………
Lect. Kiattisak Sritrakulchai,
Ph.D. (Mechanical Engineering)
Member

……………………………..…..………
Assoc.Prof. Sombat Thanawan,
Ph.D., Acting Dean
Faculty of Graduate Studies
Mahidol University

……………………………..…..………
Lect. Worawit Isarangkul,
M.S. (Technical Management)
Dean
Faculty of Engineering
Mahidol University

ACKNOWLEDGEMENTS

This research project would never been succeeded without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

First of all, I would like to express my deepest gratitude to my major advisor, Dr. Ronnachai Sirovetnukul, for the patient guidance and encouragement. And he also patiently corrected my writing and advice throughout my several past years as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. And I would like to express my appreciation and deeply thank to Dr. Jiraprabha Kimsunthorn, Dr. Kiattisak Sritrakulchai and Dr. Mongkol Thianwiboon for their kindness in valuable suggestion and time sacrifice through this study.

I would also carry my thankfulness to my first advisor, Dr. Thitikorn Limchimchol, who let me experience the research of injection molding machine selection process for his excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research during the first years. I would like to thank Mr. Manus Sudsri, who helped me collect data, granted me his knowledge and excellent skills, gave me all his documents I needed and his precious time during the days in field as well as everyone at Hi-Q Plas Co., Ltd. I would like to assert that my research would not have been possible without their helps.

Particularly, I am enormously indebted to my father and mother who have been continuously giving me great support for my study and this research. Finally, I thank my elder sister, elder brother, friends, and all involved persons at Mahidol University who have been always supportive and financial reinforcement for my first International Conference attendance in Barcelona. I would never have those wonderful days without them.

Sirinut Suwannasri

AN AUTOMATED MACHINE SELECTION SYSTEM WITH FUZZY LOGIC FOR THE PLASTIC INJECTION INDUSTRY

SIRINUT SUWANNASRI 5237550 EGIE/M

M.Eng. (INDUSTRIAL ENGINEERING)

THESIS ADVISORY COMMITTEE: RONNACHAI SIROVETNUKUL, Ph.D., KIATTISAK SRITRAKULCHAI, Ph.D., MONGKOL THIANWIBOON, Ph.D.

ABSTRACT

This research aims to develop an injection molding machine selection system as a tool and to enhance machine-job assignment in a plastic industry. This study apply the idea of fuzzy logic, which is able to deal with quantitative and qualitative criteria, to improve current planning process with precision, repeatability, and time saving. The six machine selection criteria are obtained by theoretical review and an industry case study to construct the system on MATLAB platform. The proposed system is composed of two sub-systems that are Fuzzy Inference System (FIS) and System II. The FIS contains the fuzzy sets of four inputs to generate 384 fuzzy rules for selecting the machine group. System II returns the most appropriate machine number. To determine the conformity of the developed system to the requirement of activity and the satisfaction of the developed system to user requirement, the application of verification and validation are performed by testing. Finally, these processes prove that the proposed system executes with higher accuracy and efficient computational time when compared to the current system. The computational time is reduced by the proposed system from 63.45 to 0.01 second per job.

KEY WORDS: MACHINE SELECTION / FUZZY LOGIC / INJECTION MOLDING MACHINE

261 pages

ระบบการเลือกเครื่องจักรอัตโนมัติด้วยตรรกศาสตร์คลุมเครือสำหรับอุตสาหกรรมฉีดพลาสติก
AN AUTOMATED MACHINE SELECTION SYSTEM WITH FUZZY LOGIC FOR THE PLASTIC INJECTION INDUSTRY

สิรินุช สุวรรณศรี 5237550 EGIE/M

วศ.ม. (วิศวกรรมอุตสาหการ)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : รณชัย ศิโรเวฐนุกูล, Ph.D., เกียรติศักดิ์ ศรีตระกูลชัย, Ph.D., มงคล เทียนวิบูลย์, Ph.D.

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบเลือกเครื่องฉีดพลาสติกโดยอัตโนมัติ และใช้เป็นเครื่องมือในการปรับปรุงการมอบหมายงานให้กับเครื่องจักรของโรงงานฉีดพลาสติก การศึกษาได้ประยุกต์ใช้ตรรกศาสตร์คลุมเครือที่สามารถจัดการกับเกณฑ์การเลือกทั้งในเชิงปริมาณและคุณภาพ เพื่อปรับปรุงกระบวนการวางแผนการผลิตในปัจจุบันให้ดีขึ้น โดยที่มีทั้งความแม่นยำ สามารถทำซ้ำได้ และประหยัดเวลา เกณฑ์การเลือกเครื่องจักรทั้ง 6 ชนิด ถูกรวบรวมจากทฤษฎีและโรงงานกรณีศึกษาและพัฒนาบนโปรแกรม MATLAB ระบบที่เสนอประกอบด้วย 2 ระบบย่อย นั่นคือ Fuzzy inference system และ System II โดย Fuzzy inference system ประกอบด้วย Fuzzy set ของข้อมูลนำเข้า 4 ชนิด ก่อทำให้เกิดเงื่อนไขทั้งหมด 384 เงื่อนไข ซึ่งมีหน้าที่เลือกกลุ่มของเครื่องจักรที่เหมาะสม ส่วน System II จะทำการเลือกเครื่องจักรที่เหมาะสมที่สุดหนึ่งเครื่องเพื่อทำการผลิต งานวิจัยยังได้ประยุกต์กระบวนการตรวจสอบความถูกต้อง (Verification) และการยืนยันความถูกต้อง (Validation) มาใช้ เพื่อยืนยันว่าระบบสามารถทำงานได้ถูกต้องตรงตามที่ออกแบบและความต้องการของผู้ใช้งาน ท้ายที่สุดระบบที่เสนอใหม่สามารถเลือกเครื่องจักรได้ถูกต้องและมีประสิทธิภาพมากกว่าระบบเดิมในด้านเวลาที่ใช้ในการเลือกเครื่องจักร ซึ่งลดลงจาก 63.45 วินาทีเป็น 0.01 วินาทีต่องาน

261 หน้า

CONTENTS

CONTENTS (Cont.)

CONTENTS (Cont.)

CONTENTS (Cont.)

CONTENTS (Cont.)

LIST OF TABLES

LIST OF TABLES (Cont.)

LIST OF TABLES (Cont.)

LIST OF FIGURES

LIST OF FIGURES (Cont.)

LIST OF FIGURES (Cont.)

LIST OF FIGURES (Cont.)

LIST OF FIGURES (Cont.)

CHAPTER I
INTRODUCTION

1.1 Background and problem statement

Machine selection processing is part of production planning. This process aims to assign a job to a specific machine. Over the last decade, a machine-job assignment problem has been integrated in harmony with job-shop scheduling theory. But it does not feasible in practice due to the lack of sequencing procedure for evaluation. Although, the previous studies had investigated it based on main criteria (e.g. cost, processing time, knowledge, experience from a planner, etc.) but other variables are particularly required. Other variables such as the number of machines, varity of machine specifications and other environmental factors that are complex decisions should be considered with the machine performance trading off.

In addition, the effective machine - job assignment is an important issue in the injection molding industry. All injection molding machines will be assigned to a job based on their capacity or physical properties (e.g. clamping force, the dimension of tie bars and previous job histories. Not only an injection molding machine is able to influence on the quality of products directly, but the machine performances and life - times is also. There are a number of common defects in the injection molding caused by the inappropriate machine such as black streaking, burn marks, poor part finish, poor surface detail, color fading, short shot, etc. Besides, the malfunction of electrical and hydraulic system can be indirectly affected by machine overloading. Then, the processing time will be extended if a production line is stopped to apply some modification during mass production. An experienced planner or experts commonly function to select the machine. These situation risks an irregular machine using as opposed to an automatic computational scheduler when they have resigned or inexperienced officer did it. Having the decision support system is able to enhance the current process and also decrease the human error with more precision, better accuracy and faster results. However, the implementation of the machine selection system mechanism should considers all the possible methods which are familiar with human decision making, lower cost and shorter time.

A new theory of Machine Intelligence has been called Artificial Intelligence (AI) that is important role to handle the uncertainty of virtualized organization. The machine intelligence mechanism should involve the uncertainty management and provide reason back in linguistic pattern automatically. Ma, Chen and Xu (2006) stated that the intelligence machine was developed to enhance the machine's ability to learn the knowledge from environmental changing dramatically, seeking suitable knowledge, making decision ability under knowledge consideration and apply them to make better conclusions. A branch of AI that has been become popular in dealing with the production uncertainty is a fuzzy logic theory. It is easy to model any ambiguous concept with descriptions and information by using linguistic terms (Ma, Chen and Xu, 2006). Thus, it is a good alternative way unless other classical logic. Fuzzy logic has great advantages of description and uncertainty for machine intelligence application, is suitable for the problem that could not be modeled in mathematical form and handles the incomparable linguistic inputs. Due to the outstanding properties of fuzzy logic theory, it gains the possibility to invent an efficient machine selection system concerning uncertain information. The proposed system will be used to solve the machine-job assignment problem with similar to human thinking in a short time.

Therefore, this research aims to develop an automated machine selection system with fuzzy logic for the plastic injection industry using MATLAB with product consideration. After that, a comparison between the current system and the proposed system will be illustrated in the following section with the simulation method.

1.2 Objectives

The objectives of this research are as follows:

1) To develop the injection molding machine selection system on MATLAB using fuzzy logic in the plastic industry.

2) To improve the efficiency of the machine selection method that influences on computational time and defect reduction.

1.3 Scope of Work

1) The scope of this study is limited to the case study of a medium-size plastic manufacturer in Pathumthani.

2) The data are only observed from machine selection process made by a factory planner.

3) The user of the proposed system is the planner of the plastic manufacturer.

1.4 Methodology

- Observation and identification of problem
- Literature Review
- Data collection
- Research planning
- Design framework's system
- Encoding with MATLAB
- System testing
- Result reporting
- Conclusion and Discussion

1.5 Expected Results

- The proposed system can reduce the evaluation time of the machine selection process to help a factory planner in the preparation of the production plan.
- The proposed system can reduce the defective product when compared to the current system.

- Without loss of generality, the methodology in this thesis can also be applied to other plastic manufacturers.

CHAPTER II
LITERATURE REVIEW

This chapter involves foundation of background and literature review consisting of four important parts. First, it addresses machine selection definition and role of importance. And relevant theories will be discussed. Second, it focuses on the characteristic of injection molding machine selection process in industry. In this regard, such previous researches may share some perspectives with this study and solution methods that are beneficial to this research. Third, it is given in detail of fuzzy logic theory. Finally, it includes orientation of MATLAB as the development platform of this research.

2.1 Machine selection

This part concerns with the machine selection concept and role in production planning.

2.1.1. What is machine selection

Machine selection is a part of production planning which plays on the assessment of the most suitable machine to produce the products. All requirements will be identified and matched to the available machines. The selection of machine is a part of resource allocation. These resource quantities could be either discrete or continuous. Generally, the machines are arranged to be a single machine, parallel machines, or machines in a flow shop, job shop, or open shop (Shabtay and Steiner, 2007). In case of the cost is the highest priority, the machine that has lowest price will be selected (Tabucanon et al., 1994). This process has judged into a multi-criteria decision making problem where qualitative and quantitative factors are involved. And it is emphasized as a significant strategy of the production management, complicated and talent (Tabucanon et al., 1994; Rao, 2007; Guldogan, 2011).

There is the difficulty of dividing the machine selection from generic scheduling problems. Many researches had given precedence to the scheduling problem term instead of machine scheduling. An effective scheduling always synchronizes to sufficient manufacturing's factors appropriation.

Scheduling problem is mapping of jobs to the machines. The definition is finding of a schedule when a set of **jobs** $J = \{J_1,..., J_n\}$ have to be assigned to a set of **machines** $M= \{M_1, . . . ,M_m\}$. And the processing times should be feasible and based on optimization constraints (Figure 2.1). Scheduling is represented by Gantt charts or job-oriented.

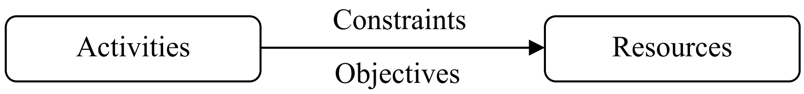


Figure 2.1 Scheduling problem process

Scheduling of machine is possibly constrained by machine capacity, shop floor layout or job arrival phenomena. The differences of machinery and manufacturing types lead to a diverse set of variables. According to the validation of a machine schedule, there is a classification of scheduling problems to shop configuration, job arrival and operation times while some researches called machine environment and job characteristics. For example, even if a rotary injection molding machine is a single machine, its behavior is similar to a parallel machine. Thus scheduling of the rotary injection machine prefers to a parallel machine scheduling problem (Urs, 2005).

A group of the published articles reviewed the influencing factors in production scheduling such as late works, setup times, costs, machine availability, equal processing times, etc. (Sterna, 2011; Allahverdi et al., 2008; Ma, Chu and Zuo, 2010; Kravchenko, 2011; Nowicki and Zdrzalka, 1990; Shabtay and Steiner, 2007). The objective of these problems is seeking of the production schedule by minimizing completion times and costs. There are not detailed in the differently machine's characteristic.

2.1.2. Role of machine selection

The manufacturing decisions consist of the selection of material, process, machining method, robot, inspection system, designing software, process layout, facility location, vendor and performance evaluation, process condition and product design. The machine selection is also involved in, not only the selection decision upon purchasing process but the job assignment is also (Rao, 2007). Machine selection consequence had raised the flavoring machines to match for any given job and then scheduled all assigned jobs. The effective selection decision requires comprehension of the current situation and unpredictable events of manufacturing conditions. The necessary logical idea always depends on skill and the experience of the planner. Thus, the computer is underdeveloped for this (Nagarur, Vrat and Duongsuwarn, 1997). In major of researches in production planning areas, they often participated in machine selection solution in the context of production or machine scheduling problem. But the conventional scheduling theory had not emphasized in the machine selection process. Since the day of computer technology still not common and widespread, the solution of planning required all proficient calculation and decision skills of human-being thoroughly. Even higher computer technology revolutions were introduced, rather many industries keep employing the expert or experienced planner to handle this with the following reasons:

- The realistic of an outstanding selection process's property that must deal with uncertainty and imprecision,
- The machine selection mechanism is difficult to form mathematical model or programming,
- Human logic has skills to learn knowledge and reacts to problem solving along with unsharpness information and the problem that cannot be modeled,
- Conventional machine scheduling theory had not concerned with the machine selection process (the process before the job assignment).

These reasons cause an irregular operational flow as opposed to an automatic computational scheduler.

There is no denying the fact that the importance of high manufactured volume is replaced by the role of the high business speed which is the rapid reaction to customers' orders (Tanev et al., 2004). According to enhance the planning process, the upstream innovation, the automation resource allocation is a major key to support that issue. The intelligence system trend is increasing. Major of researchers who paid attention in developing the expert system to be a decision-maker had been occurred. The benefits of the planning by the expert system are getting more effective plan base on similar standard in every time and no experience required as human (Chujo et al., 2005).

2.2 Relevance studies of the machine selection

Over the last decades, selection problems have been involved with many researchers. These researches can be categorized into two major groups based on their objectives which are the machine selection for purchasing and production. Machine tools and any specific machines were paid attention to study. The studies aimed to select the best machine to purchase and produce the required product. Beside of the mentioned studies, the selection of process parameter and operation were usually interested to be their topics. Although, these selection styles are not in this research area, but the concept and the process are similar to the injection molding machine selection. Thus, it has potential to apply to our study.

2.2.1 Review of applications

Tabucanon et al. (1994) was developed the decision support system (DSS) by applying an Analytical Hierarchy Process (AHP) and a rule-based technique to select the most suitable machine for a Flexible Manufacturing System (FMS). Their study used part information and machine database as the input data. The system was included by three sub-systems. The first one is the selection the best machine from four types of CNC machine. The second aimed to reduce the number of candidate machines to be lower than seven machines within the same type. The rule-based technique was used as a machine selection model. And the third is the system for selecting the best machine by using the concept of multi-criteria machine selection. One advantage of AHP is to allow users to add or deduct machines directly in the database via an application of Database Management System. However, a disadvantage is that the first and second systems rely on all answers and the third is depended on ratings of users by their high level of knowledge. The problem when this step has been done by the persons with different skills is the different standard of rating scores. The DSS named MACSEL by Tabucanon et al. was also used by Iç and Yurdakul (2009) for the purpose of machining center selection. The using of AHP was applied in Lin and Yang (1996), Arslan, Çatay & Budak (2004) and Çimren, Çatay, and Budak (2007). All those studies must required score rating based on user skills. However, Rao (2007) suggested that it was not the best machine selection approach by comparing to other approaches.

Lin and Yang (1996) developed an expert system by applying of AHP to select the machine types among FMC, NC and conventional machines. The survey of the expert's experiences has done and used as the evaluation principle. Arslan, Çatay & Budak (2004) used a decision support system to buy the new machine tool. Their software made the decision from machine database. The criteria of the software are machine working conditions, e.g. force and power and the scores of productivity, flexibility, etc. could be calculated by the model. The multi-criteria weighted average was used to calculate the weights from many manufacturers. Then, a few numbers of candidate machines would be demonstrated and analyzed the cost/benefit. Çimren, Çatay, and Budak (2007) analyzed the cost also (i.e. purchasing, operation and maintenance costs). Moreover, the result of ranking based on reliability, precision and robustness analysis were demonstrated by sensitivity analysis was used as the final process. And Tsai et al. (2010) surveyed the machine tool selection criteria from management level and consultants. Because, the standards of making the decision are different so they used AHP as the tool for applying in 4-axis machining center selection. They used the software package named 'Expert Choices' in the calculation of weight and results.

Chu & Lin (2003), Byun and Lee (2005) and Iç and Yurdakul (2009) used fuzzy TOPSIS to select a robot, a rapid prototype and the machining center respectively. The TOPSIS was also applied together with AHP by means of fuzzy extension, namely fuzzy TOPSIS and fuzzy AHP, to weight inexplicit criteria by Önüt, Kara, Efendigil (2008). Fuzzy AHP was early proposed in 1992. It was the using of prominent point of fuzzy theory to enhance users' evaluation with interval values. The interval values are more convenient than fixed values. Because, they are able to represent the importance of each criteria by fuzzy number which is better than precise number. They used fuzzy TOPSIS for assessing scores of CNC machining centers rated by purchasers.

According to the study of Lababidi and Baker (1999), fuzzy number was also used in the expert system in a pattern of fuzzy multiple attribute decision-making (FMADM) model. The system aimed to select the batch-dryer for foodstuffs. Wang, Shaw, Chen (2010) used the same kind of model to purchase new machine for supporting of Flexible manufacturing cells (FMC). This model is realistic and economically.

From these studies, fuzzy theory was emphasized to be a highly effective approach to simulate human decision logically. Then, a number of articles were adapted this principle as the core tool for solving machine selection problem. One of them is the study of Hanna and Lotfallah (1999). They used fuzzy logic to solve the crane selection problem in order to support the project. This problem required the transformation of parameter values in qualitative and the term that cannot be used by the classical decision process to be quantitative values. Subramaniam et al. (2000) proposed the selection process by three machine selection rule that are Lowest average cost, Least average process time and Least aggregate cost and process time. These rules were applied to machine selection by considering the minimization of processing cost and time. They can support the functionality of production scheduling by means of dispatching rules under production factors such as due dates, number of suitable resources to process a job operation, and machine breakdown. Subramaniam, Ramesh and Raheja (2002) identified the most suitable machine for scheduling with the fuzzy machine selection process. The simple machine selection rules were cooperated to enhance the performance of each of the four dispatching rules. This study used the fuzzy inference system which is included of the normalization, fuzzification, fuzzy inferencing, defuzzification and denormalization processes. The outputs of a fuzzy inference system are TC’ (Tardiness Coefficient) and CC’ (Cost Coefficient). After that, the lowest weight machine would be selected based on the calculation model of (TC’ x SPTi) + (CC’ x SPCi). SPTi (Shortest processing time of each machine) and SPCi (Shortest processing cost of each machine) are metrics which demonstrate the performance measurement of dispatching rules. After the efficiency comparison between fuzzy machine selection approach and machine selection rules was performed, the results showed that the proposed approach could improve the scheduling significantly both in terms of Mean Job Tardiness and Mean Job Cost. For this application, the fuzzy approach can be adapted in a traditional job shop environment that uses rigid rules and the processing condition are always changing. The comfortable in dealing with the machine selection problem of fuzzy approach was found in Al-Hawari et al. (2010) as well. They reinforced the matching of 2D laser scanner with many aspects by fuzzy logic. The 2D laser scanner is including of various models and types. The 11 factors and weights were collected from the experts which are composed of common selection factors, they are static and dynamic factors. The explanation of static factors is independent factors of the applications, the type of scanner though such as cost. And dynamic factors are opposite such as measurement accuracy. The linguistic value of the fuzzy logic processing will transform static factors to be as in fuzzy sets and IF-THEN rules for dynamic factors. The most suitable laser scanner would be the highest rating score (the value of Center of gravity, COG). Kulak et al (2005) showed the satisfying association of crisp criteria and fuzzy criteria in the problem with low data volume. Moreover, fuzzy logic was applied in other aspect such as fuzzy – based multi - objective goals or fuzzy goal programming by Chan and Swankar (2006). Their objectives were maintaining of total machining cost, set-up cost and material handling cost to be lower than the goal.

All of the studies has been used the selection criteria that vary by the nature of the machine. The criteria of each study are represented by the parameter values of machine which consists of machine specification, costs, productivity, procedure, etc. And these criteria show the relationship between the parameters and the objectives or KPI. In fact, multi-criteria decision making problem is able to optimize limited objectives. And we can use the same or different parameters for the same machine types depend on the objectives. The parameter is usually machine specification. But in case of the objective or KPI is the minimization of cost, for example, a user might be required to add more parameters in terms of purchasing cost, production cost or maintenance cost to his model. For the details of parameter, objective or KPI (Key Performance Index) of each study is reported in Table 2.1

Table 2.1 Previous studies in area of machine selection

Year	Authors	Techniques	Parameters	Objectives / KPI	Machine Types
1994	Tabucanon et al.	A decision support system using Analytical Hierarchy Process (AHP) and Rule-based technique approaches	Flexibility, adaptability, continuous operation, special features, total cost, total productivity, power requirement and space requirement	Maximum weight of all preference (parameters)	CNC turning center
1996	Lin and Yang	Analytical Hierarchy Process (AHP)	Machine procedure, lead time, labor cost, operation shift	Maximum weight of all preference (parameters)	Flexible manufacturing cells (FMC), NC (numerically controlled) machine and conventional machine
1999	Lababidi and Baker	Fuzzy expert system	Dryer throughput, moisture content, mean particle size, temperature sensitivity, cohesiveness and fragility	Improve accuracy, flexibility and adaptability	Batch-dryer for foodstuffs
	Hanna and Lotfallah	Fuzzy logic approach for machine type selection for construction project	Site condition, building design, economy, capability	Maximum efficiency centers	Mobile crane, Tower crane and Derrick crane

Table 2.1 Previous studies in area of machine selection (Cont.)

Year	Authors	Techniques	Parameters	Objective / KPI	Machine Types
2000	Subramaniam et al.	A new machine selection rule for two-stage scheduling approach and simulation	Cost, process time, average cost	Lowest average cost, Least average process time, Least aggregate cost and process time	Not specified
	Wang, Shaw and Chen et al.	Fuzzy multiple attribute decision-making (FMADM) model	Total purchasing cost, total machine floor space, total machine no. and productivity of constructed FMC	Maximum degree of fuzzy preference relation. / Distinguishability, robustness and triangular distribution comparison	CNC milling machine
2002	Subramaniam et al.	Fuzzy machine selection approach increased dispatching rules scheduling performance	Performance measures, cost rate of machine, processing time of an operator and due dates	Minimum weighted sum. / Mean job tardiness and mean job cost	Not specified
2003	Chu and Lin	Fuzzy TOPSIS method	Man-machine interface, programming flexibility, vendor's service contract, purchase cost, load capacity and positioning accuracy	Maximum closeness coefficient	Robot
2004	Arslan, Çatay and Budak	A decision support system using multi-criteria weighted average and AHP methods	Productivity, flexibility, space, adaptability, precision, cost, reliability, safety and environment, maintenance and service	Maximum score based on all criteria. Break-even production quantity	Machine tools

Table 2.1 Previous studies in area of machine selection (Cont.)

Year	Authors	Techniques	Parameters	Objectives / KPI	Machine Types
2005	Chtourou, Masmoudi and Maalej	Expert system simulation approach	No. of machine to machine department M, list of all machine departments (descendents), list of machine departments with a machine lack problem, list of machine departments with a machine surplus problem	Resource suggestion (no. of machine to machine department M), minimize mean batch tardiness and minimize mean batch earliness. Mean no. of batch waiting, mean waiting time of a batch, mean machine utilization rate, maximum allowable utilization, minimum allowable utilization rate, utilization state, diagnosed problem, mean tardiness, mean advance and manufacturing system state	Not specified
	Kulak et al.	Multi-attribute fuzzy AD approach (software-MAXD)	Fixed costs per hour, variable costs per hour, and equivalent costs of standard tools per hour. Length of sheet size, thickness of sheet metal, no. of strokes, simultaneous axis speed, tool rotation speed and sufficiency of service.	Most suitable alternative with respect to the designer's requirements. Minimum total unit index.	Punching machine

Table 2.1 Previous studies in area of machine selection (Cont.)

Year	Authors	Techniques	Parameters	Objectives / KPI	Machine Types
2005	Byun and Lee	A decision support system using a MADM method called TOPSIS	Dimensional accuracy, surface roughness, part cost, build time and material properties	Maximum closeness coefficient	Rapid prototype
2006	Chan & Swarnkar	Fuzzy goal programming-based approach	No. of machine, tools available, part type and operations, set-up cost, material handling cost, batch size, machining cost, machining time	Total machining cost should be less than goal machining cost. Total set-up cost should be less than goal set-up cost. Total material handling cost should be less than goal material handling cost.	Machine tool and operations
2007	Rao	A comparison of TOPSIS Method, Modified TOPSIS Method, AHP, WPM, SAW Method, GTMA and fuzzy MADM	Total purchasing cost, total machine floor space, total machine no. and productivity of constructed FMC	Maximum score	CNC machine
	Çimren, Çatay, and Budak	A decision support system using multi-criteria weighted average and AHP methods	Productivity, flexibility, safety and environment, adaptability	Maximum score,	Machine tools

Table 2.1 Previous studies in area of machine selection (Cont.)

Year	Authors	Techniques	Parameters	Objectives / KPI	Machine Types
2008	Önüt, Kara and Efendigil	A hybrid method of fuzzy AHP and fuzzy TOPSIS approaches	Cost, operative flexibility, installation easiness, productivity, compatibility, safety, maintainability & serviceability, compatibility and user friendliness	Maximum CC*	Machine tool
2009	Iç and Yurdakul	Fuzzy analytical hierarchy process and Fuzzy TOPSIS namely MACSEL to select machining center	Spindle speed, power, tool change time, positioning accuracy, tool no., tool diameter and table size	Maximum score	Machine center
2010	Al-Hawari et al.	Fuzzy logic on MCDM approach	Error, accuracy, scanning angle, angular resolution, scanning frequency, data interface and transfer rate, cost, weight, power consumption, calibration procedure and user's experiences	Maximum Center of gravity value (COG)	Laser scanner
	Tsai et al.	MCDM approach	Capability, space, maintenance and service, safety, environment, reliability and cost	Maximum overall preference value (capability, cost, reliability, maintenance & service, safe & environment, space)	CNC machine tools

Table 2.1 Previous studies in area of machine selection (Cont.)

Year	Authors	Techniques	Parameters	Objectives / KPI	Machine Types
2011	Guldogan	Knowledge-based expert system and genetic algorithm	Daily capacity, cost, space, precision, cutting thickness, reliability, safety and environment, maintenance and service	Minimize cost, machine variety, maximize machine/operation suitability	Wood-cutting machine

Some papers aimed to find out a suitable machine before purchasing, they also considered machine's price. Others are the selection of machine tool, process parameters and operations, and construction project. Thus, there is no application for injection molding machine selection where any selection based on precision-based methods (concept of accurate measure and crisp evaluation) and the unique decision criteria of it. Other research applications have used many methods such as simulated annealing, fuzzy algorithms, expert systems, goal programming methods, neural networks and genetic algorithms, etc. All of them require a lot of knowledge and higher advanced computation than human brains. Therefore, a simpler and systematic approach is the better alternative choice (Rao, 2007). Typically, the selection process of an injection molding machine has to deal with qualitative and quantitative mixing categories that are difficult to be transformed into mathematical models. An expert system is the system that has individually been made decision by itself. It initiates alternative approaches in Artificial Intelligence or Intelligence Machine to more possibility. Generally, the modeling and analysis of decision systems with fuzzy set application has been applied in a large number of production management researches. Because, it has ability to deal with imprecision both quantitatively and qualitatively variables like human's decision (Guiffrida and Nagi, 1998). Moreover, fuzzy sets theory applications utilize development costs and time better than other AI techniques even same results, less decision maker's knowledge-base requires and more realistic.

We have found no research attempted to investigate in the injection molding machine selection between planning process. Almost rather interested in scheduling and process parameter optimization as in Table 2.2. And these scheduling approaches have not mentioned on machine allocation by matching the specification of each machine to desirable product.

During 1998 to 2009, we could find out more 400 application articles that mainly applied fuzzy set theory to production and operation management. Most of them were dedicated in the area of capacity planning, scheduling, inventory control, and product design researches. Their approaches were majoring on the combination of fuzzy logic with other Artificial Intelligence techniques such as genetic/evolutionary algorithms and neural networks. Nevertheless, all applications do not restrict only one type of fuzzy techniques but depending on the problem's characteristic. There were many publishing researches interested in selection process modeling using fuzzy logic and expert system. They determined the statements of high benefits to handle imprecision and vagueness that this ensured the attractive performance to handle any imprecision and uncertainty of the manufacturing environment. Table 2.2 indicated that although MCDM or MADM approaches were implemented in selection problems similar with fuzzy logic. But the core competency of fuzzy logic is applicable and practicable, this induced fuzzy logic is more preferable in industrial environments where the high complexity algorithm and time consuming are unacceptable.

Table 2.2 Examples of previous studies in area of injection molding machine

Type of problems	Authors	Techniques
Scheduling	Nagarur et al. (1997)	Goal programming
	Harris et al. (1998)	shared more than one job on a machine
	Süer et al. (1999)	-
	Dastidar & Nagi (2005)	Mathematical model
	Cao et al. (2005)	Heuristic search algorithm based on Tabu search
	Huang et al. (2011)	Sequential Genetic Algorithm (SGA)
Process parameter optimization	Kwong et al. (1997)	Case-based reasoning
	Shelesh-Nezhad, & Siores (1997)	Case-based reasoning
	He et al. (1998)	Fuzzy logic and other techniques
	Bozdana & Eyerci˙og˘lu, (2002)	Expert system
	Mok & Kwong (2002)	Fuzzy logic
	Metaxiotis et al. (2003)	Fuzzy logic
	Nagahanumaiah et al.(2008)	Fuzzy logic
	Lin & Lian (2010)	Fuzzy logic

2.2.2 Fuzzy logic core competency

We could synthesize beneficial properties of taking on fuzzy logic to be the solution for injection molding machine selection via previous researches as following issues,

1) Broad applicable and practicable in selection problems

2) Flexibility to modify by user without high knowledge background

3) Realistic

4) Handle imprecision and vagueness in the dynamic environment of industrial

5) Lower computational time than other AI techniques

However, this term does not represent the best answer when the manufacturers have to face with unsteady decision making. Nevertheless, fuzzy logic implementation is a foundation of user-friendly, realistic process, understandable for un-expert decision maker, minimization of programming and computation times, enhance planning process by decreasing human's bias in general organization (Hanna and Lotfallah, 1999; Chujo et al., 2005) and adaptable with another system as in Harris et al.'s research. However, this technique most invented in academic researches and a few of business improvement projects (Wong & Lai, 2011). Thus, more intelligence applications in industries are necessary.

This research aims to be a pioneer who applies the benefits of fuzzy set theory via fuzzy logic system in developing the automated plastic injection machine selection system. This system will enhance and gaining more profitability of production planning by decreasing planner's obligation and errors from experienced variations. The details of fuzzy set theory will be discussed further in next section.

2.3 Injection molding machine selection

This topic emphasizes on general information about the injection molding machine selection, common selection criteria of its, impact of inappropriate injection molding machine selection on products, and common defects in the injection molding industry.

2.3.1 General information

The real-world case of machine selection, the injection machine is obviously representative of complexity machine selection process. Their scheduling is an instance of flexible job shop scheduling problem (FJSS) class which a finite set of customers' orders will be assigned on a finite set of machines (Tanev et al., 2004). But other researches had judged in parallel machine scheduling (Nagarur et al., 1997; Huang et al., 2011; Cao et al., 2005).

The injection molding is a huge market section in the world. One of statement has been mentioned regarding in the assessment the molders economic index (MEI) value indicated. In the United States, the popularity of medical molding industrial is caused by growing of healthcare-service consumption along with an aging population rapidly. The injection molding plants generally involved with highest productivity rates, highly advanced product design and materials technology (Dastidar and Nagi, 2005). New companies often maintain their economic condition by keeping away from spending unnecessary budgets to set up a new plant with many machines. They are going to select some of them to ensure the profitability caused by higher demand and sufficient capitals. After that they will let new machine's specification launch variously. Injection molding machine uses Clamping Force (ton) to refer machine's capacity. The opening-closing direction of mold in each of these machines is controlled by their tie bars (Figure 2.2) which different dimension of tie bars will specify mold's dimension (Wang, et al., 2005; Dastidar & Nagi, 2005).

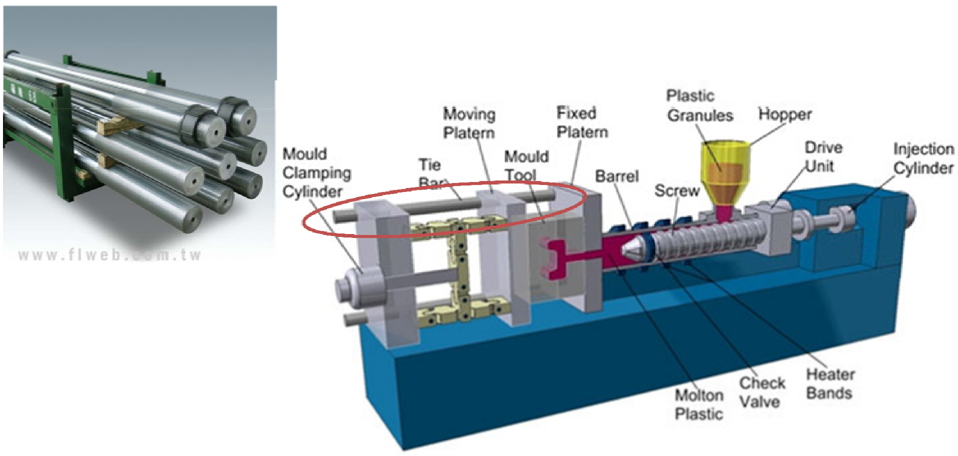


Figure 2.2 Tie bars of the injection molding machine

Source: fulongweb.en.alibaba.com and www.crown-cm.com/plastic-injection-molding.asp

An injection molding machine could have similar or different dimension of tie bars and some of them might have overlapping range of clamping force. Therefore, a company requires variety machine's specifications to variety support in a wide range of customer orders. Typically, the injection molding operation is a single-stage manufacturing process. A number of products will be produced on shared machines with their specific mold (Dastidar and Nagi, 2005). Many manufacturers used their machine with more than one product colors because a mold (or die) is expensive (10,000 dollars each). Thus, the number of mold models available was limited. Molds should be considered as secondary resources, but the previous researches said secondary resources are not restricted (Chen, 2007). One mold is able to use with more than one injection molding machine with different machine's capacity and machine's dimension. But the available mold or the tie bar dimension were constraints. As the result, a mold can be shared in many machines with a specific range of distance between tie bars. However, manufacturer prefers to assign a mold to a certain machine after optimal condition was determined. Therefore, these referred to the one-to-many relationship between the goods and molds and between the molds and the machines (Tanev, et al., 2004).

2.3.2 Injection molding machine selection criteria

In 1997, Nagarur et al concerned with scheduling for injection molding machine with mold-machine compatibility as a major factor. They called that the compatibility factors which include both technical and economic factors.

- Technical factors, including of tonnage, pressure, open distance, tie-bar space, mold dimension, temperature, etc.
- Economic factors, including of shot weight size

Some research referred clamping force as a rule of injection machine selection decision to process a product (Nakason, 2004). It is an internal control variable which has high potential to influence on product quality (Lin and Lian, 2010). In a plastic manufacture case study, the way of machine selection requires more than knowing clamping force as supported by Rao's research. The most suitable machine cannot be selected by single criteria, but it needs to deal with technological, economic, ethical, political, legal and social factors (Rao, 2007). Although machine size which is stated as clamping force have proved to be the main criteria but the others still be used differently in each manufacturer. Presently, there are no other criteria in published articles except clamping force.

2.3.3 Impact of the inappropriate selection of the injection molding machine

The factors in the occurrence of defects in injection molding do not only plastic resin treatment condition, process parameters setting. But also the selection of the injection molding machine is also (Yusoff, Rohani, Hamid & Ramly, 2004). Because, inappropriate machines effect to decrease product quality, machine performance and injection mold both directly and indirectly. The major solutions were the improvement materials preparation, machine modification and mold maintenance. It supports with the saying that the successful injection molding is officially depends on three things that have mentioned (Gordon, 2010; Bown, 1979).

Gordon (2010) has found that the problem in plastic injection involves with the matching properties. For example, short shot is the character of the indented product or uncompleted part caused by the variation and low quality of virgin resin. It induces the cooling and solidifying in the media before it will be filled in a cavity (the hollow space in the mold where the plastic is formed to shape) of the mold. In the view of machine, part stuck, machine error, pump run off, mold blockage and mis – alignment are the reasons. Yusoff, Rohani, Hamid and Ramly (2004) illustrated the relationship between process parameters and short shot parts. These parameters consist of backpressure, manifold temperature, holding pressure transfer, and screw rotation speed. The parameter of pressure can be inferred from the insufficient clamping force of the machine have applied to injection mold. In opposite, too low clamping force (ton), too high injection pressure and incompletely mold setup cause flashing of the product. By the factor of perfectly mold setup is the tie bars. Tie bars are four steel rods that control the direction of closing unit. The tie bar corrosion should be investigated regularly to avoid the leakage of melting plastic from the closing mold. Voids and sink marks are the shrinkage of injected parts after passed through the nozzle or partly cooling down inside the barrel. According to avoid these defects, shot size should be increased. This solution also improves the amount of poor surface detail. Discoloration and delamination will appear when the type and color of resin has changed on the next batch. Furthermore, short of purging time, unsatisfactory cleaning of screw barrel and nozzle are factors as well. Sometimes, delamination is caused by over shot size of machine. Therefore, appropriate shot size is able to reduce both delamination and poor surface detail. Other effect of insufficient purging for screw barrel nozzle is burn marks (on part). The different melting point of plastic can effects to the contaminated low melting point plastic burned in high melting point plastic. To reduce the risk of burn marks on the product, the similar type and color of material should be used on the same machine. Thus, it cannot hesitate to say that selecting the right machine can avoid the risk of more one defect at once. For the details of defect descriptions and causes will be placed on the next section.

2.3.4 Common defects in plastic injection molding industry

The defects in plastic injection have a variety of types and can be classified by the appearances. Some of them might have similar causes and some might not. The plastic manufacture usually found some defects but other manufacturer may not find the same because the opportunity of occurrence is different. Inspector and quality staff are persons who have the authority to identify the defects of a part. A defective can be identified more than one defect depends on the clarity of each defect and knowledge of the inspector. Thus, if the quality staffs have a better understanding of the characteristics and factors in the occurrence of each defect, the determination of problems and solution are able to be enhanced. From the studies, the most common defects in the plastic industry have collected as follows.

1) Air bubbles — Air entrapped in the molded part. The empty space in the rigid molded part (Figure 2.3).

Causes: Moisture, entrained air or not typical material flow pattern.

Figure 2.3 Air bubbles

2) Black dot or streaking — Black spots or brown streaks appear on the surface of molded part (Figure2.4).

Causes: Contaminated plastication system or improper drying.

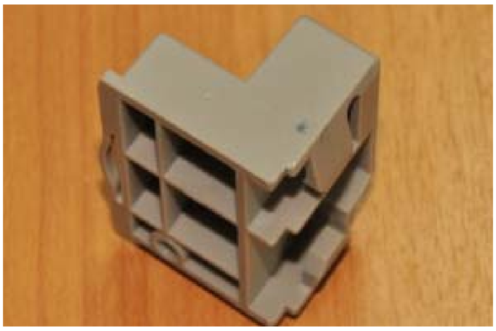


Figure 2.4 Black dot

3) Brittleness — The tendency of a molded plastic part to break or crack under conditions in which it would not normally do so.

Causes: Processing temperatures too high, contaminants/fines present, improper drying, wet resin, improperly sized cylinder, poor mold design or too high regrind.

4) Blush, frost, and folds — A clouded discoloration normally found at gate locations, but they can be found anywhere on the part surface.

Causes: The stress which caused by process conditions or mold dimension lead to melt fracture around the gate.

5) Burn mark — A degraded or oxidized plastic part in a molding. It shows as small dark brown or black discolorations on the surface (Figure 2.5).

Causes: Excess shear heat, too high melting temperature or low temperature pigments.

Figure 2.5 Burn mark

6) Cracking — Fine cracks or deeper cracks on the surface of a molding (Figure 2.6).

Causes: High stress in mold or too high external force for the part, incompatible chemical that applied to the finished part.

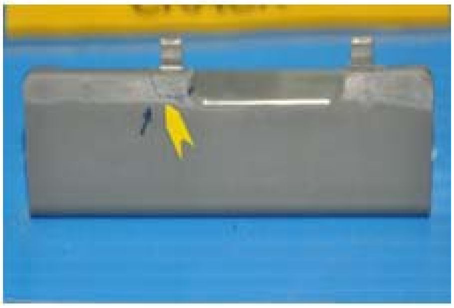


Figure 2.6 Cracking

7) Discoloration — The color of the material changes from the standard (Figure 2.7).

Causes: Improper size of cylinder, excess processing temperature or contamination.

Figure 2.7 Discoloration

8) Distortion (warping) — The alteration of the original shape of the molding from that required (Figure 2.8).

Causes: Irregularly packing density, no uniform flowing of material, not enough cooling in ejection, improper knockout system or poor part design.

Figure 2.8 Distortion (warping)

9) Delamination (Pits, Orange peel, Wrinkles) — The layers of melted resin which separate from the main selection of a molding, look like the onion skins.

Causes: Low temperature process, improper drying or contamination.

10) High gate — A long feathery string of plastic are the little tip left on the part between the runner and the part after ejection: it is called a high gate (Figure 2.9). The string or high gate vestige gives the negative look and feel of the product when it is contacted with the user's hand (Pötsch and Michaeli, 2008).

Causes: Too high nozzle temperature, too early gate freeze off, too high decompression time, poor heat insulation at gate or poor valve gate operation.

Figure 2.9 High gate

11) Pin mark (Ejector mark) — The streak left by the ejector pin on the surface of molded part (Figure 2.10).

Causes: Not enough cooling of material during the ejection process.

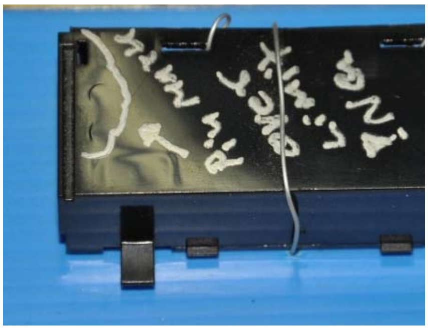


Figure 2.10 Pin mark

12) Excess shrinkage (small parts) — The shrinkage of the parts.

Causes: Improperly balanced mold temperatures (colder on movable half), surface irregularities in the mold (polish cavity surfaces) or improperly cycle time.

13) Flashing (on parting line) — The penetrated material on the edge of a molded part (Figure 2.11).

Causes: Too high internal cavity pressure, the damage on the contacting surface of two sides of the mold from over packing, insufficient mold clamping force, too high melt temperature or mold malfunction.

Figure 2.11 Flashing

14) Flow mark (flow lines) — The areas of dark streaks or lines that are generally found during the cavity filling process for some plastics like POM or PE. Directionally "off tone" wavy lines or patterns (Figure 2.12).

Causes: The material may cool before the mold cavity is full or injection speed too slow (Bociąga and Jaruga, 2006).

Figure 2.12 Flow mark

15) Gate smear — The surface flaw appearance on the molded part.

Causes: Unsuitable gate location, too much injection fill rate or small gate size.

16) Jetting — A ripple pattern on the surface of a molded part, usually emanating from the gate area.

Causes: An irregularity spreading of melt into cavity at high injection speeds.

17) Nozzle drool — Resin leakage from the nozzle or from the nozzle area into the mold during the injection phase (Dominick et al, 2000).

Causes: Nozzle too hot, barrel front zone too hot, pressure or plastic trapped between the nozzle tip and mold bushing.

18) Part dimensions (too large) — Part is too large when the specification was compared.

Causes: Poor mold design or overpacking.

19) Poor surface finish — Scars, gloss or wrinkles that appear on the surface of the molded part.

Causes: Rubber particle aggregation from improperly temperature and packing pressure, the shear/extensional stress that generated in the melt during mold filling from the frozen part of the melt or thermoplastic stresses that produced from the material that has frozen in the interior of the plastic (Provatas, Edwards and Choudhury, 2002).

20) Over cut – The indented finish part from cutting process (Figure 2.13). Causes: The molded part is trimmed over by human or into a tool using negative or positive pressure.

Figure 2.13 Over cut

21) Odor (gassing) – The odor of burning material which is contaminated with the plastic resin. To identify the type of contaminants, burning characteristic will be considered together with the odor.

Causes: High melt temperature, contamination or resin holdup spots

22) Freeze – off — Material freezing off before complete mold filling

Causes: Processing temperatures too low or improper nozzle design

23) Scratch — White appearance or a large size flaw from surface damage under scratch condition (Figure 2.14).

Causes: Under the stress raiser restricting the applicability during tensile, impact or fatigue loading.

Figure 2.14 Scratch

24) Scuffing — A roughness on the surface of molded part.

Causes: Material decomposition (Provatas, Edwards and Choudhury, 2002) or amorphous resins that is still hot and soft when ejected from the mold cavity.

25) Short shots — An incomplete molded part (Figure 2.15).

Causes: Insufficient melt volume, high - pressure drop in mold, blockage in flow at the feed-throat, inadequate shot size or mold malfunction.

Figure 2.15 Short shot

26) Splay or silver streaking — A “splash-like" appearance on the surface of a molded part, mainly occur near the gate (Figure 2.16).

Causes: Contaminated material.

Figure 2.16 Splay or silver streaking

27) Sink mark — An unwanted depression in the surface of a molding during the cooling process if some area is not cooled adequately (Figure 2.17).

Causes: Poor part design, mold malfunction, undertaking, insufficient part packing process or tooling

Figure 2.17 Sink mark

28) Staining — A dull or discolored area of a mold.

Causes: Contamination by foreign matters or other resin, contamination by eroding material of the molding machine or fats/oils in contact with the melt.

29) Sticking (cavity) — The sticking in the molding by mechanical or vacuum adhesion of the molding to the surface.

Causes: Too high polish, too low polish, scratch in the mold, over packing in the mold, degraded materials, crazing of the part surface, insufficient ejection force, insufficient part cooling, too high nozzle temperature, and too high melt temperature (Bozzelli, 2007).

30) Sticking (sprue) — The cold runner that blocks the sprue gate.

Causes: Scratch in sprue, overpacking, sprue too soft, sprue not frozen, too high nozzle temperature, and too high melt temperature (Bozzelli, 2007).

31) Voids — Voids are hollows in the thick sectioned part. Voids occur when material is flowing from hot center section to cold mold walls. Material volume shrinks after gate or other melt - path areas have frozen (source: http://www.kenplas.com/service/imtroubleshooting.aspx#Voids).

Causes: Insufficient part packing process or tooling.

32) Weld lines — The distinct dark lines on the surface of injection molded part where the confrontation of two or more melts flow possibly had occurred.

Causes: Nonuniform material flow, poor mold design or mold malfunction.

33) Window — An area of transparent material in an otherwise translucent molding. The large knit or weld lines.

Causes: The two or more flow are flowing around an object standing proud in a plastic part and meets together at the end of filling (http://en.wikipedia.org/wiki/Injection_molding).

34) Witness mark (Faint mark) — A mark or line on a molding where different parts of the mold come together or 'mate'. It will occur along the parting line. Causes: Flow stagnation during injection-compression transition. Two mating mold components

35) Unmelted — Some articles or weak part that contaminated on the surface of the molded part.
Causes: Surface defect or weak part.

2.4 Fuzzy logic

Experts usually use their common sense to solve the problems that involved with indistinct terms. For example, they might say, "The temperature is slightly high, we can accept it for a while." (Negnevitsky, 2005; Ross, 2010). Other experts have no difficulties to understand this statement if they have the background with the kind of problems. But, how to use computer to represent the expert knowledge vague and ambitious terms is still difficult. In 1930s, Jan Lukasiewicz introduced the fuzzy logic (or multi-valued logic) by studying the representation of fuzziness based on such terms as tall, old and hot. Instead of classical logic operates with only two values 1 (true) and 0 (false), Lukasiewicz proposed the interval range of truth values to all real numbers between 0 and 1. Instead of the probability, a number in this interval represents that whether a given statement was true or false. Later, in 1937, the first simple fuzzy set and basic fuzzy set operations were defined by Max Black. In 1965, 'Fuzzy sets' paper was published by Lotfi Zadeh. He explored the possibility theory into a formal system. An application of natural language terms which called 'fuzzy logic' was introduced by him also, let him became the master of fuzzy logic (Negnevitsky, 2005; Yen, Langari and Zadeh, 1994; Ross, 2010).

Fuzziness can be described by fuzzy logic based on the concept of degree of all things; such as beauty, age, height, temperature, distance, etc. It illustrates how people think and models the meaning of words, decision making and common sense in formal pattern. It is a new, more human-liked, intelligent systems. By definition, fuzzy set theory covers fuzzy logic, fuzzy arithmetic, fuzzy mathematical programming, fuzzy topology, fuzzy graph theory, and fuzzy data analysis (Wong and Lai, 2011). In the same way as classical logic, in application, fuzzy logic have been integrated into areas of industrial process control through consumer product, aerospace and Bioengineering. Fuzzy logic plays a similar role in shaping a suitable rule-based, or linguistic, control strategy. There are succeeded applications and implementations of fuzzy set theory for years. Then, the development of the fuzzy set theory has been supported by a lot of national and international associations, i.e. the International Fuzzy Systems Association (IFSA), the Spanish Association of Fuzzy Logic and Technologies. The using of fuzzy set theory as an important methodology for modeling and analyzing decision systems have particularly interested in industrial engineering research due to the ability to model both quantitatively and qualitatively problems. It can handle the modeling gaps in descriptive and prescriptive decision models.

In the manufacturing section, production planning and control (PPC) techniques are employed in various dimensions. There are a high number of publications in fuzzy production planning and control researches after Zadeh's research in 1965. Fuzzy logic applications were distributed into 2 major groups that are business sector and social problems. And decision support system application on production planning and scheduling is judged in the business sector. The benefits of fuzzy logic by Zadeh are accurate, reliable, flexible and manageable the uncertainty of humans. Next challenges are fuzzy logic users should feel more trust than non-user in handling with fuzziness of real data to be more satisfied and wider application (Metaxiotis, Psarras, and Samouilidis, 2003).

In 1994, Guiffrida and Nagi surveyed all articles that relied on fuzzy set theory against production management research, they have found 695 papers using it in the way of Fuzzy multiple objective decision making and 450 papers of production control. Guiffrida and Nagi referred to Karwowski and Evans's statement regarding to the disturbances of the decision maker's mental model by imprecision and indistinctly. They require experience and massive data to create assessment concepts (Guiffrida and Nagi, 1994). Mula et al. had reviewed published researches for 1983 to 2004 concerning in uncertainty modeling in production planning problems. They presented models of fuzzy set theory and fuzzy logic that applied in manufacturing uncertainty. Fuzzy modeling approach was proposed via aggregate planning, material requirement planning, manufacturing resource planning, inventory management and supply chain planning (Mula et al., 2006).

Moreover, it was a good cooperator with artificial intelligence in planning on constructing a production plan and it is an appropriate methodology making headway with this science (Metaxiotis, Psarras and Samouilidis, 2003; Guiffrida and Nagi, 1998).

2.4.1 Fuzzy logic system

After the introduction of the fuzzy set concept as a math tool for dealing with uncertainty by Zadeh in 1965. Fuzzy logic is used to reduce the gap between symbolic processing and numeric computation. They form the basis of control strategies, including decision making or supervisory control. In 1985, the first logic chip at Bell Telephone Laboratories was developed by Masaki Togai and Hiroyuki Watanabe. This design guided fuzzy computers. Their structures are more similar to the human brain than the present-day computers.

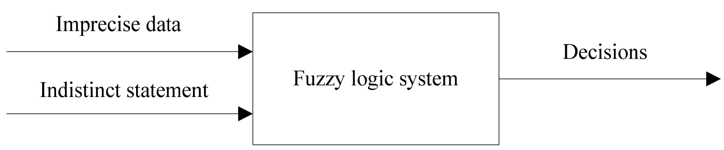


Figure 2.18 A fuzzy logic system (Negnevitsky, 2005)

The fuzzy logic system enhances real world problem solving management, which contains high complexity and fuzziness. Fuzzy sets help to model the uncertainty involving with ambiguity, imprecision, and information insufficiency. The process of fuzzy logic system utilizes all imprecise data and indistinct statements and converts to suitable decisions as in Figure 2.18. Negnevitsky (2005) referred that "Fuzzy logic is determined as a set of mathematical principles for knowledge representation based on degrees of membership rather than on crisp membership of classical binary logic".

From Negnevitsky (2005), fuzzy Logic is the mechanism for reasoning with fuzzy rules, rules take the form:

"If x_1 is $\tilde{A}_1$, x_2, is $\tilde{A}_2$, …, x_n is $\tilde{A}_n$, then y is $\tilde{B}$"

2.4.2 Fuzzy set

A fuzzy set is defined as a set with fuzzy boundaries (Negnevitsky, 2005) or not obvious boundaries. And let any uncertainty can be modeled; ambiguity, imprecision, and lack of information regarding a problem or a plant, etc.

2.4.2.1 Mapping of classical sets and fuzzy sets to the functions

Generally, it is mapping of elements or subsets on one universe of discourse to the elements or sets in another universe. When X and Y are two different universes of discourse, if an element x is contained in X and corresponds to an element y contained in Y, it is represented as $f : X \rightarrow Y$ (the mapping from X to Y).

If X is universe of discourse and its elements are x, in classical set theory, crisp set A of X is defined as function $f_A(x)$ called the characteristic function of A.

$$f_A(x) : X \rightarrow 0,1,$$

where

$$f_A(x) = \begin{cases} 1, & if\ x \in A \\ 0, & if\ x \notin A \end{cases}$$

Fuzzy set contains the elements. These elements vary degrees of membership in the set. Nevertheless, members of a crisp set would not be members unless their membership was completed in that set. Elements in the fuzzy set can also be members of the other fuzzy set on the same universe. A real number value is mapped with fuzzy set. If *x* is a member of fuzzy set *A*, then the mapping is as below:

$$\mu_A (x) \rightarrow [0, 1].$$

The fuzzy set A of universe X is defined by function $\mu_A(x)$ called the membership function of the set A

$$\mu(x): X \rightarrow [0,1],$$

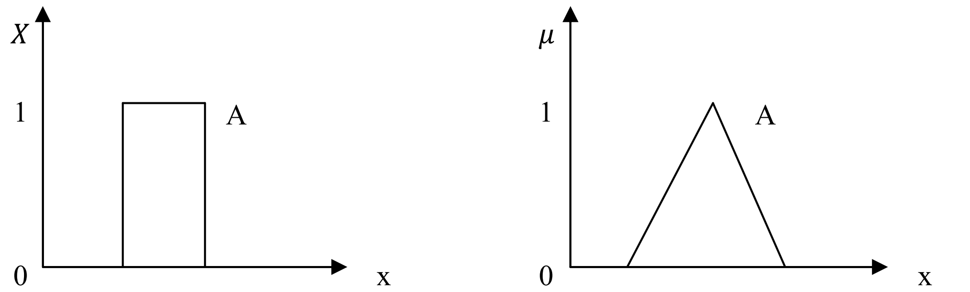
where

$\mu_A(x) = 1$ *if x is totally in A;*

$\mu_A(x) = 0$ *if x is not in A;*

$0 < \mu_A < 1$ if x is partly in A.

The membership value will be "1" if it belongs to the set. But the membership value will be "0" if it is not a member of the set. Thus, membership in a set is binary. The membership of the crisp set A and fuzzy set A are shown in Figure 2.19 and Figure 2.19 respectively.

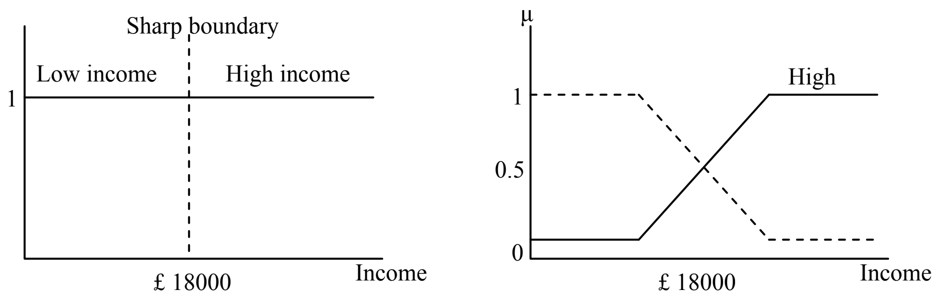


(a) Membership function of crisp A (b) Membership function of fuzzy set A

Figure 2.19 Mapping of membership function of (a) crisp A and (b) fuzzy set A

The degree of membership is the value in the range 0 to 1, can be called a degree of membership or membership value, but less sharp from the set in classical theory. Figure 2.20 is the set of high income dividing in two views of set in classical theory and fuzzy set. The left figure shows the sharp boundary of low income and high income, the right figure is gradually changing of boundary. Number 1 represents the income is completely in the set of high income, if the value is 0, means that income value is not in the set of high income.

Note that each and every term in the term set is defined as a fuzzy set via its membership function, e.g. $\mu_{High} : X \rightarrow [0, 1]$.



(a) Set in classical theory (b) Fuzzy set

Figure 2.20 Set of high income is illustrated as in (a) set in classical theory and (b) fuzzy set

Fuzzy and crisp sets can be also presented as shown in Figure 2.21. The boundaries of crisp set are sharp, but the fuzzy subset is ambiguous. Crisp set is used to explain no uncertainty and fuzzy set is used for uncertainty situations. The transition of fuzzy set can be gradually changed.

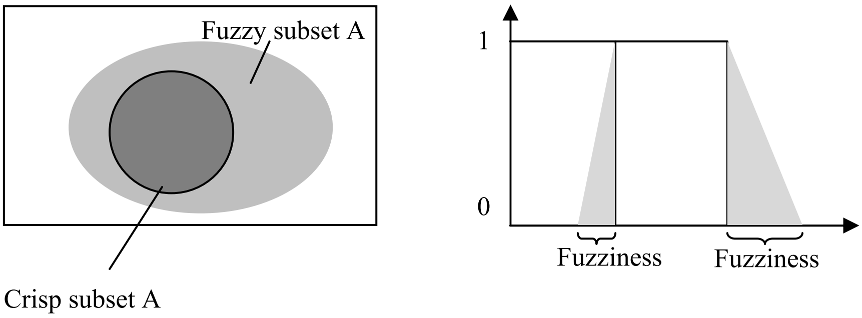


Figure 2.21 Fuzzy and crisp sets (Negnevitsky, 2005)

2.4.3 Membership function

The characteristic of the membership function is composed of three properties i.e. core, support and boundary. The Figure 2.22 illustrates those properties.

2.4.3.1 Core

The membership value in this region is 1 or it means full membership (1) in the set *A*. The elements, which have the membership function as 1, are the elements of the core, i.e., here $\mu_A(x) = 1$.

.

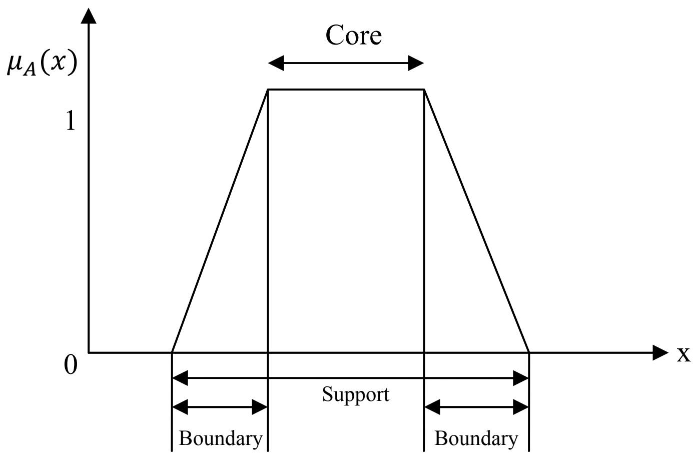


Figure 2.22 Features of membership function (Negnevitsky, 2005)

2.4.3.2 Support

If the region of universe is characterized by nonzero membership in the set *A*, this defines the support of a membership function for fuzzy set *A*. The support has the elements whose membership are greater than 0.

2.4.3.3 Boundary

If the region of universe has a nonzero membership, but not full membership, this defines the boundary of a membership; this defines the boundary of a membership function for fuzzy set *A*:

The boundary has the elements whose membership are between 0 and 1, $0 < \mu_A(x) < 1$. These are the standard regions defined in the membership functions.

The membership function needs to be calculated first and then gaining of outputs. The construction of membership function has to define a parametrizable membership function, not point to point which is good to use in major of triangular membership function and trapezoid membership function because of good solution output for application. If the learning of membership function with neural network is required, the Gaussian membership function will be a good choice.

To construction of fuzzy set, one of the most practical approaches relies on the knowledge of a single expert. The expert will be interviewed for his knowledge.

2.4.4 Linguistic variables and hedges

The linguistic variables are basic of fuzzy set theory and used in fuzzy rules. The probable range value of a linguistic variable indicates the universe of discourse of that variable. For example,

If speed is slow then stopping_distance is short

The advantages of linguistic term

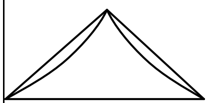
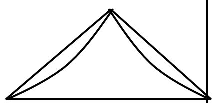
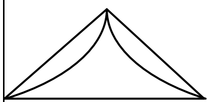
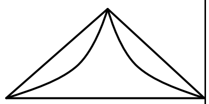
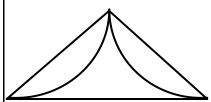
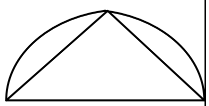
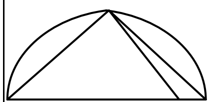
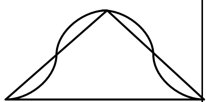
1) Easier to express knowledge in linguistics term than human experts.

2) More understandable.

In addition, hedges are terms that modify the shape of fuzzy sets such as very, somewhat, quite, more or less and slightly. Hedges can be the verb, adjectives, adverbs modifiers or even whole sentences and also break down continuums into fuzzy intervals and more clearly show a fuzzy set overlap which a human could not identify. The examples of an internal pressure in the mold fuzzy set are very high, moderately high, slightly high, neutral, slightly low, moderately low and very low

The applications of hedges give mathematical modification in the fuzzy sets as in Table 2.3.

Table 2.3 Representation of hedge in fuzzy logic

Hedge	Mathematical expression	Graphical representation	Hedge	Mathematical expression	Graphical representation
A little	$[\mu_A(x)]^{1.3}$		Slightly	$[\mu_A(x)]^{1.7}$	
Very	$[\mu_A(x)]^2$		Extremely	$[\mu_A(x)]^3$	
Very very	$[\mu_A(x)]^4$		More or less	$\sqrt{\mu_A(x)}$	
Somewhat	$\sqrt{\mu_A(x)}$		Indeed	$2[\mu_A(x)]^2$ if $0 \le \mu_A \le 0.5$ $1-2[1-\mu_A(x)]^2$ if $0.5 \le \mu_A \le 1$	

Source: (Negnevitsky, 2005)

2.4.5 Basic fuzzy set operations

This topic gives three basic operations of fuzzy sets.

1) Union

Definition Let $\tilde{A}$ and $\tilde{B}$ be fuzzy sets defined over U. Then their union, $\tilde{A} \vee \tilde{B}$, is defined in terms of $\mu_{\tilde{A}}(u)$ and $\mu_{\tilde{B}}(u)$ as $\mu_{\tilde{A}\vee\tilde{B}}(u) = max(\mu_{\tilde{A}}(u), \mu_{\tilde{B}}(u))$.

2) Intersection

Definition Let $\tilde{A}$ and $\tilde{B}$ be fuzzy sets defined over U. Then their intersection, $\tilde{A} \wedge \tilde{B}$, is defined in terms of $\mu_{\tilde{B}}(u)$ as $\mu_{\tilde{A}\wedge\tilde{B}}(u) = \min(\mu_{\tilde{A}}(u), \mu_{\tilde{B}}(u))$

3) Complement

Definition Let $\tilde{A}$ be a fuzzy set defined over U. Then its complement, $\neg\tilde{A}$, is defined in terms of $\mu_{\tilde{A}}(u)$ as $\mu_{\neg\tilde{A}}(u) = 1 - \mu_{\tilde{A}}(u)$.

By the way, the union, intersection and complement equivalent to OR, AND and NOT respectively. Figure 2.23 contains truth tables for more understandable the ideas.

A	**B**	**Min(A,B)**
0	0	0
0	1	0
1	0	0
1	1	1
AND		

A	**B**	**Max(A,B)**
0	0	0
0	1	1
1	0	1
1	1	1
OR		

A	**1-A**
0	1
1	0
NOT	

Figure 2.23 Standard logical operations and basic fuzzy set operations

The diagrams for fuzzy set operations are shown in Figure 2.24 which contain three basic operations of fuzzy set; Intersection, Union and Complement. Another operation such as Containment, describes a set can contain other sets; will not be stated in this research.

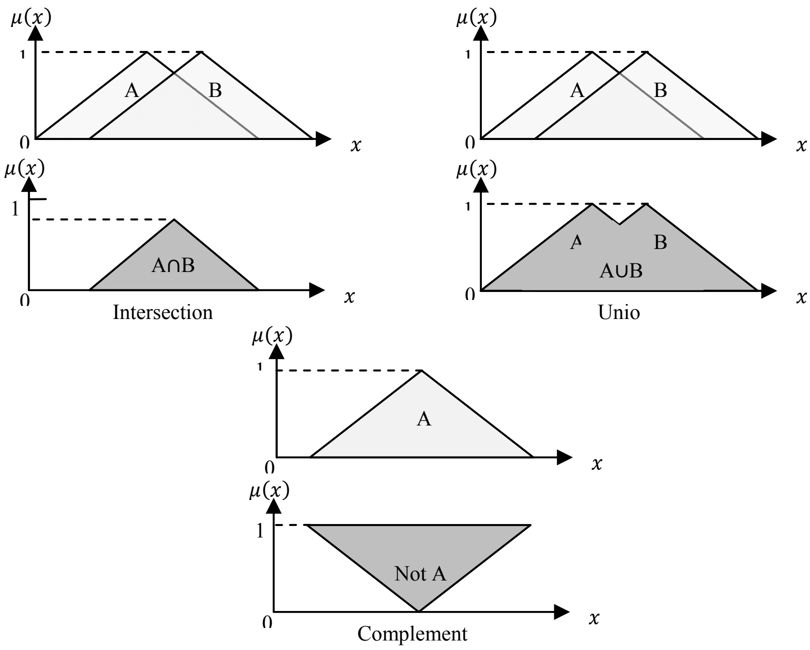


Figure 2.24 Basic operations of fuzzy sets (Negnevitsky, 2005)

2.4.6 Fuzzy IF-THEN rules

Lotfi Zadeh proposed new method by capturing human knowledge in fuzzy rules for the complex system. The fuzzy sets and fuzzy operators act as the subjects and verbs of fuzzy logic. Conditional statements, IF-THEN rules, make fuzzy logic useful, lead to the completely sentence (www.coursehero.com).

The structure of a single IF-THEN rule is as follows:

If x is A then y is B

Where,

- A and B are the linguistic term of a fuzzy set (for example High)
- x is the membership value of an income input ($\mu_{HIGH}(x)$)

By,

- 'x is A' is called the antecedent and describes a condition
- 'y is B' is called the consequent and describes a conclusion

And if using with MATLAB will be understood by form:

If x == A then y ==B

Both antecedent and consequent can have multiple parts. For example, in the laundry problem, the two inputs would determine the washing machine cycle, a rule with multiple parts in the antecedent could be:

If quantity if Small and softness is Soft then the cycle is Light

2.4.7 The difference between classical and fuzzy rules

As in Table 2.4, the difference of classical rules and fuzzy rule are exampled.

Table 2.4 IF-THEN rules in classical and fuzzy rule forms.

Classical IF-THEN rule		Fuzzy If-THEN Rules	
Rule: 1		Rule: 1	
IF	injection speed is > 100	IF	injection speed is high
THEN	stopping_distance is long	THEN	stopping_distance is long
Rule: 2		Rule: 2	
IF	injection speed is <40	IF	injection speed is low
THEN	stopping_distance is short	THEN	stopping_distance is short

A fuzzy rule can have multiple antecedents and multiple consequents also, for example:

IF clamping_force is low
AND distance_between_tie-bar is small
AND shot_weight is low
THEN machine is low

The output of each rule is a fuzzy set, but usually we need to obtain a single number representing the expert system output which is a precise solution. The method to transform these output fuzzy sets to be a single number will be detailed in section 2.4.8.

2.4.8 Fuzzy inference

Fuzzy inference is a process of mapping from giving input to an output, using the theory of fuzzy sets. Fuzzy inference techniques were proposed in 2 methods; Mamdani-style and Sugeno-style inferences. The Mamdani-style inference process is shown in Figure 2.25.

2.4.8.1 Mamdani-style inference

This method is the most commonly used since 1975. Professor Ebrahim Mamdani built the first fuzzy system to control the machines of steam engine and boiler. The process is performed in five steps:

1) Fuzzification of the input variables

The crisp inputs will be fuzzified against the appropriate linguistic fuzzy sets.

2) Rule evaluation

The fuzzified inputs will be applied to the antecedents of the fuzzy rule. The fuzzy operator (AND or OR) is used to obtain the result as a single number from multiple antecedents. This number is then applied to the consequent membership function. For the disjunction of the rule antecedents, they will be evaluated by the OR fuzzy operation which the union is typically used in fuzzy expert systems. The *max* and the *probor* build-in OR methods are contained in MATLAB Fuzzy Logic Toolbox.

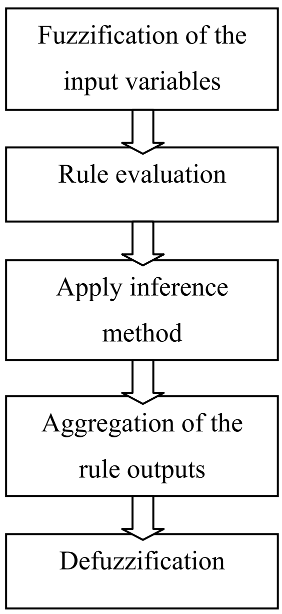


Figure 2.25 Mamdani-style inference process (Negnevitsky, 2005)

The probabilistic OR is also called *algebraic sum* which could be calculated from:

$$\mu_{A \cup B}(x) = probor\ [\mu_A(x), \mu_B(x)] = \mu_A(x) + \mu_B(x) - \mu_A(x) \times \mu_B(x)$$

The conjunction evaluation, the AND fuzzy operation *intersection,* is calculated from:

$$\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)]$$

or *prod* method in the Fuzzy Logic Toolbox,

$$\mu_{A\cap B}(x) = prod\,[\mu_A(x), \mu_B(x)] = \mu_A(x) \times \mu_B(x)$$

Note that different methods might cause dissimilar results. However, the customization of AND and OR operations allowed for user in most fuzzy packages.

3) Apply inference method

This process is the conclusion of the rule based on the degree of matching. The matching can be separated into two main methods: the clipping and the scaling methods. The clipping method, correlation minimum, is most common method. The scaling method, correlation product, is better method for maintaining the original shape of the fuzzy set.

4) Aggregation of the rule outputs

An input often matches with multiple fuzzy rules. To combine the inference results of these rules, the output of the aggregation process is one fuzzy set for each output variable.

5) Defuzzification

Defuzzification is the last step in the fuzzy inference process. In order to express the crisp number (single number) in the final output of a fuzzy system, the defuzzification is required.

There are two major defuzzification techniques: the Mean of Maximum (MOM) and the centre of gravity (COG) or centroid technique.

The Centre of Gravity (COG) – This is the most popular method to discover the point where a vertical line would slice the aggregate set into two equal masses to calculate the weighted average of a fuzzy set. In theory, the COG defuzzification is calculated over a continuum of points in the aggregate output membership function which can be expressed as follows:

$$COG = \frac{\int_a^b \mu_A(x)xdy}{\int_a^b \mu_A(x)dx}$$

But in practice, COG can be obtained by calculating it over a sample of points.

$$COG = \frac{\sum_{x=a}^{b} \mu_A(x)x}{\sum_{x=a}^{b} \mu_A(x)}$$

2.4.8.2 Sugeno-style inference

Michio Sugeno was first introduced this fuzzy inference. Sugeno-style fuzzy inference is quite related to Mamdani method, Sugeno changed only a rule consequent (Negnevitsky, 2005). Negnevitsky also indicated that the Mamdani-style inference require much computation time to find the centroid of shape. A comparison format of two inference processes is placed in the Table 2.5.

2.4.9 The comparison of fuzzy inference system

The Sugeno-style inference, the output of each fuzzy rule is constant. In the same way, all consequent membership functions are performed by singleton spikes. A zero-order Sugeno system might be suitable for some problem's needs which require singleton output functions.

Table 2.5 Comparison of two fuzzy inference processes

Mamdani-style inference	Sugeno-style inference
IF x is A1 AND y is B1 THEN z is C1	IF x is A AND y is B THEN z is f(x,y) or z is k
x, y and z are linguistic variables A1, B1, and C1 are linguistic values	x, y and z are linguistic variables A and B are fuzzy sets f(x,y) is a mathematical function k is constant

Source: Negnevitsky (2005)

For more clearly consider of applying fuzzy inference type, the comparison of advantage for Mamdani-style and Sugeno-style was shown in Table 2.6.

Regarding to trading-off advantages of these two inference styles, Sugeno style (Figure A.1) is suitable for problems with the functioning or mathematical is output. Thus, this does not relate to our automated machine selection system and does not support to our objective. Therefore, Mamdani-style inference system will be raised to be the solution method. Figure 2.26 shows the operation of Mamdani-style inference system.

Table 2.6 Advantages of two fuzzy inference processes

Mamdani-style inference	Sugeno-style inference
- Widely acceptance for capturing expert knowledge - Instinctive act - Similar with human's decision	- Efficient computational time - Suitable with optimization and adaptive technique - Very attractive in dynamic nonlinear systems - The output surface is continually - More suitable for the mathematical model output

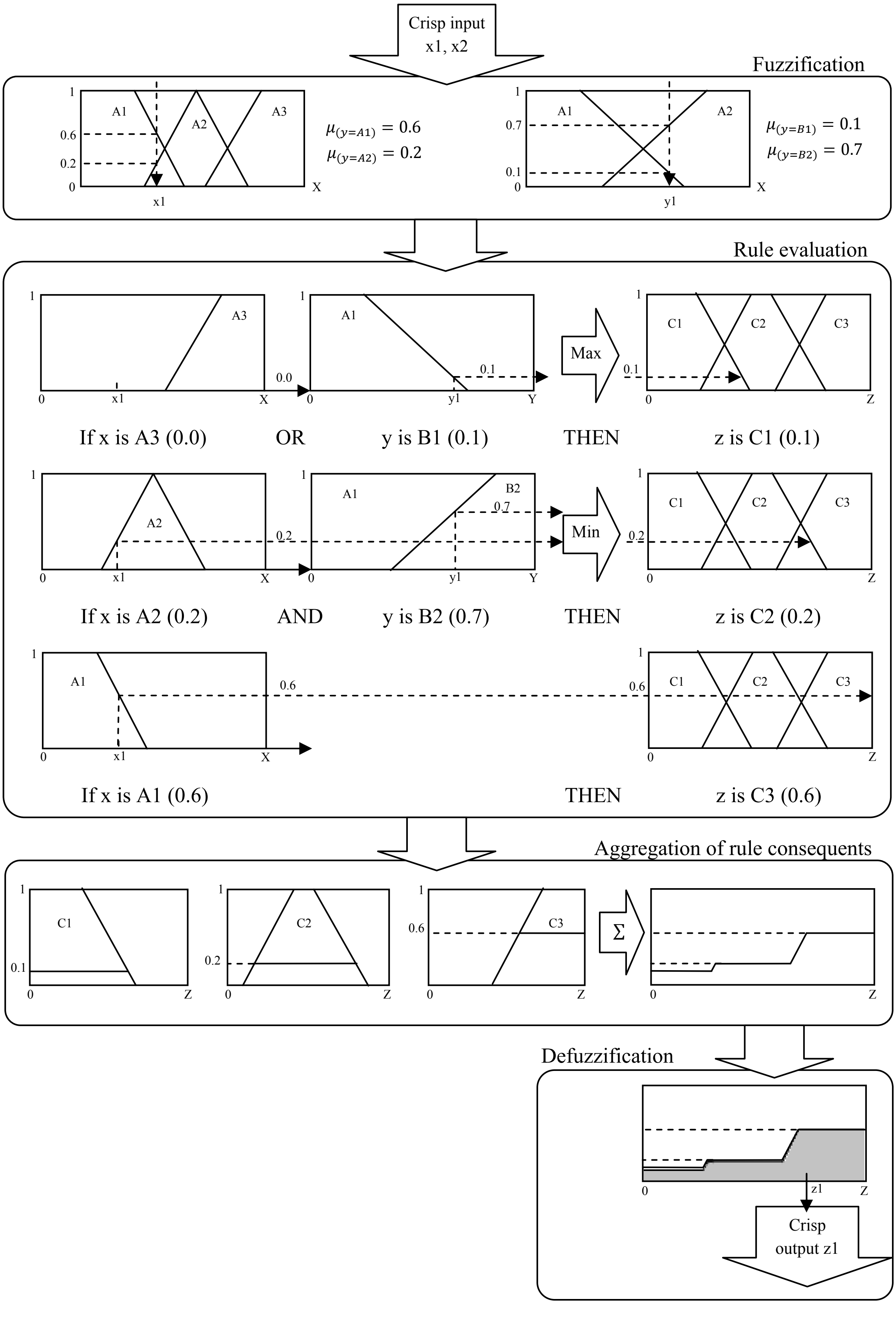


Figure 2.26 Mamdani-style fuzzy inference structure (Negnevitsky, 2005)

2.5 MATLAB

The previous topics concerned about background of problem, characteristic and solution method that involve in this research. Thus, the development platform would be described. This research, the computing program named MATLAB will be employed as a tool of research.

MATLAB is a Numerical Computing Environment program that has as own computing environment, and having own language for programming. Name of MATLAB is the combination of words ‘Matrix’ and ‘Laboratory’. It originated in 1970 as a Fortran language’s connection between LINPACK (the libraries that used in linear algebra computing) and EPISPACK (the libraries that used in Eigen Value and Eigen Vector computing) in accord to minimize user time to learn Fortran. In 1984, MathWorks rewrote MATLAB using C language and founded JACKPAC libraries. They have been developing continuously until including GUI (Graphical User Interface) and Simulink (the simulation solution) at present (Sa-Nguansatra, 2010). Mathematica is a one competitor of MATLAB. It is a type of symbolic computation program. However, MATLAB is more convenient because it is similar to common imperative programming languages, is a programming in terms of stating that change a program state, while Mathematica uses its own functional language.

MATLAB allows easily matrices manipulation, plotting, algorithm implementation, user interfaces creation and interface with programs in other languages. It is used in industry and academic and be able to compile on widespread operating systems such as Windows, Mac OS, Linux and Unix, etc. MATLAB applications, in area of classification and assessment problems that were proposed, present the possibility of applying their existing features on desired fuzzy system although have not discovered its late application on machine selection expert system (Pravadalioglu, 2005; Mahanty and Gupta, 2007; Wen, 2008; Akgun, in press; Li et al., 2010).

CHAPTER III
RESEARCH METHODOLOGY

This chapter contributes idea of research study area for selection of data source and collection, details of methodology for prototype system development and design.

3.1 Research methodology flowchart

The overall research steps are illustrated in Figure 3.1. It begins with observation and identification problem. And it is finished with conclusion and discussion. The inputs and outputs of each step are determined in this flowchart. This flowchart will be used to be a main guide through this research.

3.2 Detail of methodology

To ensure the study approach and scope, research methodology was identified in initial phase of working. All details of this research methodology are shown as the following steps:

3.2.1 Observation and identification problem

First of all, an identification of problem should start with desired area observation for taking on ideology from operators in shop-floor. In case of this research work, interviewing the planner from from a plastic manufacturer is conducted. Then, the existing work flow can be explored.

After the investigation had finished, the planning operation that the machine selection is involving can be generated. It is found that a planner has to deal with various orders per month manually and matches a suitable machine to those orders by using machines information database and working procedure from engineer department. Then, the planner rearranges production scheduling. This working process reflects production planner's role deeply. If the planning process have stopped immediately and have not sub-agent, production line might be disrupted.

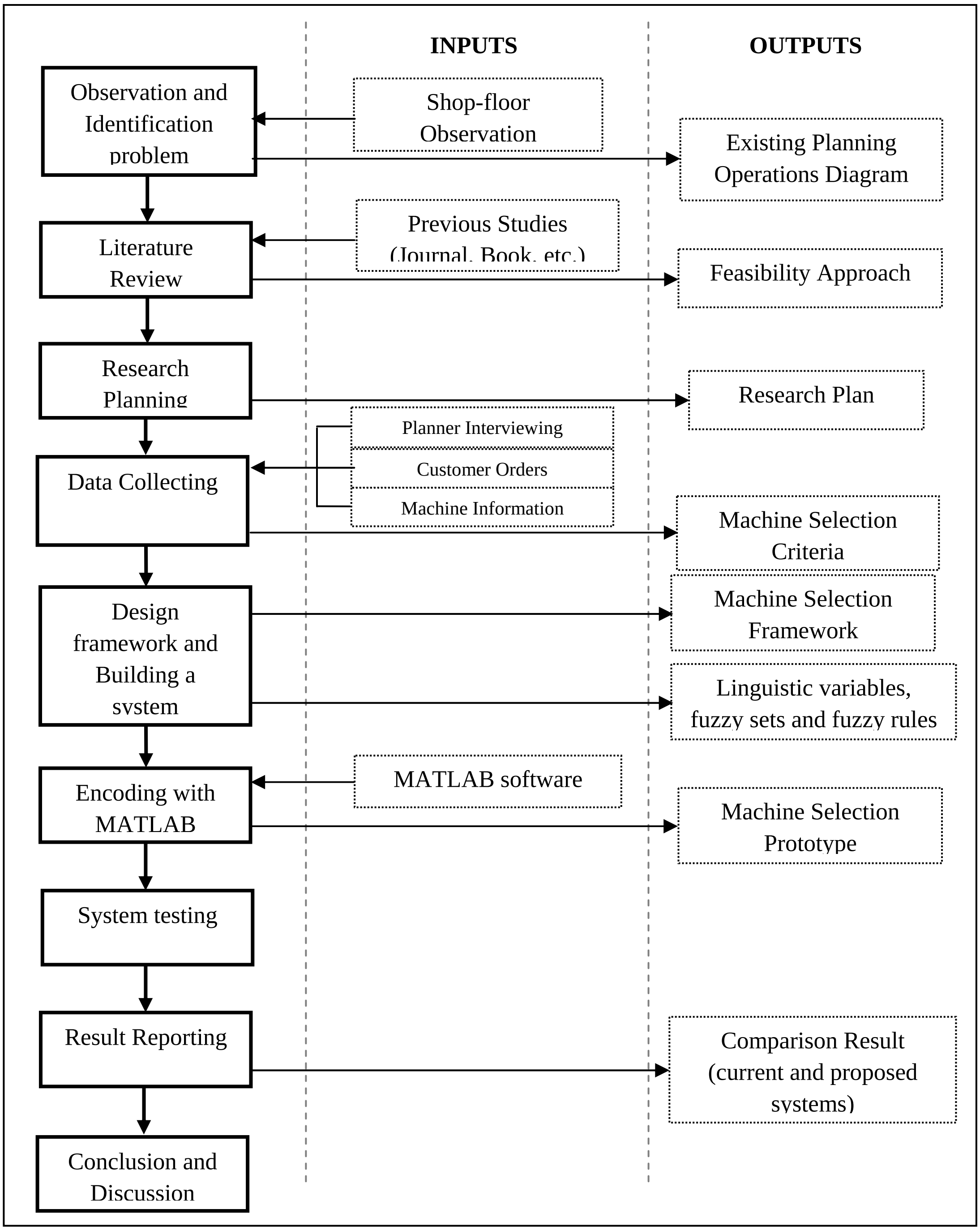


Figure 3.1 Overall research methodology flowchart

3.2.2 Data collection

Research used injection molding machines information from the case study of a plastic manufacturer in Pathumthani, a medium-size industry which always handles more than circulating 300 products that need planner's decision to allocate among 35 machines every month depending on orders from 50 customers above. Moreover, a number of urgent orders always induce additional decision before new schedule's arrangement. The interview was executed to a production planner to comprehend the selection logic idea of each resource allocation cycle. This research figures out which machine or job attributes will be required. Then, all influent machine and job parameters during machine selection phase were obtained and use as antecedents of rules in fuzzy logic system along this study. In order to avoid the excessive information and complicated system, eleven injection machines were selected for developing the program. They have increasing parameters familiarly trend. Table 3.1 shows the machine's information of sample machines.

Table 3.1 Machine information from the case study of a plastic manufacturer

Machine number	Machine Name	Clamping force (Ton)	Distance between tie bars X (mm)	Distance between tie bars Y (mm)	Shot weight (g)
1	TOSHIBA IS30EPN-1Y	30	260	260	28
2	JSW J30 ED	30	260	260	28
3	JSW J50 EII	50	310	310	56
4	NIGATA NN50E	50	310	300	60
5	TOSHIBA IS50EP-1.5A	50	310	310	56
6	NIIGATA	50	300	300	45
7	NIIGATA	50	300	300	45
8	NIIGATA	50	300	300	45
9	JSW J55 EII	55	310	310	23
10	NIIGATA	75	350	380	100
11	NIIGATA	75	350	380	100

The work flow of a planner's machine selection decision is analyzed to design a new automated machine selection system. The linguistic variables and their ranges are assigned, fuzzy sets are determined in universe of discourse and membership values to each parameter are defined.

3.2.2.1 Traditional injection molding machine selection approach

The medium-size plastic manufacturer has 35 injection molding machines which have been operating for more 50 of 300 products every month. They received customer orders, include product specification and then machine information to construct production plan and scheduling,

Overall business process is composed of information sharing network for production planning process has showed with the Figure 3.2. Purchase order are initially sent from customer to a planner by sales and a job will be matched with a suitable machine by using machines information database and working procedure from engineer department. Then, the planner expresses production scheduling.

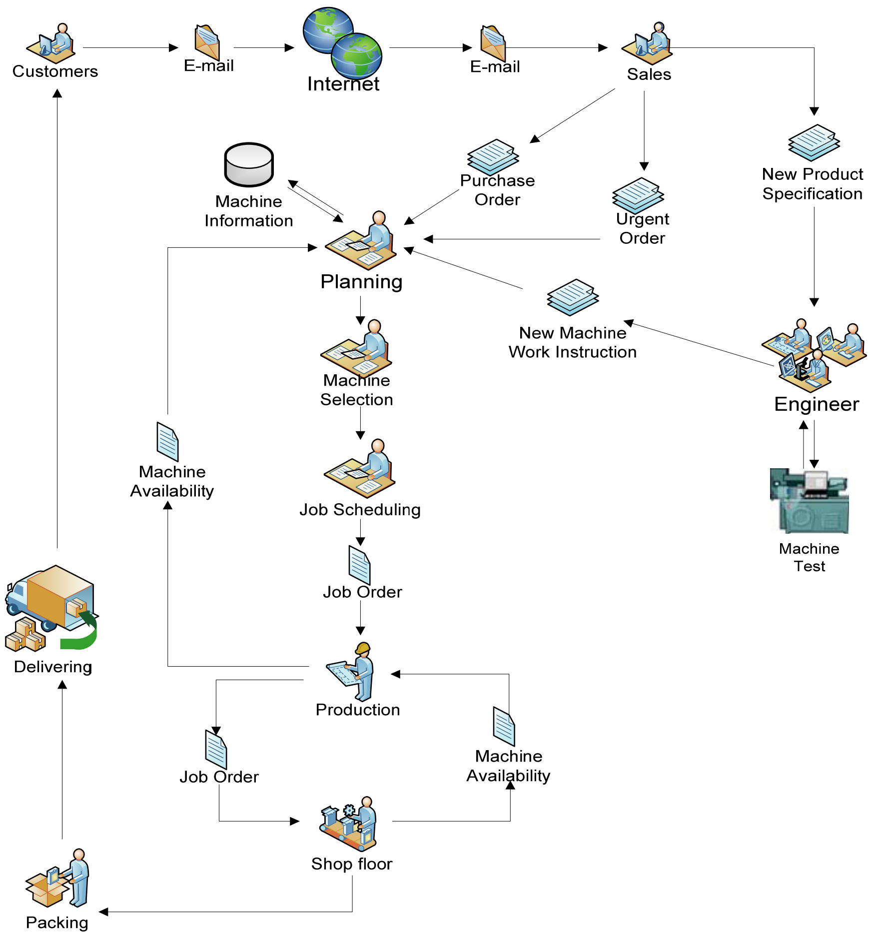


Figure 3.2 Current plastic production planning operations

3.2.2.2 Elapsed time of current machine selection process

The machine selection step will be started after a planner received orders from sales department. This process will be finished when the target machine number was identified. The total elapsed time of current machine selection process per a product is 147 seconds. Total selection time average is at 63.45 seconds. Table 3.2 shows the total elapsed time of the machine selection process for one product (one task).

Table 3.2 The total elapsed time of current machine selection process for one product

No.	Operations	Time (second)
1	Receive task description and check	20
2	Access the databases of product specifications	22
3	Compare task description and machine's specification and identify machine number. The machine's specification are included of: - Clamping force - Distance between tie bars - Shot weight - Product color of previous job - Material grade	105
Total elapsed time		147

3.2.2.3 Machine selection criteria

After all injection molding machine selection criteria were observed from the production planner. Main criteria are following into

- Clamping Force (Ton)
- Distance between Tie Bar (HxV)
- Shot Weight (g or oz.)
- Product Color
- Material grade

Figure 3.3 is showing how to select an injection molding machine by judgment with machine's specification, capacity and structure, and product requirements from the plastic injection molding industry.

3.2.2.4 User requirements

This study has collected necessary requirements from user to develop the system. All user requirements involving with the proposed system are here as follows:

1) Simple checks

2) User is planner

3) Develop user manual

4) The selection criteria are clamping force, distance between tie bar X, distance between tie bar Y, shot weight, product color and material grade

5) The system can handle the batch input

6) The system shows a notification box if any input is out of the range

7) The output is machine number

8) The system can be integrated with Microsoft Excel

9) Simple user interface

10) Develop checklists

11) Simple modify by user

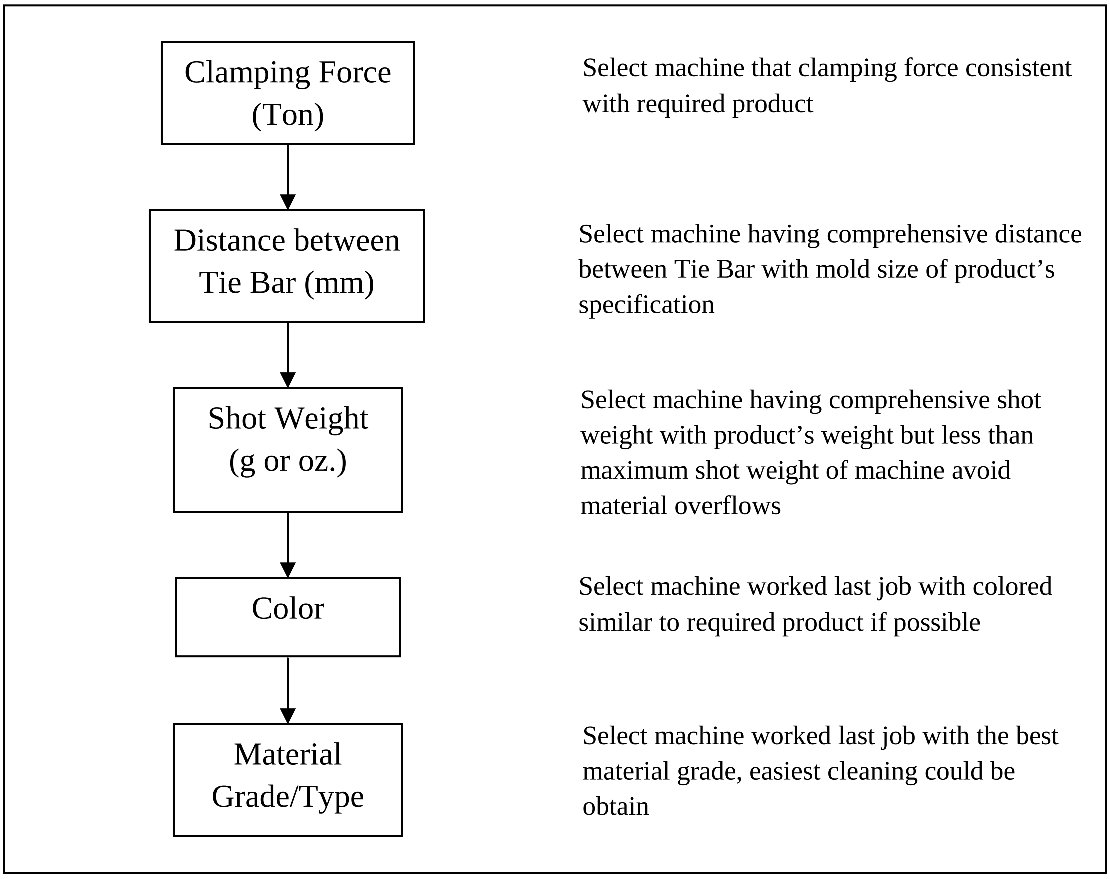


Figure 3.3 Injection molding machine selection process

3.2.3 Design framework and building a system

Regarding to integrate a new system to a current planning process, the new business workflow and overall system diagram will be designed.

3.2.3.1 New business workflow

After the analysis of work flow of planner's machine selection decision to design a new automated machine selection system, the current working process was improved concordantly with a new machine selection system. The new workflow introduced in Figure 3.4 and Figure 3.5 is a new business process diagram that integrated with the proposed machine selection system.

3.2.3.2 Overall system diagram

In the sections of MATLAB analyzing and machine database, the overall system diagram is designed in Figure 3.6. The system is composed of two subsystems, i.e. fuzzy inference system and system II.

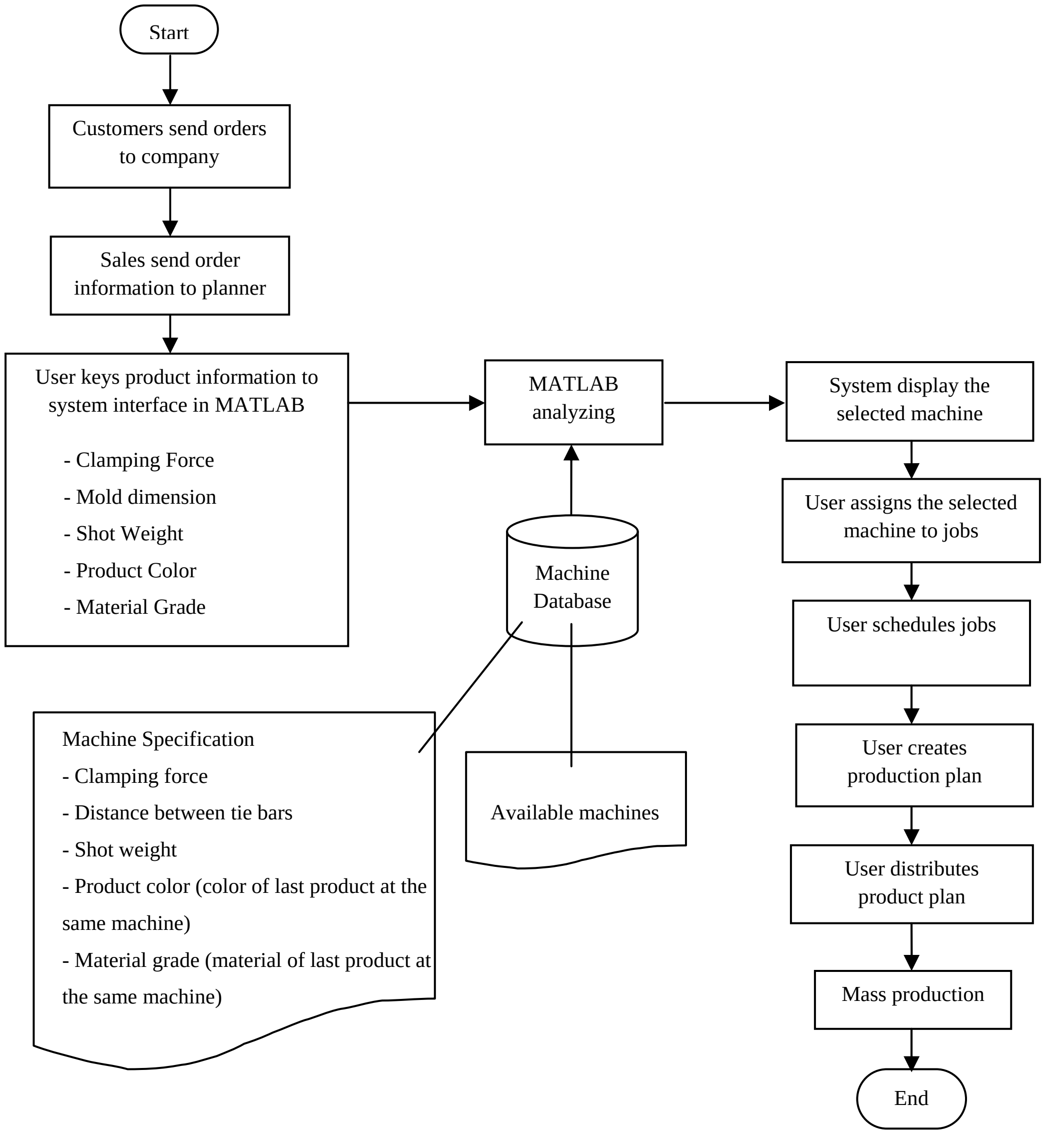


Figure 3.4 Machine selection framework

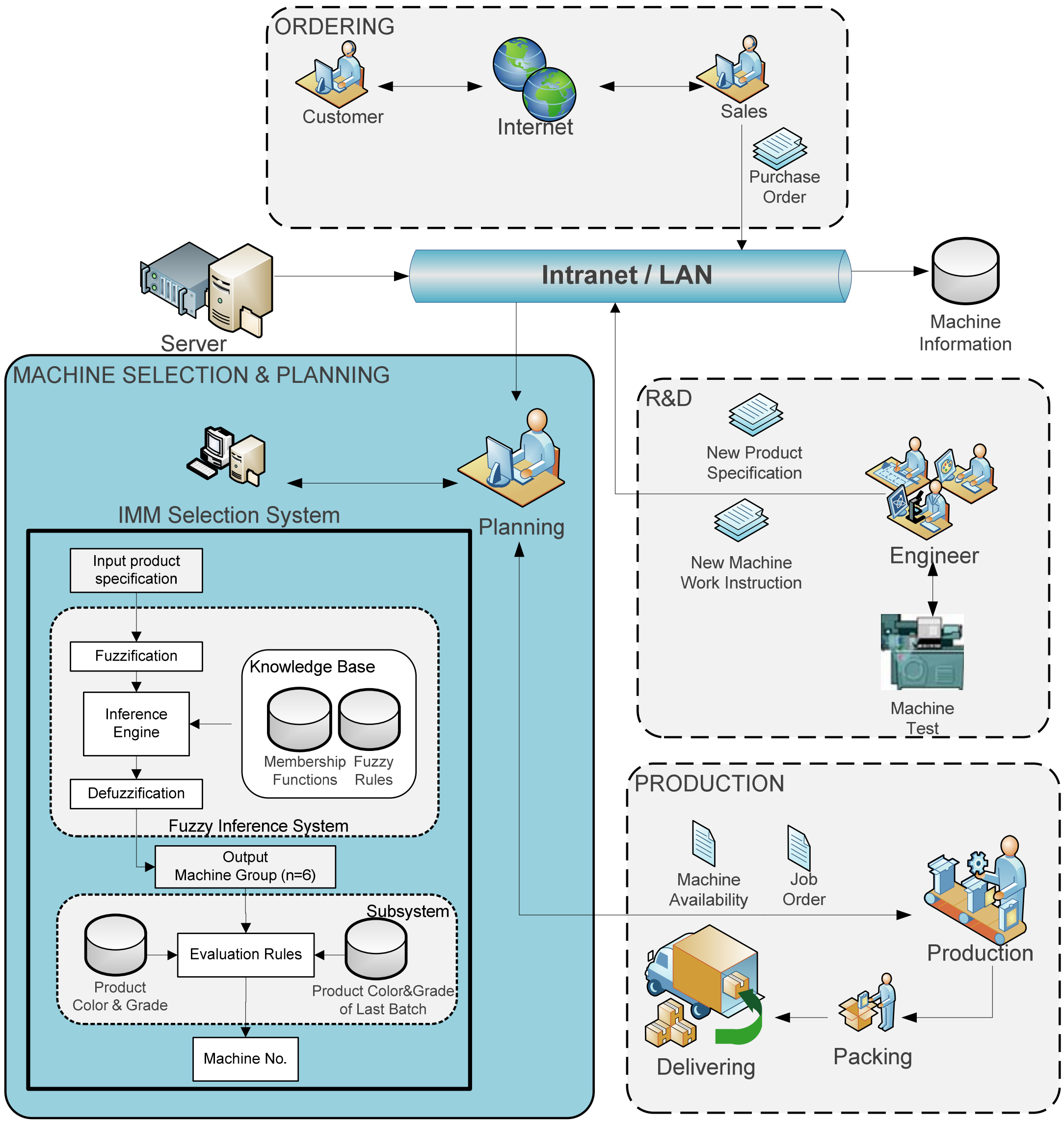


Figure 3.5 New business process diagram integrated with machine selection system

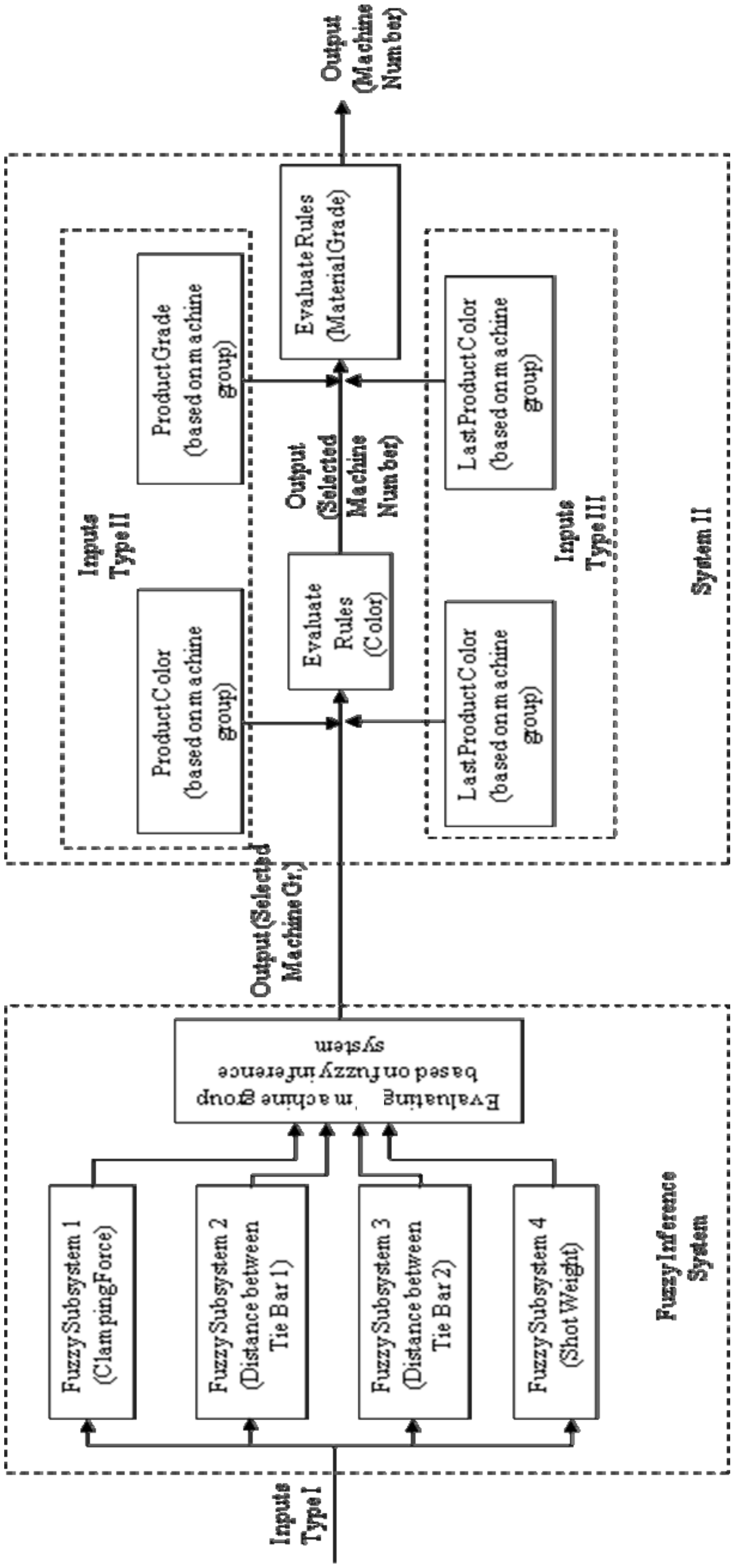


Figure 3.6 Overall system diagram

1) Fuzzy inference system: This system is used to match machine group to an order. The inputs of this section are product specification or task description which is consists of clamping force, distance between tie bars and shot weight. In other hand, output is the number of machine group (n) which are 1 to 6. The details of each group will be shown in Table 3.3.

The data for construct the fuzzy inference system are considered step by step as follows:

- Clamping force: This is the force between both sides of injection mold in the unit of ton. If an injection mold is provided with adequate clamping force, two sides of injection mold will be jointed completely without leakage. Thus the expected machine must not have clamping force lower than the product requirement. Figure 3.7 demonstrates a close step of forced injection mold by clamping force.

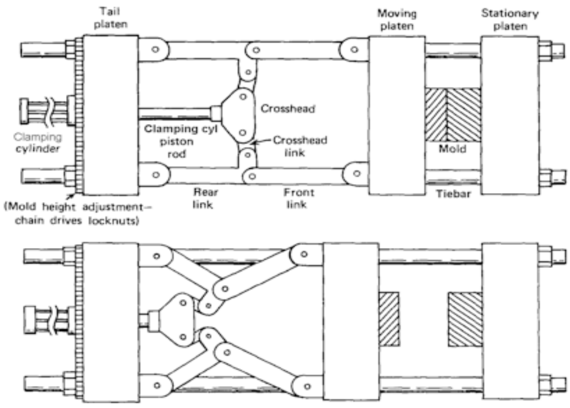


Figure 3.7 Close and open steps of forced injection mold by clamping force

Source: www.beejaymolding.com/Clamping-unit.html

- Distance between tie bar: The length between tie bars both vertical and horizontal axis. Tie bars help to maintain the right direction during closing - opening of injection mold. Any injection mold can be installed into a machine if its dimension does not exceed this value. In this research, the variable can be separated to be two fuzzy sets as follows:

- Distance between tie bar X: The length between of two tie bars in horizontal axis. From Figure 3.8, the distance between tie bar X is 1,520 mm.

- Distance between tie bar Y: The length between of two tie bars in vertical axis. From Figure 3.8, the distance between tie bar Y is 1,280 mm.

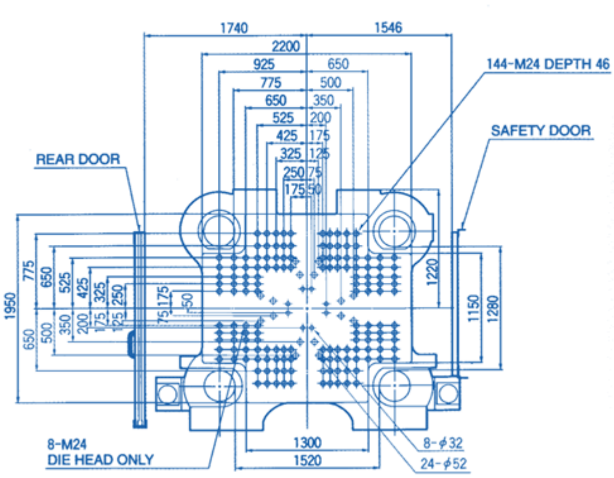


Figure 3.8 Distance between tie bars of the injection molding machine

Source: www.mhi-pt.co.jp/injec_e/products/MMV/1600MMVtool.htm

- Shot weight: The maximum weight of plastic is injected directly into injection mold per feeding. The product should have weight equal or lower than maximum shot weight of machine to avoid material overflow.

2) System II: This system operates to select the most suitable machine. The most suitable machine is the machine that used to produce the preceding job with similar material color with required job and also has ever produced the material with easiest cleaning. The output of this system is the machine number.

3.2.3.3 Fuzzy variable determination

In Figure 3.6, output of fuzzy inference system is a selected machine group. That means the 11 machines was classified by considering similarity of machine. The number of group of machines is represented by ‘n’. Thus, there are 6 groups of injection molding machine. The details of machine groups had given in Table 3.3.

Table 3.3 List of sample machines

Machine Group	Machine number	Machine Name	Clamping force (Ton)	Distance between tie bars X (mm)	Distance between tie bars Y (mm)	Shot weight (g)
1	M1	TOSHIBA IS30EPN-1Y	30	260	260	28
	M2	JSW J30 ED	30	260	260	28
2	M6	NIIGATA	50	300	300	45
	M7	NIIGATA	50	300	300	45
	M8	NIIGATA	50	300	300	45
3	M3	JSW J50 EII	50	310	310	56
	M5	TOSHIBA IS50EP-1.5A	50	310	310	56
4	M4	NIGATA NN50E	50	310	300	60
5	M9	JSW J55 EII	55	310	310	23
6	M10	NIIGATA	75	350	380	100
	M11	NIIGATA	75	350	380	100

3.2.3.4 Define linguistic variables

This part is the development of our system by mainly determining the linguistic variables and their ranges, fuzzy sets in universe of discourse and membership values to each parameter. All variables were classified in range by representing with linguistic term (Table 3.4) and then were transformed to the degree of membership by data changing phenomena.

For example, if clamping force is 30, then the linguistic term is ‘Very low’ then the degree of membership range is 1.0. The objective here is to advise a user to produce any product on the machine group 1 if it requires 0 - 30 tons of clamping force. That means the degree of membership in ‘Very low’ set is 100%. These products should not be processed on the machines that have more than 30 tons of highest clamping force. That means the degree of membership in ‘Low’ set is 0%. And if the clamping force changed to be 31, then the degree of membership range in ‘Very low’ is 0. As a result, the degree of membership range in ‘Low’ will be 1.0. That also means products prefer to be processed on the machines that have more than

30 tons of highest clamping force. That means the degree of membership in ‘Low’ set is 100% and the degree of membership in ‘Very low’ set is 0%.

Table 3.4 Linguistic variables and their ranges

Linguistic variable	Range	Linguistic value
Clamping Force (tons)	0 -30	Very low
	31 - 50	Low
	51- 55	Medium
	56 – 80	High
Distance between Tie Bar X (mm)	0 – 260	Very low
	261- 300	Low
	301 – 310	Medium
	311 – 350	High
Distance between Tie Bar Y (mm)	0 – 260	Very low
	261- 300	Low
	301 – 310	Medium
	311 – 380	High
Shot Weight (g.)	0 – 22	Very very low
	23-27	Very low
	28-44	Low
	45 – 55	Medium
	56 - 59	High
	60 – 105	Very high
Machine group	0 – 1	G1
	1 – 2	G2
	2 – 3	G3
	3 – 4	G4
	4 – 5	G5
	5 – 6	G6

3.2.3.5 Fuzzy sets determination

Fuzzy sets can have a variety of shapes. However, a triangular or a trapezoid often satisfies the expert knowledge and simplifies the process of computation significantly. Figure 3.9 shows the fuzzy sets for all linguistic variables used in this fuzzy inference system that are clamping force, distance between tie bar X, distance between tie bar Y, shot weight and machine group.

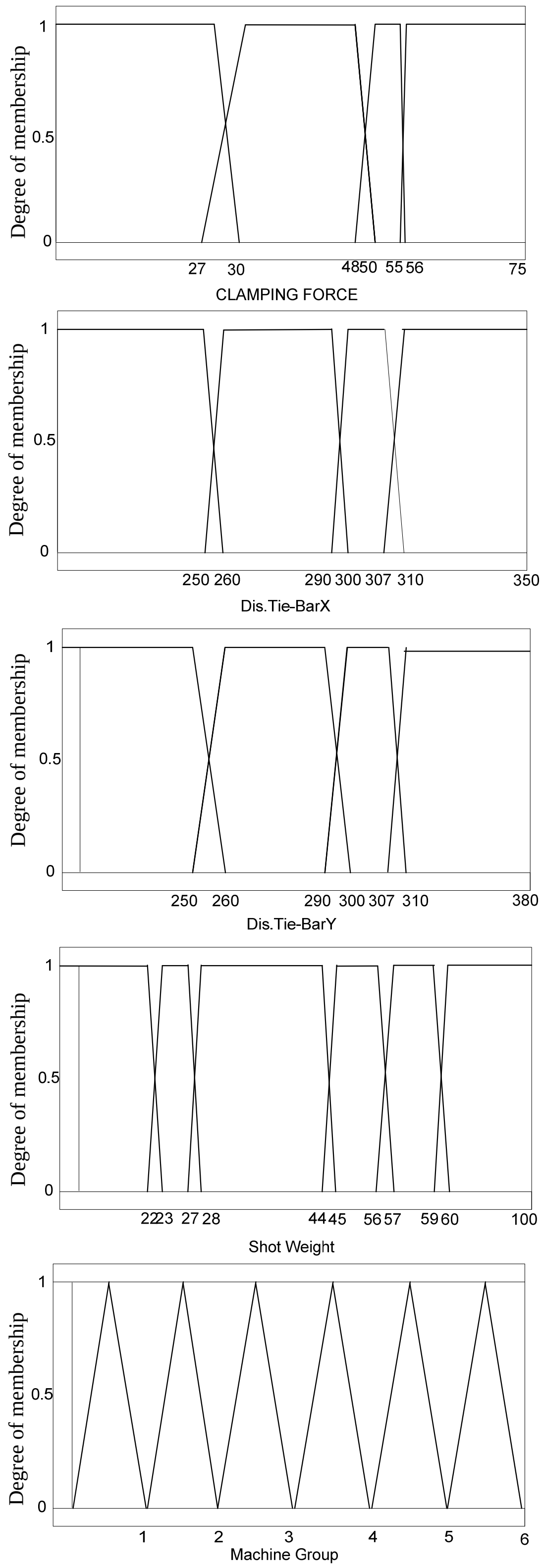


Figure 3.9 Fuzzy sets of 4 inputs and 1 output variables

3.2.3.6 Fuzzy rules construction

After fuzzy sets of input and output variables had illustrated, then the 384 rules were constructed and added into this system. The output of the fuzzy inference system in this problem is machine group. There are n groups of machine, but in this study group of machine can be classified to six machine group corresponded to machine specifications (n=6), these are classified by the similar machine specification. The fuzzy inference system diagram is illustrated in Figure 3.10.

The number of rules
= No. of clamping force's fuzzy sets ×
No. of distance between tie bar X's fuzzy sets ×
No. of distance between tie bar Y's fuzzy sets ×
No. of shot weight's fuzzy sets
= 4 × 4 × 4 × 6 = 384 rules

In this problem, the 'AND' operator was applied to construct all fuzzy rules. It provided the corrected results. These fuzzy rules will have multiple antecedents, for example:

IF	ClampingForce is Verylow
AND	TiebarX is Verylow
AND	TiebarY is Verylow
AND	Shotwt is Veryverylow
THEN	MCgroup is G1

Or,

1. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G1)
2. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G1)
3. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G1)
4. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G1)

The structure of all fuzzy rules has designed as in Figure 3.10. This diagram consists of four inputs which could generate 384 fuzzy rules. The result of each rule will be calculated via aggregation of rule consequent process. Finally, the aggregated results will be transformed by defuzzification process to be crisp output or machine group as in this case.

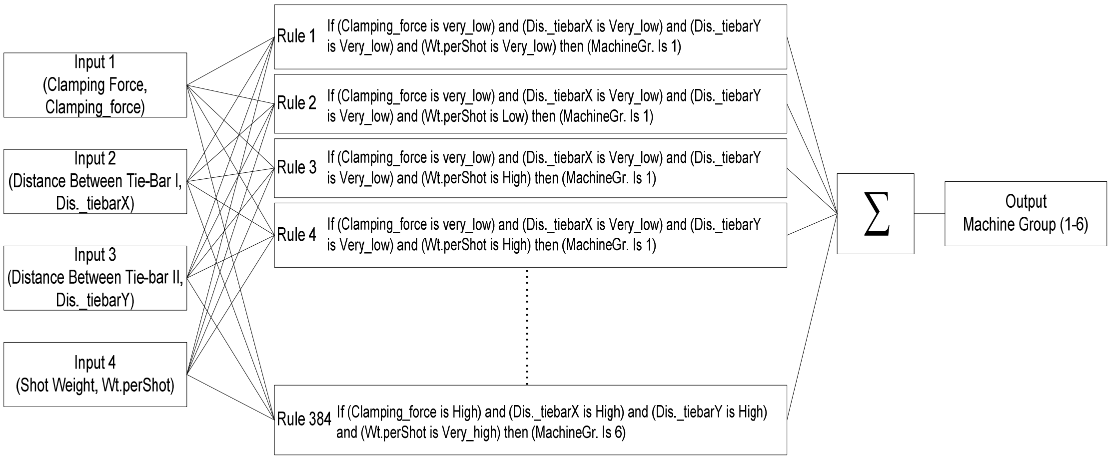


Figure 3.10 Fuzzy Inference System Diagram

3.2.3.7 System II development

In according to identify which machine is the most suitable for a job, the output of fuzzy inference system (machine group) has to be completely evaluated by other criteria that are product color and material grade. The decision flow chart of selection by these criteria is shown in Figure 3.11.

- Product color: This attribute is color of product that we want to produce. It will be matched with any machine that is used to produce the same color in the latest batch. For example, if product color is white, the product color of the latest batch of selected machine should be white also or nearly clear color (colorless). If a machine group processes in the same color in more than one machine, the material grade will be considered. In this study, there are four product colors, i.e. clear, white, grey and black.
- Material grade: This attribute refers to the complication of cleaning process and raises more risk of contamination from material burning of each material grade. If the material grade of a job is A, the last batch of a suitable machine should have produced in the same grade or nearly. The four grades of material were

assumed by four symbols that are A, B, C and D. If material grades is sort by the ability of cleaning, the A > B > C > D is achieved. That means A is the easiest cleaning material and D is the hardest one.

For example,

If a product 'A' has judged to processed on machine group 1 (n = 1) and the product specifications are 25 tons of clamping force, mold dimension is 280x280 mm, weight is 40.5 g, clear color and material is B grade. Thus, the output of fuzzy inference system is G2 because it larger than the range of G1 (260x260 mm). That means this product should be processed on the machine group 2. This group consists of three machines, i.e. M6, M7 and M8. They had produced clear, clear and black products; and material grade of them are B, A and B respectively. Therefore, M6 and M7 are feasible to process this order by considering of product color. Finally, M7 had processed with the easiest cleaning material thus the selected machine is M7.

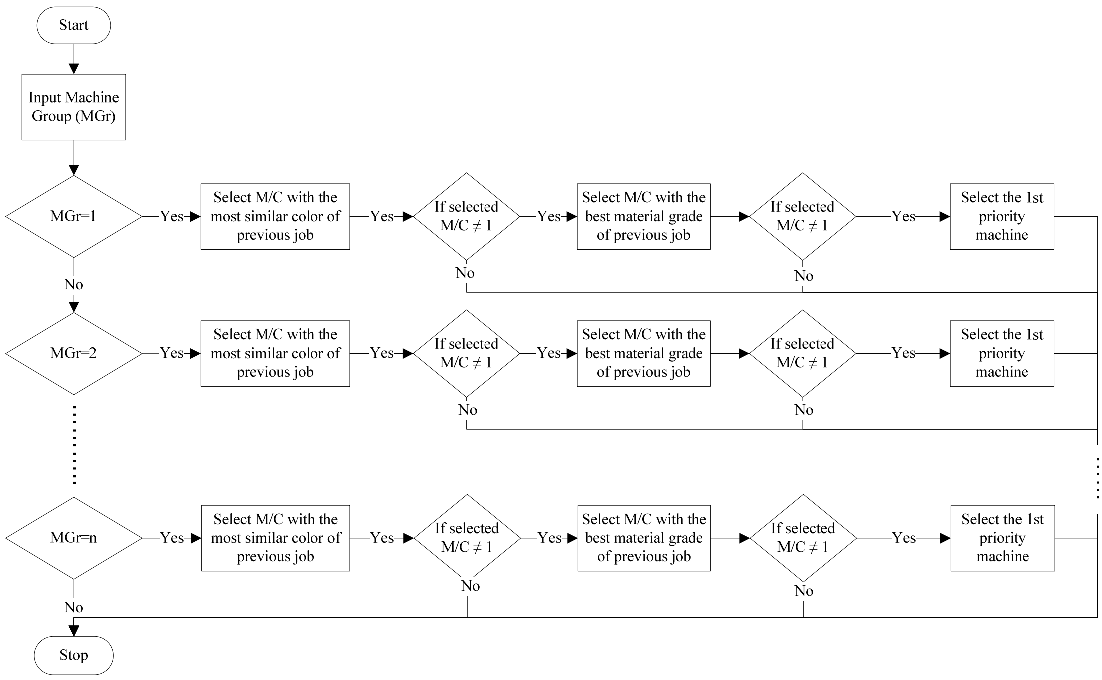


Figure 3.11 Decision flowchart of Product color and Material grade variables (n = 6)

3.2.4 Encoding with MATLAB

This step is included of encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference using MATLAB. Within this system, interval variables will be normalized to being in range of [0,1] and then will be carried on fuzzy inference process with Mamdani-style. Finally, the applicable system is enhanced via Graphic User Interface using MATLAB tools.

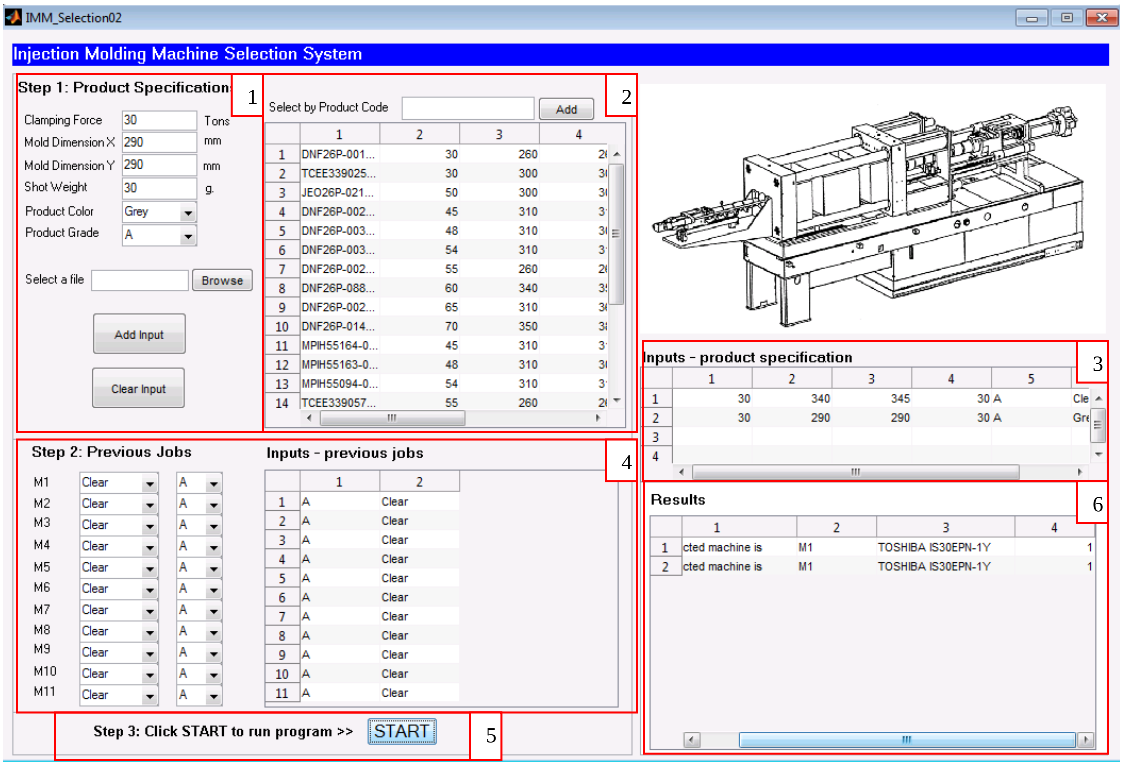


Figure 3.12 User interface of prototype system

Figure 3.12 shows the proposed system which is composed of four sections as follows:

1) Variable inputs – This section is composed of:

a. Product Specifications: A user needs to fill up with product specification or upload a data file (*.xlsx format only) before click ‘Add’ to feed data to system database or ‘Reset’ to clear all data.

b. Previous Jobs: A user can choose the historical data of each machine’s previous jobs with drop-down list of both Product color and Material grade. And then click ‘Submit’ to execute the system.

2) Variable input display – This section is composed of:

a. Inputs - Product specification: This section shows input variables, i.e. clamping force, distance of tie-bar X, distance of tie-bar Y, shot weight, product color and material grade, that were uploaded.

b. Inputs - Previous Jobs: This section shows input variables, i.e. material grade and product color of previous job that were uploaded.

3) Result display – In this section, the selected machine will be presented and details will be given in the columns of machine number, machine name and machine group and exemplified in Figure 3.13.

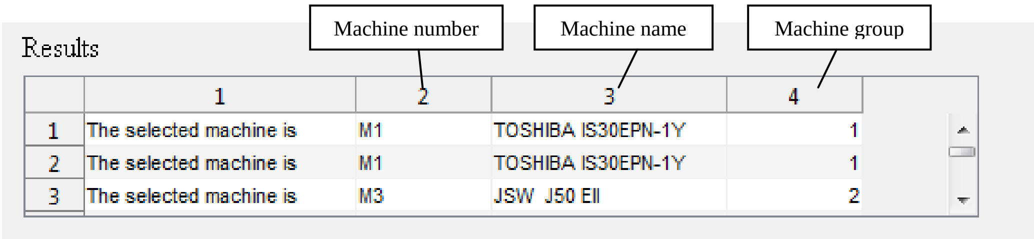


Figure 3.13 Example results that generated from proposed system.

4) Elapsed time – In this section, the computational time of system will be counted.

3.2.5 System testing

Crisp values of all fuzzy intervals are input into system and will be tested in order to ensure smoothly and accurately output results. Then, the computational time is compared to versus human performance. The data for testing are as follows:

- Clamping Force (Ton)
- Mold dimension (mm): The dimension of injection mold will be compared with the length between tie bars both vertical and horizontal axis. Any injection mold can be installed into a machine if its dimension does not exceed that machine's distance between tie bars.

- Mold dimension X: The length mold in horizontal axis. From Figure 3.14, mold dimension X is represented by 'B'.

- Mold dimension Y: The length of mold in vertical axis. Figure 3.15, mold dimension X is represented by 'D'.

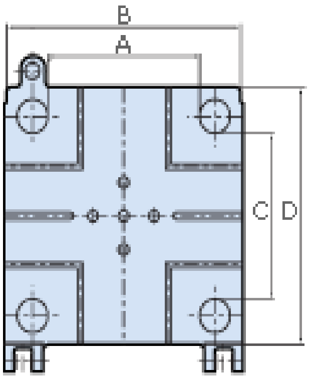


Figure 3.14 Injection mold

- Shot Weight (g or oz.)
- Product Color
- Material grade

3.2.6 Results, conclusion and discussion

The advantage and disadvantage among two systems of proposed system and old system will be represented along with result reporting. Then, the shortcoming in system for improvement in future will be discussion.

3.3 Research tools

The research tools are composed of hardware and software as follows:

3.3.1 Hardware

- Intel(R) Core™ 2 Duo P8400 @ 2.26 GHz
- 3.00 GB of RAM
- 100 GB of Hard Disk
- NVIDIA 9200M GS

3.3.2 Software

- Microsoft Windows 7 with 64-bit Operating System
- MATLAB R2010a

- Microsoft Office Visio 2007
- Microsoft Office Word 2007

CHAPTER IV
SYSTEM SCRIPTS AND DESCRIPTIONS

This chapter reports the comparison of traditional method and proposed system performances. The proposed system was the consequence which had accurate performance and closed to human being's decision. This system could decrease computational time and risks of human error from a current method.

4.1 System development

This section consists of the system development paradigm step by step and the compositions of interface and codes. The designed membership functions are used to construct our model. All features and layout are designed to support a user with user-friendly operation steps.

4.1.1 The process of system construction

The process of injection molding machine selection system can be categorized into two major steps. There are the process of Fuzzy Inference System development and System II development. In the initial phase, Fuzzy Inference System has been created owing to be the first evaluation step to getting the most suitable machine group. And the System II is used to account the most appropriated machine within the selected machine group by previous system. The following sections purpose to reveal all approaches for the proposed system development.

4.1.1.1 Fuzzy inference system

1) The access method of MATLAB is started by clicking of MATLAB R2010a in Start Menu in Figure 4.1.

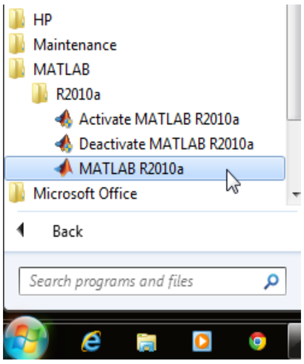


Figure 4.1 Opening of MATLAB program

2) The default desktop of MATLAB will display in Figure 4.2. The layout of the desktop is composed of Current Folder, Editor, Command Window, Workspace and Command History.

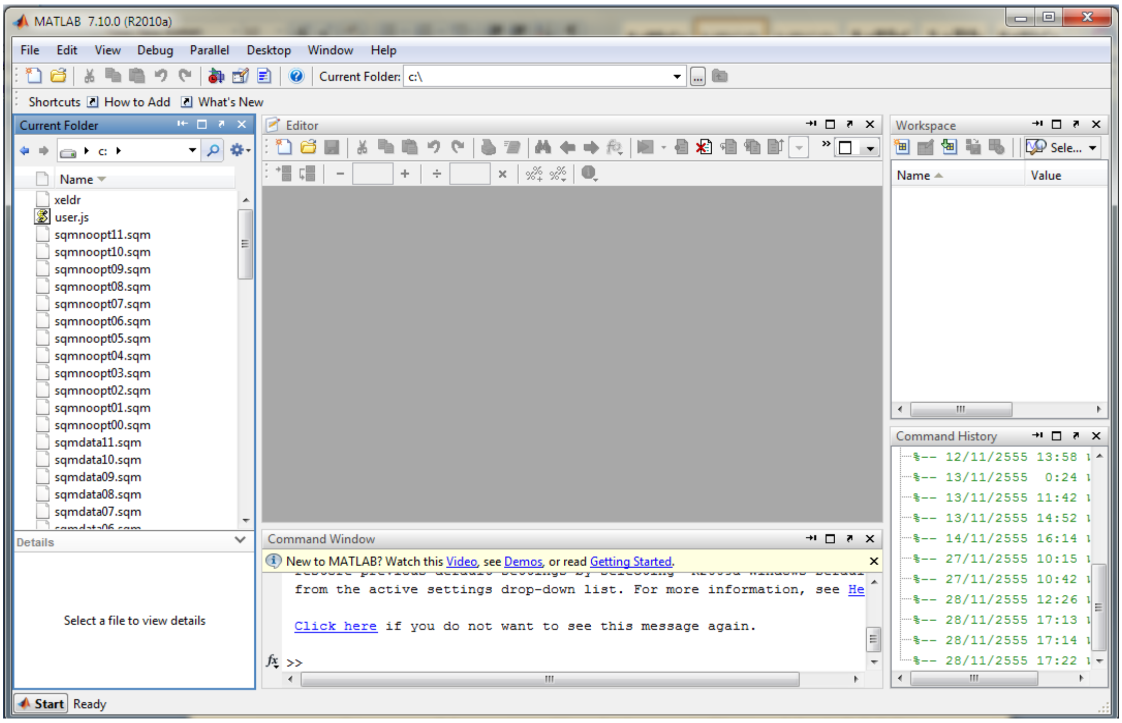


Figure 4.2 Default desktop of MATLAB 7.10.0 (R2010a)

3) Set the Current Folder as the directory that will be used to keep all files concerning the injection molding machine selection system. For this research, those files had saved in E:\THESIS\Fuzzy System01 as Figure 4.3. Please be reminded to set the same directory for the same project in every time of starting MATLAB.

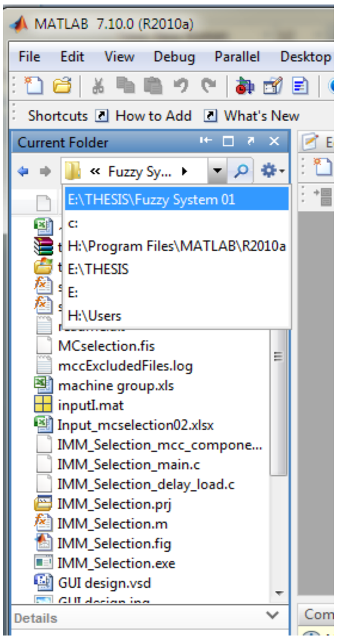


Figure 4.3 Setting the current folder in MATLAB

4) Type ‘fuzzy’ at command window (Figure 4.4) to open the Fuzzy Logic Toolbox and then press ‘Enter’.

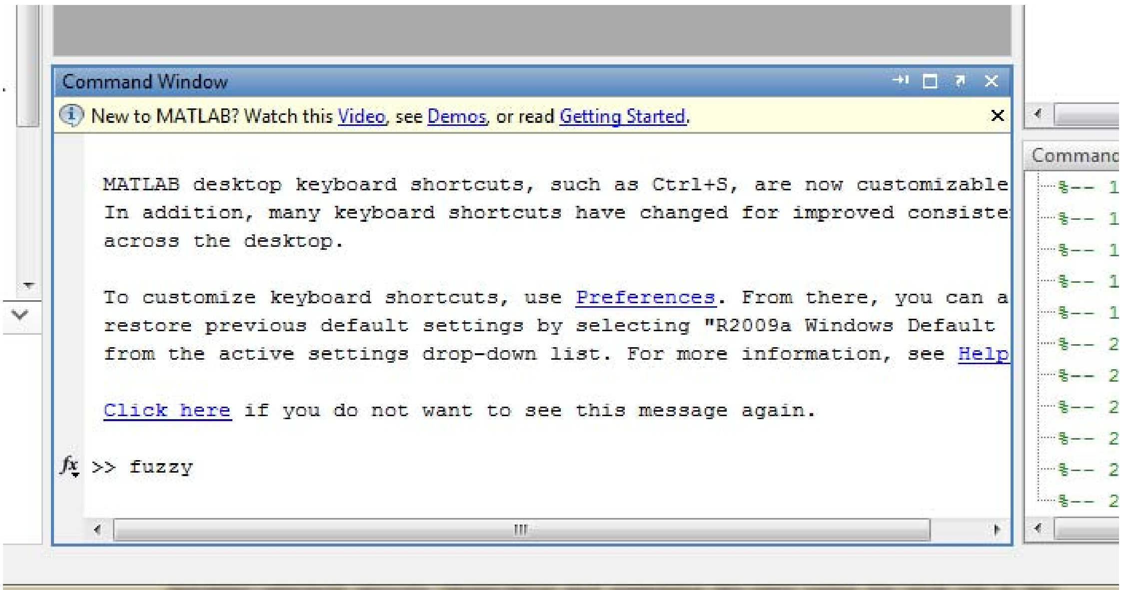


Figure 4.4 The access method of fuzzy logic toolbox

5) The program will display the window of fuzzy inference system editor in Figure 4.5.

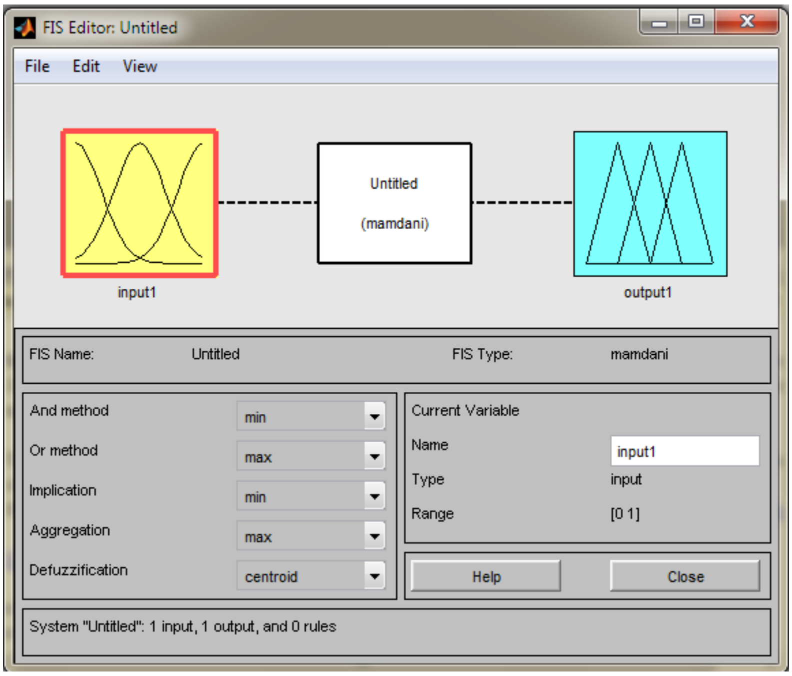


Figure 4.5 Fuzzy logic toolbox user Interface

6) Define name for FIS variable as 'ClampingForce'. Do not leave a space in the name of FIS variable. Set the function of And method, Or method, Implication, Aggregation and Defuzzification by following in Figure 4.6.

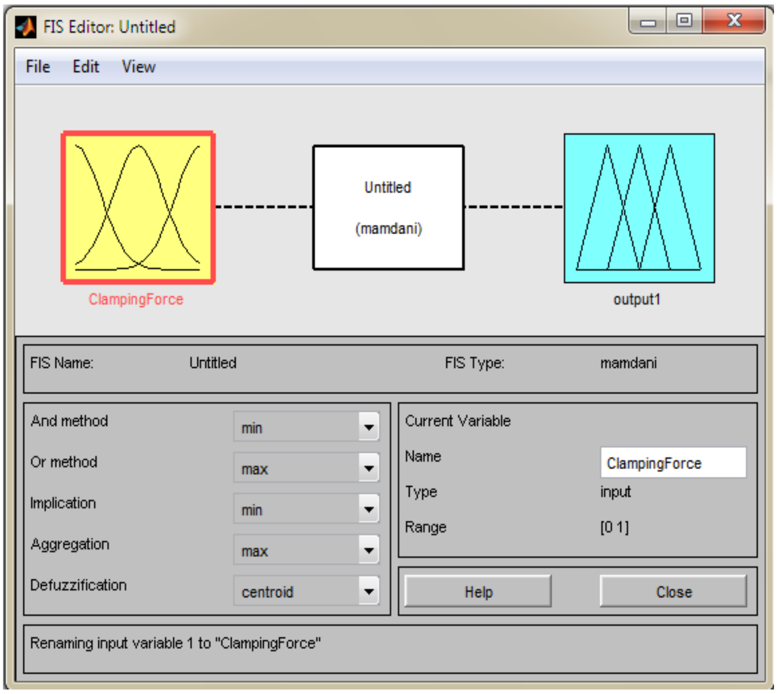


Figure 4.6 Adding name of FIS variable

7) Add new FIS variables by clicking Edit > Add Variable… > Input (Figure 4.7) and define name to them same as 6).

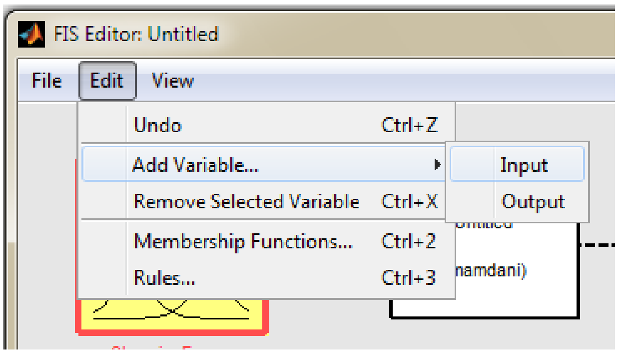


Figure 4.7 Adding new FIS variable

Figure 4.8 shows the completed variables both inputs and output.

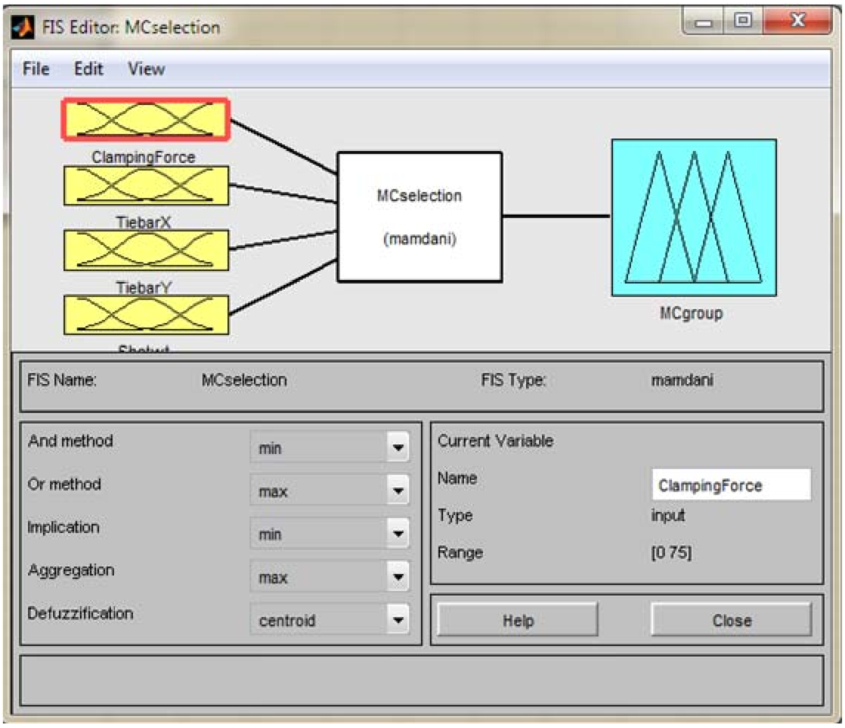


Figure 4.8 Overall FIS variables

8) Define the ranges of membership functions (MFs) for every variable by clicking Edit > Membership Functions… or double click on the graph of ClampingForce membership function. Later, define Type, Number, Params, Range and

Display Range of every MFs by following the ranges and linguistic variables of Table 3.5. Select the Type as Trapmf (Trapezoidal Membership Function) in order to satisfy the transformation of degree of membership. For example;

Add membership function of ClampingForce by clicking Edit > Add MFs…

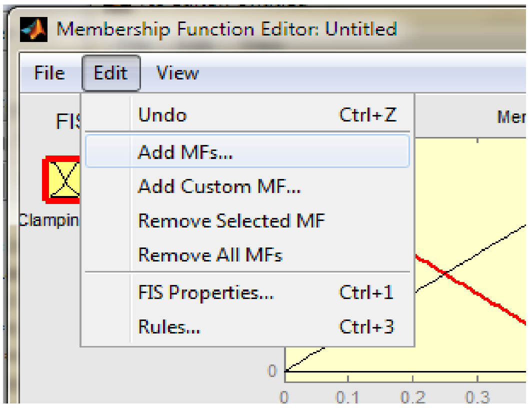


Figure 4.9 Adding MFs

Select AF type as ‘trapmf’ and Number of MFs is 4 then click ‘OK’ (Figure 4.10). Later, MATLAB will compose the four MF lines over existing lines (Figure 4.11).

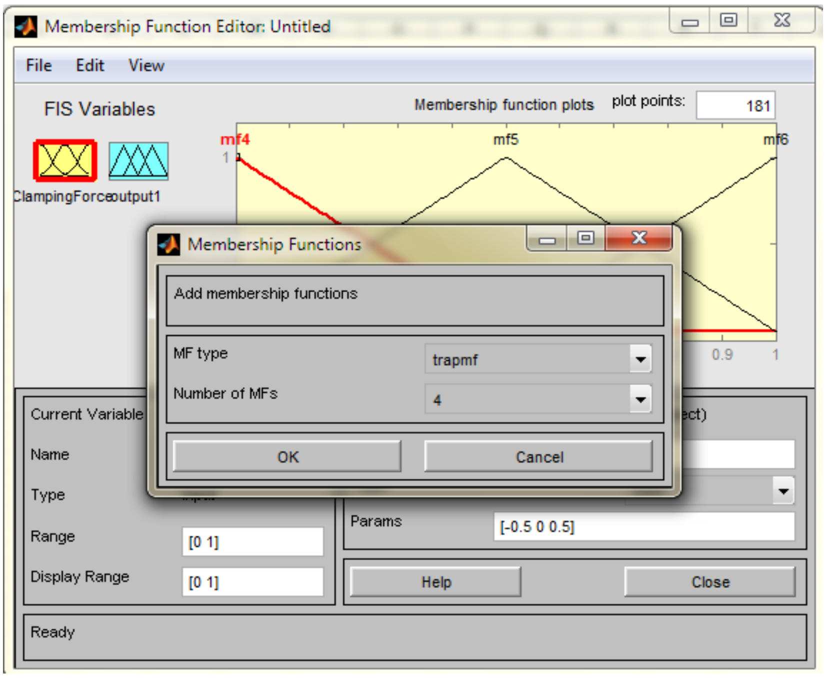


Figure 4.10 Setting MF type and number of MFs

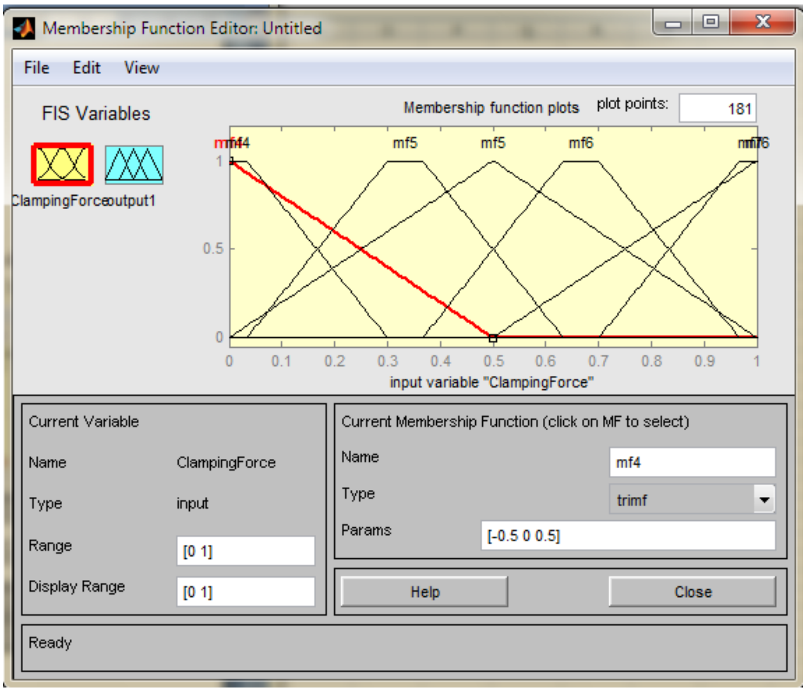


Figure 4.11 Membership function lines after add more new lines

Delete three old lines and define Range (as in Figure 4.12), Display Range, Name and Params as [-26 47 0 30 31]. Press 'Enter'. Do this approach to other input and output variables (i.e.TiebarX, TiebarY, Shotwt, and MCgroup).

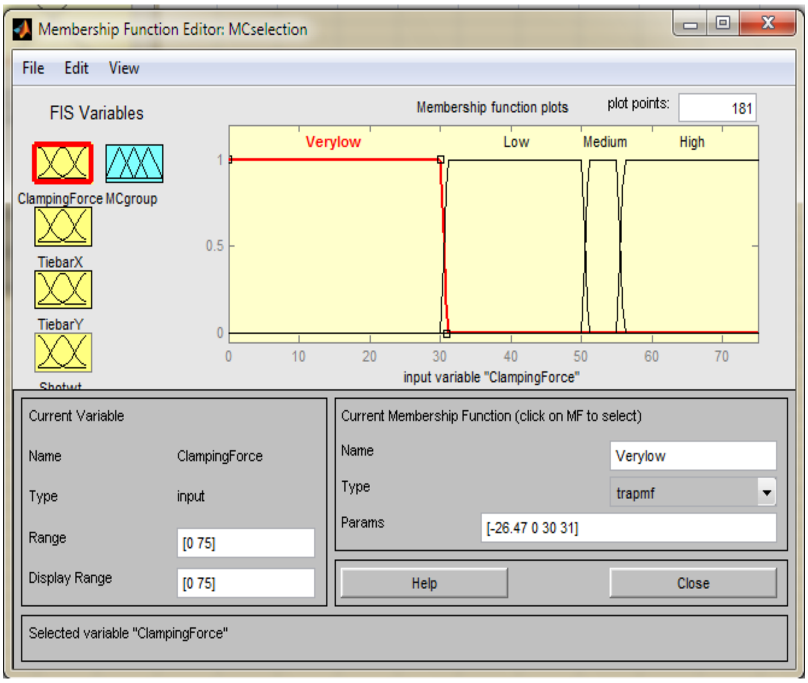


Figure 4.12 Membership function lines after modifying as trapezoidal and defined linguistic names

9) Develop fuzzy rules by click Edit. Choose Rules as in Figure 4.13.

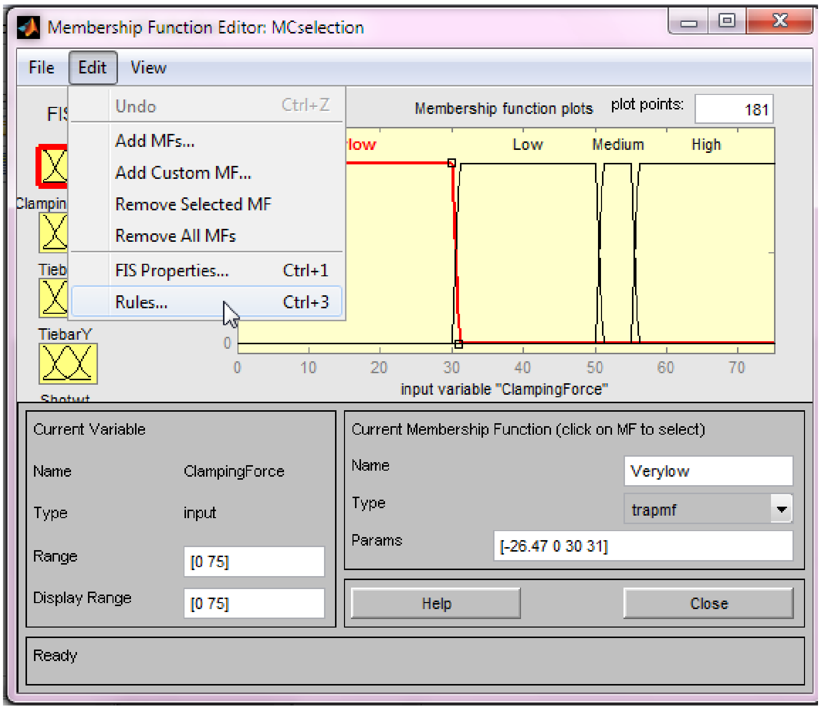


Figure 4.13 Rules menu to construct FIS rules

Add rules by selecting MFs based on our machine selection logical and then press 'Add rule' (Figure 4.14). The overall required rules is 384 rules.

For example:

1) If ClampingForce is Verylow and TiebarX is Verylow and TiebarY is Verylow and Shotwt is Veryverylow Then MCgroup is G1

Means If ClampingForce is 0 – 30 tons and TiebarX is in 0 – 260 mm and TiebarY is in 0 – 260 mm and Shotwt is in 0 – 22 mm then the MCgroup of G1 will be selected.

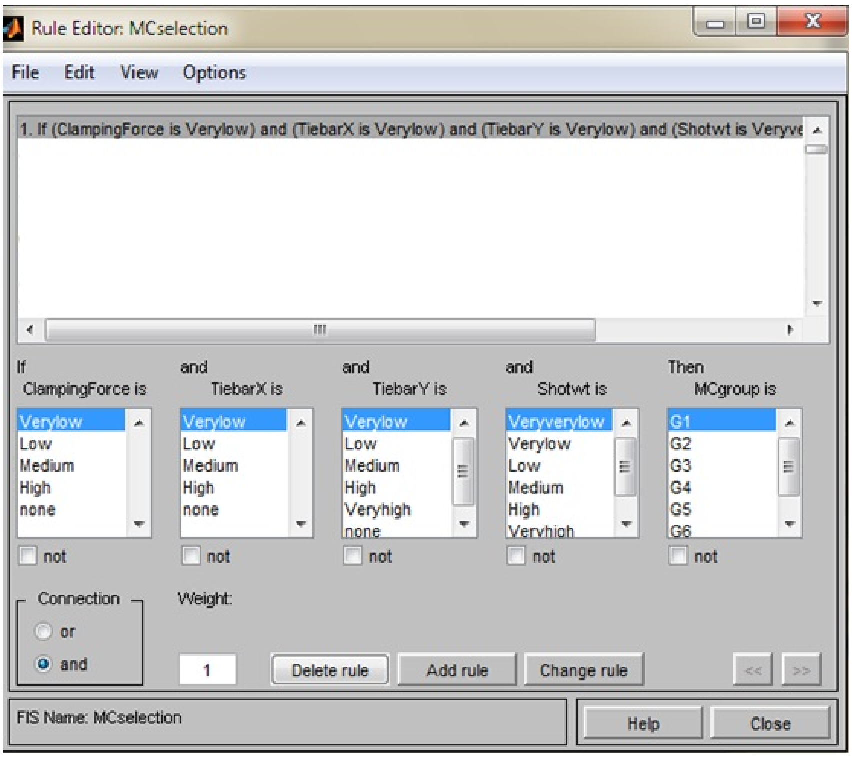


Figure 4.14 Rule editor window

10) Save the new fuzzy inference system by exporting as a .fis file in directory of Current Folder. In this case, we named it as MCselection.fis. This file will appear in the Current Folder window immediately. Please see Figure 4.15.

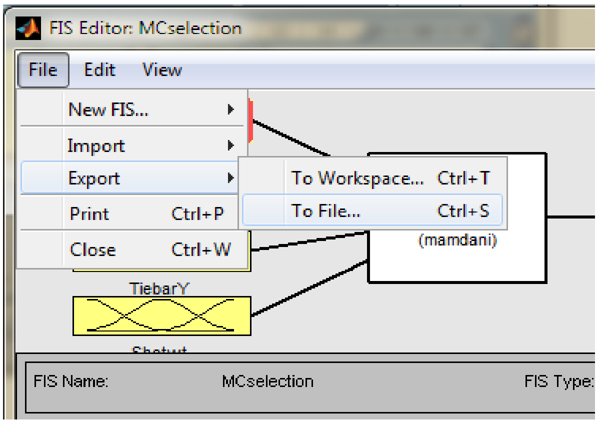


Figure 4.15 Saving a FIS project

4.1.2 Fuzzy inference system editor

The fuzzy inference system can be modified in detail via editor window. The MATLAB Editor window displays codes for making the system. Two or more systems can be merged together by inserting codes into a multiple file (M-file) and run automatically. Figure 4.16 shows MATLAB Editor with including ‘MCselection.fis’ codes. For the details of using M-file, this research will describe in the graphical user interface (GUI) development section.

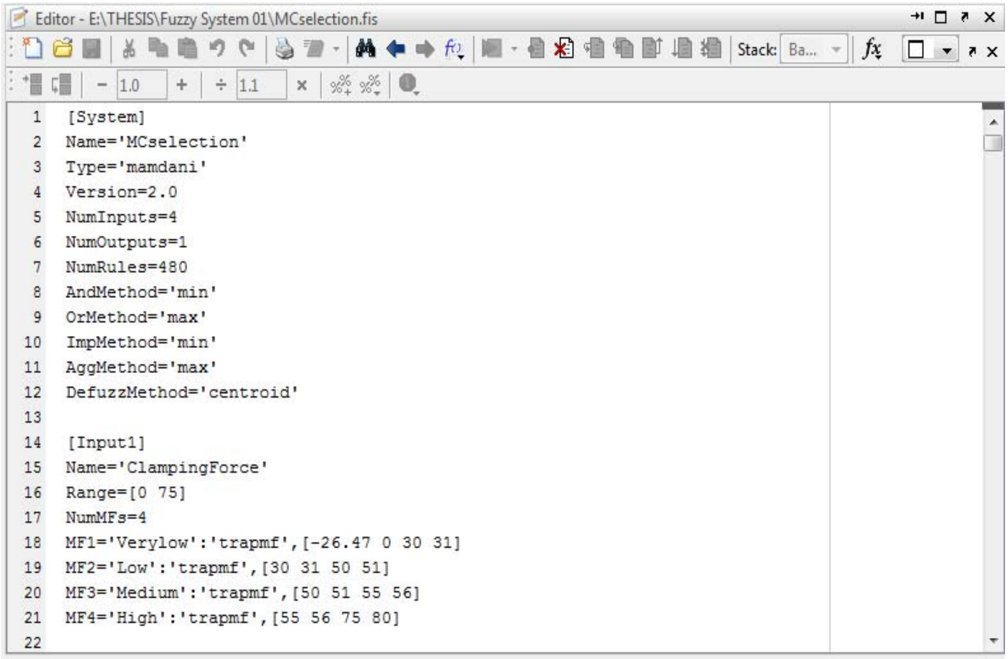


Figure 4.16 Fuzzy inference system that has opened in MATLAB editor

4.1.3 Graphical user interface development approach

The graphical user interface or GUI file is used to be the operation interface. This important stage stands for our proposed system because GUI contains MATLAB functions that control the operation of GUI. The processes of creating GUI are the laying out and programming GUI. The GUIDE Layout Editor is used to design the layout of GUI. MATLAB Editor is used to program the callbacks to perform the actions you want the GUI to perform. The methods of GUIs construction have stated in this section.

4.1.3.1 GUI Laying out

1) Select File > New > GUI as Figure 4.17. Or type ‘guide’ in Command Editor to create new GUI file.

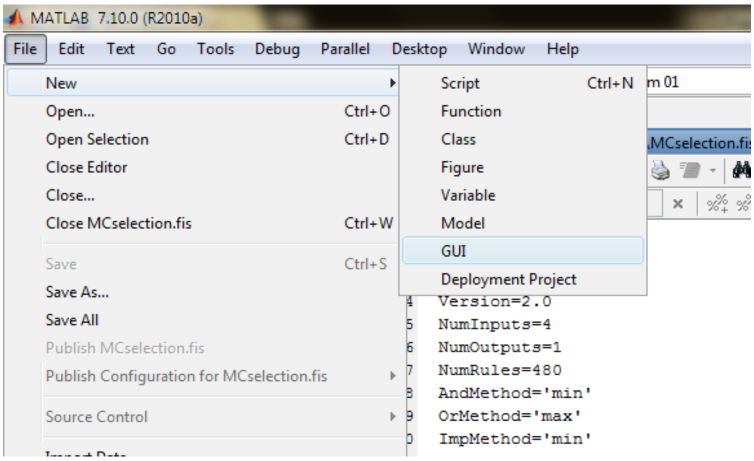


Figure 4.17 Creating GUI file

2) Within the GUIDE Quick Start window as in Figure 4.18, select ‘Create New GUI’ tab for the ‘Blank GUI (Default)’ templates and press ‘OK’.

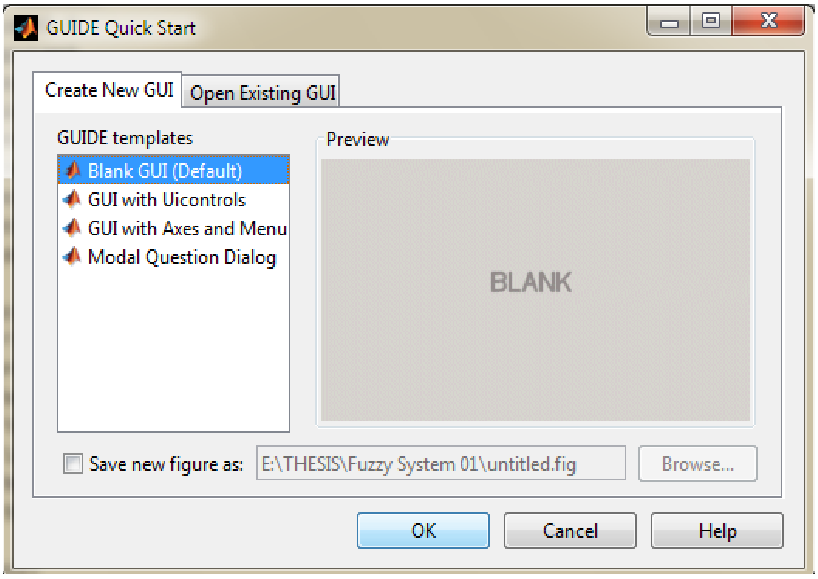


Figure 4.18 GUIDE Quick Start window

3) MATLAB will open the main window of GUIDE Layout Editor for laying out new GUI in Figure 4.19.

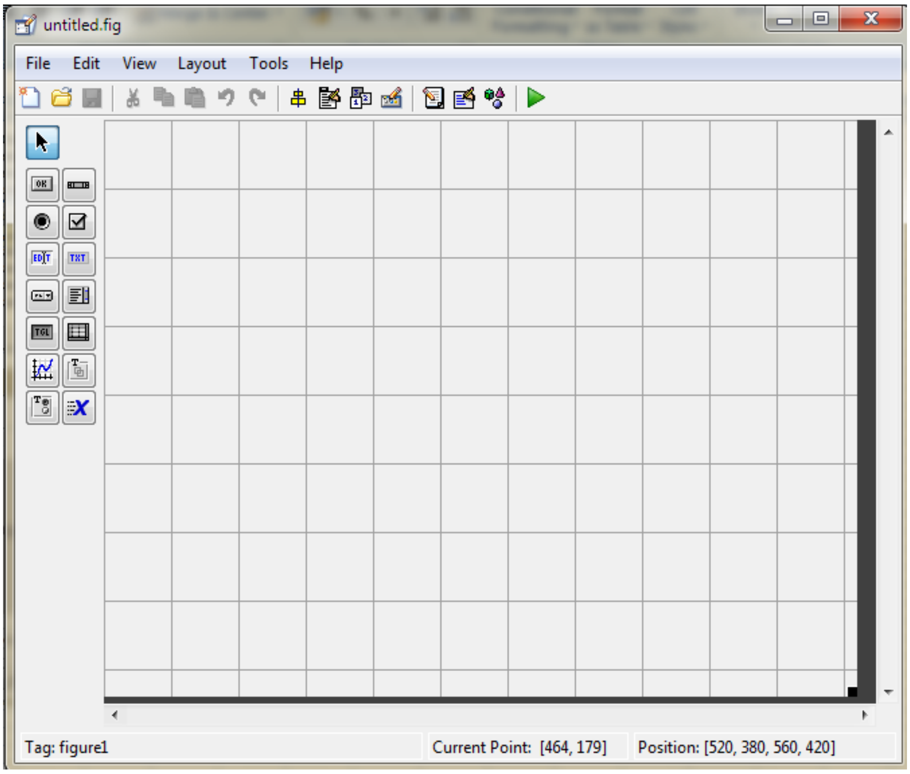
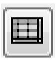
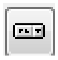


Figure 4.19 GUIDE Layout Editor

4) Use GUIDE Layout Editor to create the injection molding machine selection system. Drag and drop all required tools from the right side of the screen into the space. The tools that were used are as follows;

 : Static Text  : Push Button

 : Table  : Pop-up Menu

Edit an object by right-clicking on it and choose Property Inspector that is shown in Figure 4.20. The features and properties of the object can be modified by an Inspector dialog in Figure 4.21.

5) The finished layout is illustrated in Figure 4.22. Save GUI layout as .fig file type, names IMM_Selection.fig. At the same time, MATLAB will automatically create another one file, names IMM_Selection.m. This file is the multiple-file (M-File) type. The M-File is the text file that contains MATLAB codes Programmer can add function codes in M-File to make it execute as they want.

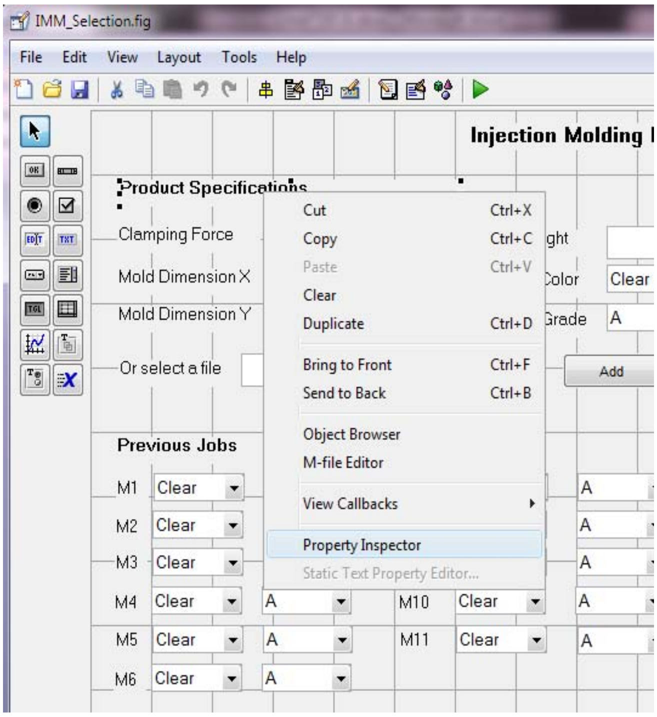


Figure 4.20 Editing GUI objects by Property Inspector

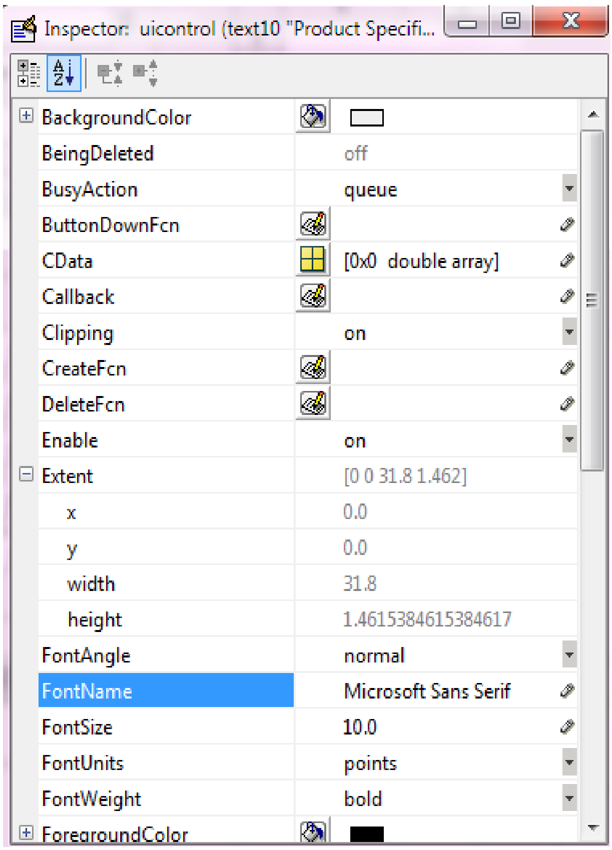


Figure 4.21 Property Inspector dialog

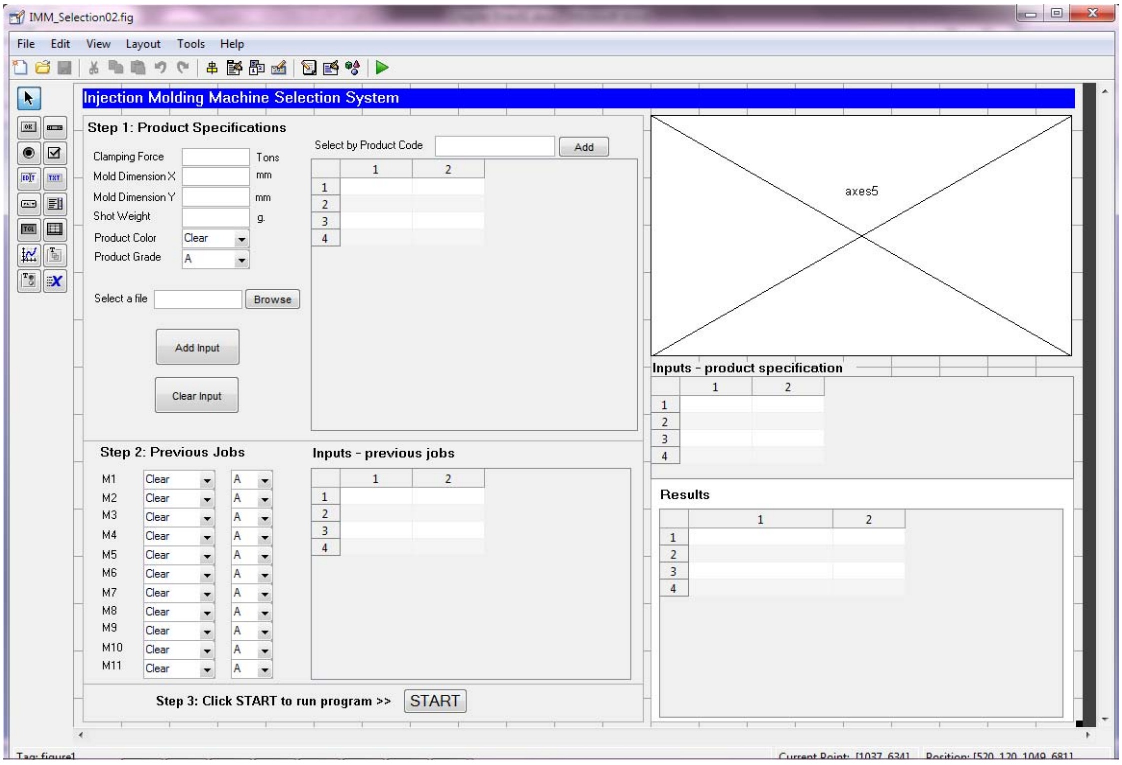


Figure 4.22 Completed GUI Layout

4.1.3.2 GUI Programming

The programming of GUI is performed on M-File within the MATLAB Editor window. The concept of M-File generating is the association all GUI's objects with functions except the ones that do not need to be calculated or have functions such as the static text object. There are particularly MATLAB scripts for each object, and consequently every object is highly required to validate their function and order. The procedure of programming the M-File is as follows.

1) Do a right-click on the object to add functions. And choose View Callbacks > Callback. Callbacks are functions that execute in response to some action by the user. After that, MC_Selection.m will be opened in the MATLAB Editor. And the collection of initializing codes will appear. The codes have followed by '%' symbol would not be operated by MATLAB Compiler. Any functions should be added to command it. There are three methods to review MATLAB codes that are:

A. Click on the object. Choose View Callbacks. And click CreateFCN. This method will be used for the Static Text object. MATLAB will show the initializing codes of Static Text object (Figure 4.23).

```
% --- Executes during object creation, after setting all properties.
function text1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

Figure 4.23 Template of Static Text object

B. Click on the object. Choose View Callbacks. And click Callback. This method will be used for Radio Button, Check Box, Push Button, Edit Text, Slider and Pop-up Menu objects.

C. Click on the object. Choose View Callbacks, and click CellEditCallback. This method will be used for Table object.

2) Add functions or codes below the last row within the same object code set. For example, if we want to change the background color of 'text1' object from the default (gray) color to blue color, paste the following function in MALAB Editor (Figure 4.24). Click Save.

```
If    ispc && isequal(get(hObject,'BackgroundColor'),
      get(0,'defaultUicontrolBackgroundColor'))
      set(hObject,'BackgroundColor','blue');
end
```

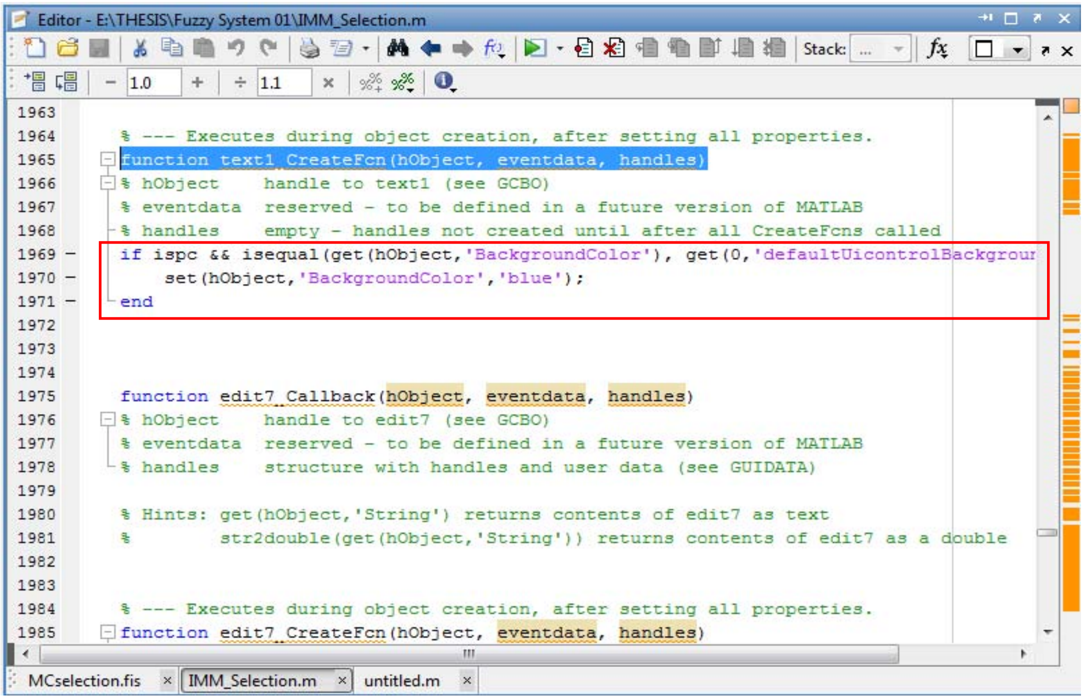


Figure 4.24 Adding the function to change Static Text background

3) Repeat 1) to 2) for other objects need to change the color. And click  to run all codes in IMM_Selection.m for validation of results.

The finished GUI Layout is shown in Figure 4.25. For other object type, other functions are put in the same position.

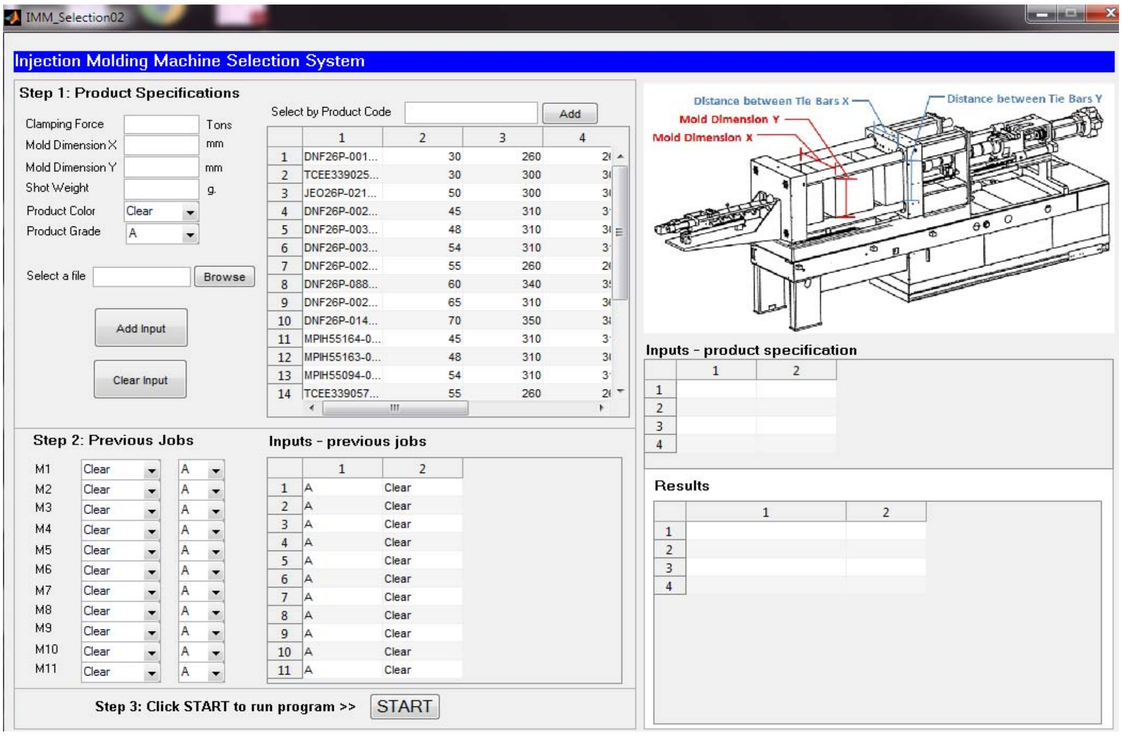


Figure 4.25 Finished GUI Layout

4.2 Script descriptions

All code was done in MATLAB using object oriented programming. The prototype system has several collections of function in order to combine the one system.

4.2.1 The composition of an object

The main components in each object code are in the following.

1) Expand/Collapse symbol: Functions are separated by the symbol as in Figure 4.25. The functions and comments of an object will be applied within it. One symbol is included the boxes of ‘-’ and ‘+’. When one click is applied, MATLAB will expand or collapse their functions (Figure 4.26).

2) “%” symbol: The symbol that is always used to avoid the compilement of that followed letters. The method of changing script to be text by applying “%” in front of scripts and that script will be changed to green color then.

Figure 4.26 Main component layout

3) The MATLAB code that implements the function of the button to be executed when applying a click by user. Any data that you want to pass between functions should be saved in the handles structure, handle is an unique number that is assigned to a graphical object, which should be located in at the end of each function.

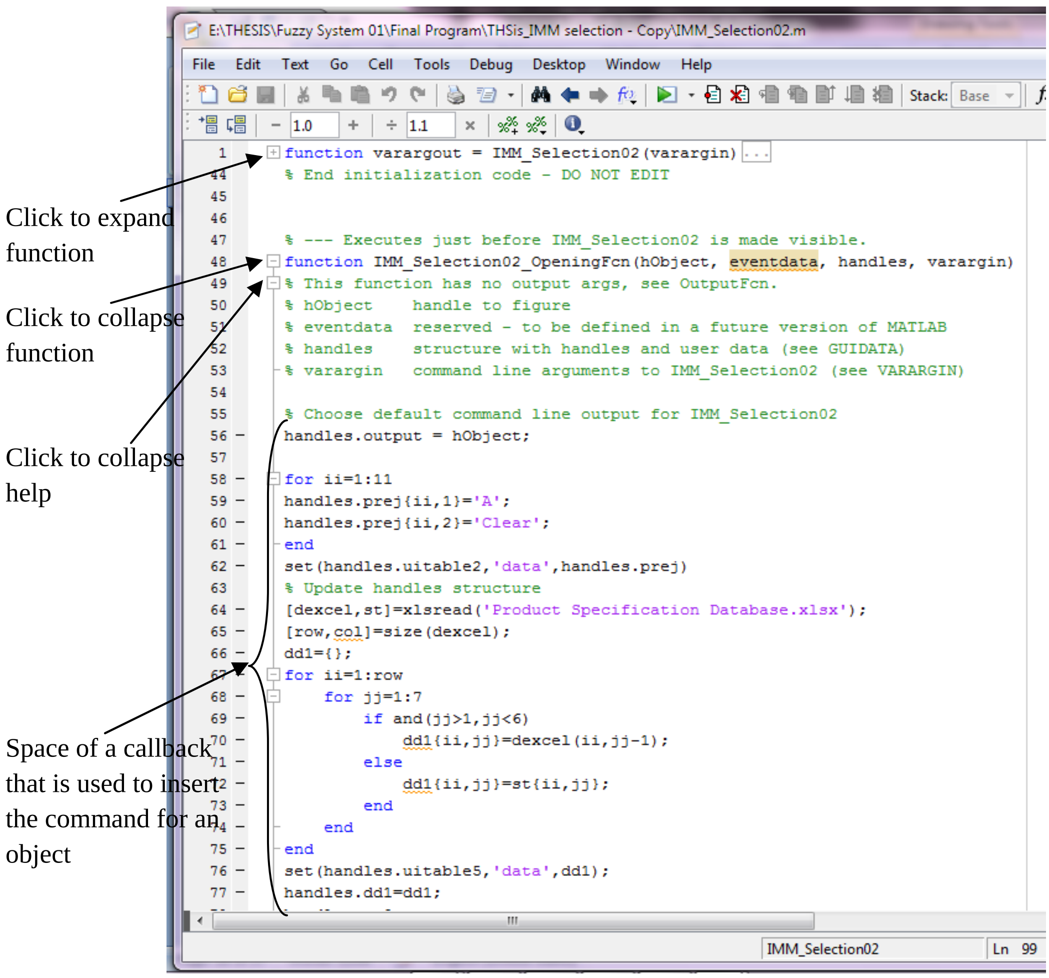


Figure 4.27 Object and compositions

The code is known as a call back. There must be a callback to implement the function of each graphical component on the GUI. The end of this code should be terminated with "end". Please see Figure 4.27.

4.2.2 Scripts of IMMSelection02.m file

This section shows the scripts that are included in each object and the explanation of those scripts. These scripts could be able to classified into four parts. First, there is the part of "Step 1: Product Specifications". Second, the table of "Inputs – product specification" will be explained within this part. Third, the scripts of "Step 2: Previous Jobs" and the table of "Input – previous jobs" will be shown here. Finally, this part will illustrate the scripts and their descriptions of "Step 3: Click START to run program" and concerned scripts for table of Results.

<u>Part 1</u>: The scripts of "Step 1: Product Specifications"

This part is composed of the objects in Figure 4.28. For more details of functions and scripts, please see the description as follows.

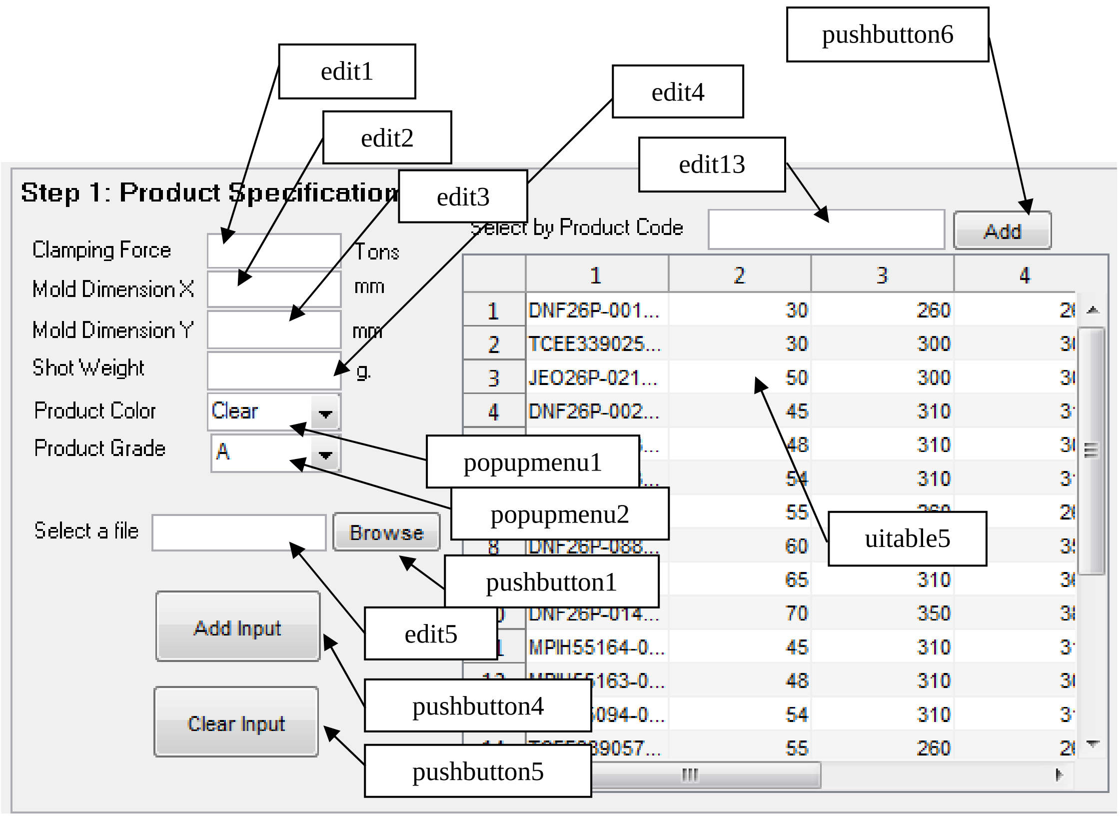


Figure 4.28 Compositions of "Step 1: Product Specifications" user interface

1) The initial code of M-file, this part indicates the name of M-file (IMM_Selection) and GUI file.

```
% IMM_SELECTION02 M-file for IMM_Selection02.fig
% IMM_SELECTION02, by itself, creates a new IMM_SELECTION02 or raises
  the existing singleton*.
%
% H = IMM_SELECTION02 returns the handle to a new IMM_SELECTION02 or
  the handle to the existing singleton*.
%
% IMM_SELECTION02('CALLBACK',hObject,eventData,handles,...) calls the
  local
% function named CALLBACK in IMM_SELECTION02.M with the given input
  arguments.
%
% IMM_SELECTION02('Property','Value',...) creates a new
  IMM_SELECTION02 or raises the
% existing singleton*.  Starting from the left, property value pairs
% are applied to the GUI before IMM_Selection02_OpeningFcn gets
  called.  An
% unrecognized property name or invalid value makes property
% application stop.  All inputs are passed to
IMM_Selection02_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help IMM_Selection02

% Last Modified by GUIDE v2.5 09-Jan-2013 00:29:46

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
```

```
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @IMM_Selection02_OpeningFcn, ...
                   'gui_OutputFcn',  @IMM_Selection02_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

2) This part is the OpeningFcn for opening the function. It is the first callback in every GUI M-file. You can use it to perform tasks that need to be done before the user has access to the GUI. In this case, OpeningFcn is used to create default data, e.g. “A” and “Clear” as in Figure 4.27. The scripts of “set(handles.uitable2,'data',handles.prej)” are used to set the default data to uitable2 to column 1 and 2. The scripts of “[dexcel,st]=xlsread('Product Specification Database.xlsx'); [row,col]=size(dexcel); dd1={};” are used to read and update data of product code and their parameters from “Product Specification Database.xls” file. By the name of these data is dd1 which are always updated and shown in uitable5 from the scripts of “set(handles.uitable5,'data',dd1);”. During the using of program, the picture of injection molding machine will be shown in user interface all the time as in Figure 4.27. The scripts of “axes(handles.axes5); im=imread('IMMresized.jpg'); imshow(im)” are the setting of IMMresized3.jpg to the axes5 object as in Figure 4.29.

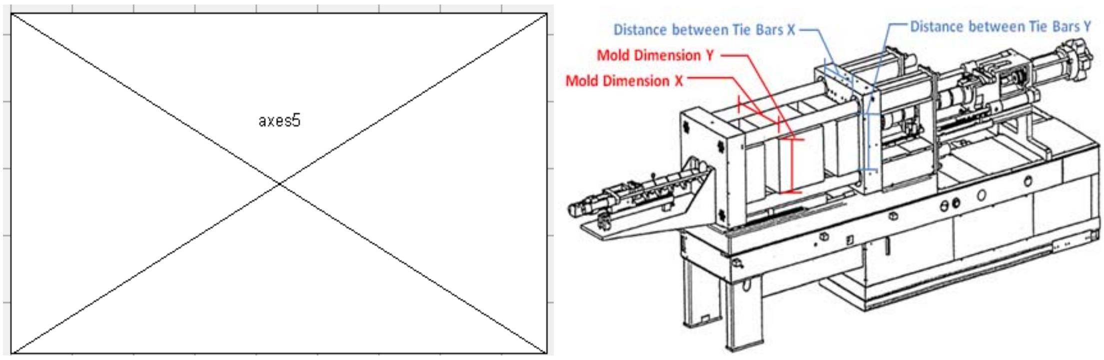
```
% --- Executes just before IMM_Selection is made visible.
function IMM_Selection02_OpeningFcn(hObject, eventdata, handles, 
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% varargin   command line arguments to IMM_Selection02 (see VARARGIN)

% Choose default command line output for IMM_Selection02
handles.output = hObject;

for ii=1:11
handles.prej{ii,1}='A';
handles.prej{ii,2}='Clear';
end
set(handles.uitable2,'data',handles.prej)
% Update handles structure
[dexcel,st]=xlsread('Product Specification Database.xlsx');
[row,col]=size(dexcel);
dd1={};
for ii=1:row
    for jj=1:7
        if and(jj>1,jj<6)
            dd1{ii,jj}=dexcel(ii,jj-1);
        else
            dd1{ii,jj}=st{ii,jj};
        end
    end
end
set(handles.uitable5,'data',dd1);
handles.dd1=dd1;
handles.n=0;
axes(handles.axes5)
im=imread('IMMresized3.jpg');
imshow(im)
guidata(hObject, handles);

% UIWAIT makes IMM_Selection wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```



(a) (b)

Figure 4.29 The axes5 object where the picture names "IMMresized3.jpg" have stored (a) Before run program (b) After run program

3) The script in this part is included function that order to return the output into the command line (in command window) by following the form of "varargout{1} = handles.output". The handles structure is a variable that GUIDE passes as an argument to all component callbacks.

```
% --- Outputs from this function are returned to the command line.
function varargout = IMM_Selection02_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

4) The edit1 object is located for inputting of Clamping Force value to the system as in Figure 4.28. This parameter value will be assigned as "string" number, represented by "c1". The limitation of input value is conditioned by the statement of "if and(c1>=30, c1<=75) else msgbox ('Please input Clamping Force between 30 to 75', 'Please input Clamping Force between 30 to 75'" which means if user enter the Clamping Force (c1) value that lower than 30 or upper than 75, the message of "Please input Clamping Force between 30 to 75" will appear. And the following codes of the

edit1 object are the setting the background color of the object to be white with "if" function.

```
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1
as a double

c1=str2num(get(handles.edit1,'string'));
if and(c1>=30,c1<=75)
else
    msgbox('Pleas input Clamping Force between 30 to 75','Pleas input
Clamping Force between 30 to 75');
end

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

5) The edit2 object is located for inputting of Mold dimension X value to the system as in Figure 4.28. This parameter value will be assigned as "string" number, represented by "c1". The limitation of input value is conditioned by the statement of "if and(c1>=260, c1<=350) else msgbox ('Please input Mold Dimension X between 260 to 350','Please input Mold Dimension X between 260 to 350')" which

means if user enter the Mold Dimension X (c1) value that lower than 260 or upper than 350, the message of “Please input Mold Dimension X between 260 to 350” will appear. And the following codes of the edit1 object are the setting the background color of the object to be white with “if” function.

```
function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2
as a double
c1=str2num(get(handles.edit2,'string'));
if and(c1>=260,c1<=350)
else
    msgbox('Please input Mold Dimension X between 260 to 350','Please
input Mold Dimension X between 260 to 350');
end

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

6) The edit3 object is located for inputting of Mold dimension Y value to the system as in Figure 4.28. This parameter value will be assigned as “string”

number, represented by "c1". The limitation of input value is conditioned by the statement of "if and(c1>=260, c1<=380) else msgbox ('Please input Mold Dimension Y between 260 to 380','Please input Mold Dimension Y between 260 to 380')" which means if user enter the Mold Dimension Y (c1) value that lower than 260 or upper than 380, the message of "Please input Mold Dimension Y between 260 to 380" will appear. And the following codes of the edit1 object are the setting the background color of the object to be white with "if" function.

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3
as a double
c1=str2num(get(handles.edit3,'string'));
if and(c1>=260,c1<=380)
else
    msgbox('Pleas input Mold Dimension Y between 260 to 380','Pleas
input Mold Dimension Y between 260 to 380');
end

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

7) The edit4 object is located for inputting of Shot Weight value to the system as in Figure 4.28. This parameter value will be assigned as “string” number, represented by “c1”. The limitation of input value is conditioned by the statement of “if and(c1>=1, c1<=100) else msgbox ('Please input Shot Weight between 1 to 100','Please input Shot Weight between 1 to 100')” which means if user enter the Shot Weight (c1) value that lower than 1 or upper than 100, the message of “Please input Shot Weight between 1 to 100” will appear. And the following codes of the edit1 object are the setting the background color of the object to be white with “if” function.

```
function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4
as a double
c1=str2num(get(handles.edit4,'string'));
if and(c1>=1,c1<=100)
else
    msgbox('Pleas input Shot Weight between 1 to 100','Pleas input
Shot Weight between 1 to 100');
end

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end
```

8) This object is called popupmenu1 which is in drop box form as in Figure 4.28. Material grade property that is required to assign to a machine will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

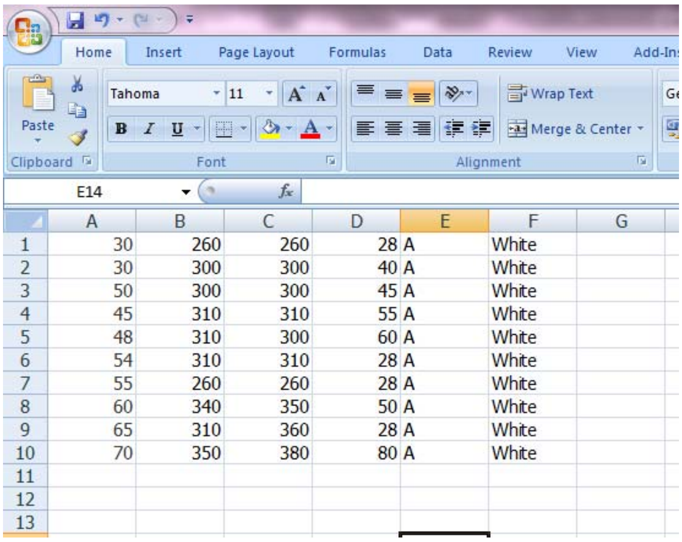
9) This object is called popupmenu2 which is in drop box form as in Figure 4.27. Product color property that is required to assign to a machine will be

selected here. All choices are included with clear, white, grey, and black colors. And these choices have set in string in uicontrol menu and will not be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu2

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

10) The Browse button is functioned by codes of pushbutton1 object as in Figure 4.28. Codes will call data from the file names “data.xls”. These data contain 4 columns of number and 2 columns of characters as in Figure 4.29. All variable values will be transformed to metric with loop function. After that, all data will show in uitable1 by collecting in handles structure both num (number) and str (string) because we want to use this function that had stored in handle again in the next step.



	A	B	C	D	E	F	G
1	30	260	260	28	A	White	
2	30	300	300	40	A	White	
3	50	300	300	45	A	White	
4	45	310	310	55	A	White	
5	48	310	300	60	A	White	
6	54	310	310	28	A	White	
7	55	260	260	28	A	White	
8	60	340	350	50	A	White	
9	65	310	360	28	A	White	
10	70	350	380	80	A	White	
11							
12							
13							

Figure 4.29 Sample data in “data.xlsx” file

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname] = ...
     uigetfile({'*.xls*';},'File Selector');
 [num,str]=xlsread([pathname filename]);
 set(handles.edit5,'string',[pathname filename]);
 data=[];
 for ii=1:size(num,1)
     for jj=1:4
     data{ii,jj}=num(ii,jj);
     end
     data{ii,5}=char(str{ii,5});
     data{ii,6}=char(str{ii,6});
 end
%  data
 set(handles.uitable1,'data',data)
 handles.num=num;
 handles.str=str;


 guidata(hObject, handles);
```

11) This object is edit5 as in figure 4.28. After selecting of data.xls from the directory, pathname and filename of data.xls will be shown in edit5 object. The function for this object are only setting of background color in white.

```
function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5
as a double
% --- Executes during object creation, after setting all properties.

function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

12) This object is Add Input button as in Figure 4.28. All data from edit1, edit2, edit3, edit4, popupmenu1, and popupmenu2 will be added to uitable1 by applying one click at Add Input button. All variables from edit1, edit2, edit3, and edit4 will then be converted from string to value (number). And they will be identified names as Clampf, TiebarX, TiebarY, and Shorwt respectively. Variables from popupmenu1 will be got to store in name Pgrds which is string for displaying in uitable1 object and also got to store in value form with name Pgrdi by scripts of “Pgrds = get(handles.popupmenu1,'string'); Pgrdi = get(handles.popupmenu1, 'value')”. Finally, “Pgrads” and “Pgrdi” will be represented by “Pgrd” by scripts of

Pgrd=Pgrds{Pgrdi}. As in Pcols, Pcoli and Pcol, these variables are used to represent the variable input of product color. Pcols represents the color in string form. Pcoli represents the color in value form. And Pcol represents Pcols and Pcoli by scripts of Pcol=Pcols{Pcoli}. The variables of Clampf, TiebarX, TiebarY, and Shotwt have to be stored in matrix i and j by i = 1 row and j = 4 column. And the variables of Pgrd and Pcol have to be stored in the column 5 and 6 of uitable1. The scripts in handles.num(ii,1)=num(1) to handles.num(ii,4)=num(4) are used to refer the function of "num" from codes of pushbutton1 and to store in column 1 to 4. For the scripts of handles.str{ii,5}=char(Pgrd) and handles.str{ii,6}=char(Pcol), these scripts are used to let MATLAB to store Pgrd and Pcol that are characters with string class in column 5 and 6.

```
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
data=get(handles.uitable1,'data');
handles.n=handles.n+1;

Clampf=str2num(get(handles.edit1,'string'));
TiebarX=str2num(get(handles.edit2,'string'));
TiebarY=str2num(get(handles.edit3,'string'));
Shotwt=str2num(get(handles.edit4,'string'));
Pgrds=get(handles.popupmenu1,'string');
Pgrdi=get(handles.popupmenu1,'value');
Pgrd=Pgrds{Pgrdi};
Pcols=get(handles.popupmenu2,'string');
Pcoli=get(handles.popupmenu2,'value');
Pcol=Pcols{Pcoli};
n=handles.n;
num=[Clampf,TiebarX,TiebarY,Shotwt];
 for ii=n:n
     for jj=1:4
     data{ii,jj}=num(1,jj);
     end
```

```
     data{ii,5}=char(Pgrd);
     data{ii,6}=char(Pcol);
     handles.num(ii,1)=num(1);
     handles.num(ii,2)=num(2);
     handles.num(ii,3)=num(3);
     handles.num(ii,4)=num(4);
     handles.str{ii,5}=char(Pgrd);
     handles.str{ii,6}=char(Pcol);
 end
%  handles.num
%  handles.str
%  data
 set(handles.uitable1,'data',data)
 guidata(hObject, handles);
```

13) This object is pushbutton5 or Clear Input button as in Figure 4.28. There scripts stand for deletion all data in uitable1. After clicking at this button, variables will be eliminated both num and str with represented by [ ] symbols in scripts.

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.uitable1,'data','')
 handles.n=0;
  handles.num=[];
 handles.str=[];
 guidata(hObject, handles);
```

14) This object is uitable5 where the database of product and their specification have stored as in Figure 4.28. This data is linked to the file names “Product Specification Database.xlsx”. User can find out the required product and its information from this table e.g. Clamping force, Mold Dimension X, Mold dimension Y, Shot weight, Product color and Product grade.

```
% --- Executes when entered data in editable cell(s) in uitable5.
function uitable5_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to uitable5 (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%   Indices: row and column indices of the cell(s) edited
%   PreviousData: previous data for the cell(s) edited
%   EditData: string(s) entered by the user
%   NewData: EditData or its converted form set on the Data property.
Empty if Data was not changed
%   Error: error string when failed to convert EditData to
appropriate value for Data
% handles    structure with handles and user data (see GUIDATA)
```

15) This object named edit13 as in Figure 4.28. Users have to type the product code here to search product code from database if they do not know the specification of that product. The function that is used here is only the setting of background color.

```
function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%        str2double(get(hObject,'String')) returns contents of edit13
as a double
% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

16) This object named pushbutton6 or Add button which has responsible to add all variables of the selected product code in uitable5 into uitable1. Please see in Figure 4.28. The scripts of “data=get(handles.uitable1,'data')” are put for setting the output after clicking “Add” button to show in uitable1. The scripts of “strfind=get(handles.edit13,'string')” are used to discover string that has specified in edit13 object from “dd1” array. The scripts of “if strcmp(strfind,dd1{ii,1}); n1=ii” are used to compare the string by concerning of capital and small letters (Case Sensitive). If the string in edit13 and dd1 array are same both capital and small letters, the return output will be 1 and will be shown on the command window. But, if that strings are different, the return output will be 0 by following the scripts of “n=1; for ii=1:row; end”. The scripts of “if n1==0; msgbox('The System cannot find your input','The System cannot find your input')” are used to inform user if the string (product code) that has entered is not same as any product code in the product specification database. In opposite, if the system can found the same string, data of that string from column 1 to 7 of dd1 array (e.g. Clampf, TiebarX, TiebarY, Shotwt, Pgrd and Pcol) will be shown in uitable1. The scripts of “handles.num(ii,1)=num(1); handles.num(ii,2)=num(2); handles.num(ii,3)=num(3); handles.num(ii,4)=num(4);” are assignment of variables to be stored in number form with handle structure. But, in lines of “handles.str{ii,5}=char(Pgrd); handles.str{ii,6}=char(Pcol);” were aimed to assign in character form. Finally, the scripts of “set(handles.uitable1,'data',data) guidata(hObject, handles)” are setting name of data in uitable1 as “data”.

```
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
data=get(handles.uitable1,'data');
handles.n=handles.n+1;

dd1=handles.dd1;
strfind=get(handles.edit13,'string');
[row,col]=size(dd1);
n1=0;
```

```
for ii=1:row
    if strcmp(strfind,dd1{ii,1})
        n1=ii;
    end
end
if n1==0
    msgbox('The System cannot find your input','The System cannot
find your input');
else
Clampf=dd1{n1,2};
TiebarX=dd1{n1,3};
TiebarY=dd1{n1,4};
Shotwt=dd1{n1,5};
Pgrd=dd1{n1,6};
Pcol=dd1{n1,7};
n=handles.n;
num=[Clampf,TiebarX,TiebarY,Shotwt];
 for ii=n:n
     for jj=1:4
     data{ii,jj}=num(1,jj);
     end
     data{ii,5}=char(Pgrd);
     data{ii,6}=char(Pcol);
  handles.num(ii,1)=num(1);
     handles.num(ii,2)=num(2);
     handles.num(ii,3)=num(3);
     handles.num(ii,4)=num(4);
 handles.str{ii,5}=char(Pgrd);
 handles.str{ii,6}=char(Pcol);
 end
set(handles.uitable1,'data',data)
 guidata(hObject, handles);
end
```

Part 2: The scripts of "Input – product specification"

This part composed by the objects as in Figure 4.30. For more details of functions and scripts, please see the description as follows.

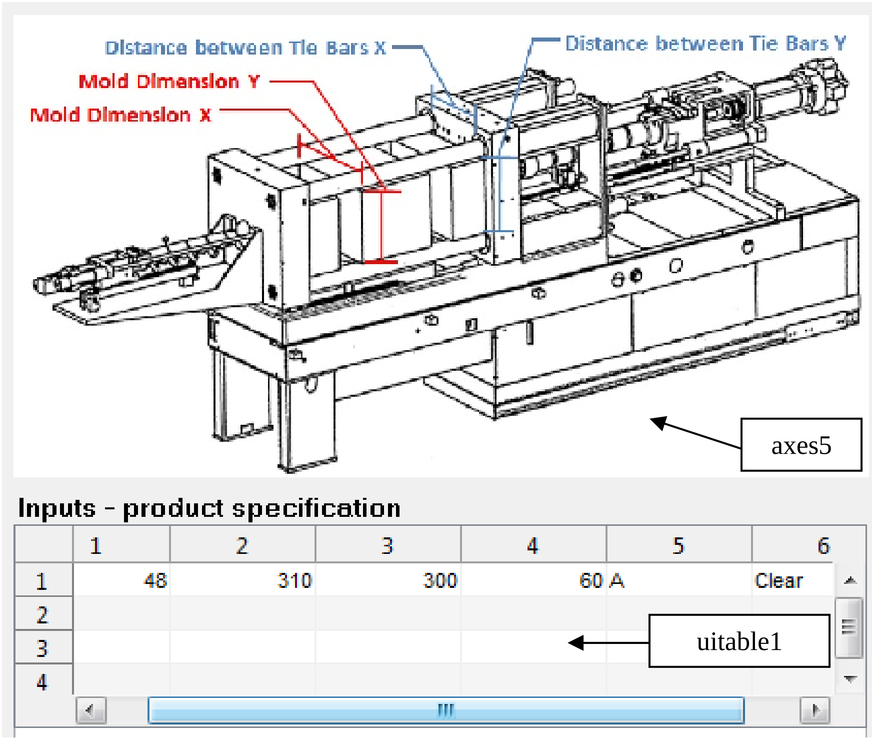


Figure 4.30 Input – product specification table

1) This object is uitable1 where all input variables will appear as in Figure 4.30. For the object names axes5, this object will show the picture of injection molding machine for guiding user to identify their variable correctly. The scripts of this object was mentioned at Openingfcn of the program.

```
% --- Executes when entered data in editable cell(s) in uitable1.
function uitable1_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to uitable1 (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%   Indices: row and column indices of the cell(s) edited
%   PreviousData: previous data for the cell(s) edited
%   EditData: string(s) entered by the user
%   NewData: EditData or its converted form set on the Data property.
Empty if Data was not changed
```

```
%   Error: error string when failed to convert EditData to
appropriate value for Data
% handles    structure with handles and user data (see GUIDATA)
```

Part 3: The scripts of "Step 2: Previous Jobs"

This part composed by the objects as in Figure 4.31. For more details of functions and scripts, please see the description as follows.

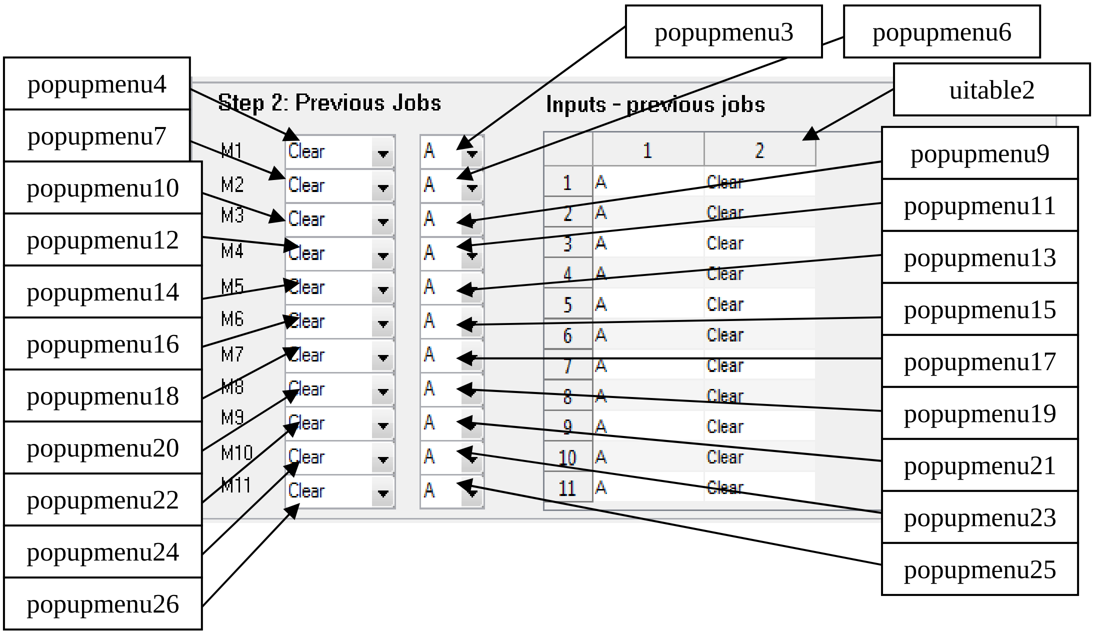


Figure 4.31 The compositins of "Step 2: Previous Jobs"

1) This object is called popupmenu3 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M1 will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu3
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu3
Lgrd=get(handles.popupmenu3,'value');
if Lgrd==1
handles.prej{1,1}='A';
elseif Lgrd==2
handles.prej{1,1}='B';
elseif Lgrd==3
handles.prej{1,1}='C';
elseif Lgrd==4
handles.prej{1,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

2) This object is called popupmenu4 in form of a drop box as in Figure 4.31. Product color property that is required to assign to machine M1 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu4.
```

```
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu4
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu4

Lcol=get(handles.popupmenu4,'value');
if Lcol==1
handles.prej{1,2}='Clear';
elseif Lcol==2
handles.prej{1,2}='White';
elseif Lcol==3
handles.prej{1,2}='Grey';
elseif Lcol==4
handles.prej{1,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

3) This object is called popupmenu6 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M2 will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu6.
function popupmenu6_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu6
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu6
Lgrd=get(handles.popupmenu6,'value');
if Lgrd==1
handles.prej{2,1}='A';
elseif Lgrd==2
handles.prej{2,1}='B';
elseif Lgrd==3
handles.prej{2,1}='C';
elseif Lgrd==4
handles.prej{2,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

4) This object is called popupmenu7 which is in drop box form as in Figure 4.31. Product color property that is required to assign to machine M2 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu7.
function popupmenu7_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu7
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu7
Lcol=get(handles.popupmenu7,'value');
if Lcol==1
handles.prej{2,2}='Clear';
elseif Lcol==2
handles.prej{2,2}='White';
elseif Lcol==3
handles.prej{2,2}='Grey';
elseif Lcol==4
handles.prej{2,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);
```

```
% --- Executes during object creation, after setting all properties.
function popupmenu7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

5) This object is called popupmenu9 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M3 will be selected here. All choices are included with A, B, C, and D grades. These choices have set in string in uicontrol menu and will not be calculated. Thus, codes in the m-file are only the setting of white background color.

```
%  guidata(hObject, handles);
% --- Executes on selection change in popupmenu9.
function popupmenu9_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu9
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu9
Lgrd=get(handles.popupmenu9,'value');
if Lgrd==1
handles.prej{3,1}='A';
elseif Lgrd==2
handles.prej{3,1}='B';
elseif Lgrd==3
```

```
handles.prej{3,1}='C';
elseif Lgrd==4
handles.prej{3,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

6) This object is called popupmenu10 which is in drop box form as in Figure 4.31. Product color property that is required to assign to machine M3 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu10.
function popupmenu10_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu10 contents as cell array
```

```
%        contents{get(hObject,'Value')} returns selected item from
popupmenu10
Lcol=get(handles.popupmenu10,'value');
if Lcol==1
handles.prej{3,2}='Clear';
elseif Lcol==2
handles.prej{3,2}='White';
elseif Lcol==3
handles.prej{3,2}='Grey';
elseif Lcol==4
handles.prej{3,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

7) This object is called popupmenu11 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M4 will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu11.
```

```
function popupmenu11_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu11 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu11
Lgrd=get(handles.popupmenu11,'value');
if Lgrd==1
handles.prej{4,1}='A';
elseif Lgrd==2
handles.prej{4,1}='B';
elseif Lgrd==3
handles.prej{4,1}='C';
elseif Lgrd==4
handles.prej{4,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

8) This object is called popupmenu12 which is in drop box form as in Figure 4.31. Product color property that is required to assign to machine M4 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
 % --- Executes on selection change in popupmenu12.
function popupmenu12_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu12 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu12
Lcol=get(handles.popupmenu12,'value');
if Lcol==1
handles.prej{4,2}='Clear';
elseif Lcol==2
handles.prej{4,2}='White';
elseif Lcol==3
handles.prej{4,2}='Grey';
elseif Lcol==4
handles.prej{4,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

9) This object is called popupmenu13 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M5 will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu13.
function popupmenu13_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu13 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu13
Lgrd=get(handles.popupmenu13,'value');
if Lgrd==1
handles.prej{5,1}='A';
elseif Lgrd==2
handles.prej{5,1}='B';
elseif Lgrd==3
handles.prej{5,1}='C';
elseif Lgrd==4
handles.prej{5,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);
```

```
% --- Executes during object creation, after setting all properties.
function popupmenu13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

10) This object is called popupmenu14 which is in drop box form as in Figure 4.31. Product color property that is required to assign to machine M5 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu14.
function popupmenu14_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu14 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu14
Lcol=get(handles.popupmenu14,'value');
if Lcol==1
handles.prej{5,2}='Clear';
elseif Lcol==2
handles.prej{5,2}='White';
elseif Lcol==3
```

```
handles.prej{5,2}='Grey';
elseif Lcol==4
handles.prej{5,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

11) This object is called popupmenu15 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M6 will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu15.
function popupmenu15_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu15 contents as cell array
```

```
%        contents{get(hObject,'Value')} returns selected item from
popupmenu15
Lgrd=get(handles.popupmenu15,'value');
if Lgrd==1
handles.prej{6,1}='A';
elseif Lgrd==2
handles.prej{6,1}='B';
elseif Lgrd==3
handles.prej{6,1}='C';
elseif Lgrd==4
handles.prej{6,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

12) This object is called popupmenu16 which is in drop box form as in Figure 4.31. Product color property that is required to assign to machine M6 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu16.
```

```
function popupmenu16_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu16 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu16
Lcol=get(handles.popupmenu16,'value');
if Lcol==1
handles.prej{6,2}='Clear';
elseif Lcol==2
handles.prej{6,2}='White';
elseif Lcol==3
handles.prej{6,2}='Grey';
elseif Lcol==4
handles.prej{6,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

13) This object is called popupmenu17 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M7 will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu17.
function popupmenu17_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu17 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu17
Lgrd=get(handles.popupmenu17,'value');
if Lgrd==1
handles.prej{7,1}='A';
elseif Lgrd==2
handles.prej{7,1}='B';
elseif Lgrd==3
handles.prej{7,1}='C';
elseif Lgrd==4
handles.prej{7,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

14) This object is called popupmenu18 which is in drop box form as in Figure 4.31. Product color property that is required to assign to machine M7 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu18.
function popupmenu18_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu18 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu18
Lcol=get(handles.popupmenu18,'value');
if Lcol==1
handles.prej{7,2}='Clear';
elseif Lcol==2
handles.prej{7,2}='White';
elseif Lcol==3
handles.prej{7,2}='Grey';
elseif Lcol==4
handles.prej{7,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);
```

```
% --- Executes during object creation, after setting all properties.
function popupmenu18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

15) This object is called popupmenu19 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M8 will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu19.
function popupmenu19_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu19 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu19
Lgrd=get(handles.popupmenu19,'value');
if Lgrd==1
handles.prej{8,1}='A';
elseif Lgrd==2
handles.prej{8,1}='B';
elseif Lgrd==3
```

```
handles.prej{8,1}='C';
elseif Lgrd==4
handles.prej{8,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

16) This object is called popupmenu20 which is in drop box form as in Figure 4.31. Product color property that is required to assign to machine M8 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu20.
function popupmenu20_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu20 contents as cell array
```

```
%        contents{get(hObject,'Value')} returns selected item from
popupmenu20
Lcol=get(handles.popupmenu20,'value');
if Lcol==1
handles.prej{8,2}='Clear';
elseif Lcol==2
handles.prej{8,2}='White';
elseif Lcol==3
handles.prej{8,2}='Grey';
elseif Lcol==4
handles.prej{8,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu20_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

17) This object is called popupmenu21 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M9 will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu21.
function popupmenu21_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu21 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu21
Lgrd=get(handles.popupmenu21,'value');
if Lgrd==1
handles.prej{9,1}='A';
elseif Lgrd==2
handles.prej{9,1}='B';
elseif Lgrd==3
handles.prej{9,1}='C';
elseif Lgrd==4
handles.prej{9,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

18) This object is called popupmenu22 which is in drop box form as in Figure 4.31. Product color property that is required to assign to machine M9 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu22.
function popupmenu22_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu22 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu22
Lcol=get(handles.popupmenu22,'value');
if Lcol==1
handles.prej{9,2}='Clear';
elseif Lcol==2
handles.prej{9,2}='White';
elseif Lcol==3
handles.prej{9,2}='Grey';
elseif Lcol==4
handles.prej{9,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

19) This object is called popupmenu23 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M10 will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu23.
function popupmenu23_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu23 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu23
Lgrd=get(handles.popupmenu23,'value');
if Lgrd==1
handles.prej{10,1}='A';
elseif Lgrd==2
handles.prej{10,1}='B';
elseif Lgrd==3
handles.prej{10,1}='C';
elseif Lgrd==4
handles.prej{10,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);
```

```
% --- Executes during object creation, after setting all properties.
function popupmenu23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

20) This object is called popupmenu24 which is in drop box form as in Figure 4.31. Product color property that is required to assign to machine M10 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu24.
function popupmenu24_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu24 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu24
Lcol=get(handles.popupmenu24,'value');
if Lcol==1
handles.prej{10,2}='Clear';
elseif Lcol==2
handles.prej{10,2}='White';
elseif Lcol==3
handles.prej{10,2}='Grey';
```

```
elseif Lcol==4
handles.prej{10,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

21) This object is called popupmenu25 which is in drop box form as in Figure 4.31. Material grade property that is required to assign to machine M11 will be selected here. All choices are included with A, B, C, and D grades. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu25.
function popupmenu25_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu25 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu25 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu25
```

```
Lgrd=get(handles.popupmenu25,'value');
if Lgrd==1
handles.prej{11,1}='A';
elseif Lgrd==2
handles.prej{11,1}='B';
elseif Lgrd==3
handles.prej{11,1}='C';
elseif Lgrd==4
handles.prej{11,1}='D';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu25_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu25 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

22) This object is called popupmenu26 which is in drop box form as in Figure 4.31. Product color property that is required to assign to machine M11 will be selected here. All choices are included with Clear, White, Grey, and Black. And these choices have set in string in uicontrol menu and will not to be calculated. Thus, codes in the m-file are only the setting of white background color.

```
% --- Executes on selection change in popupmenu26.
function popupmenu26_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu26 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu26 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu26
Lcol=get(handles.popupmenu26,'value');
if Lcol==1
handles.prej{11,2}='Clear';
elseif Lcol==2
handles.prej{11,2}='White';
elseif Lcol==3
handles.prej{11,2}='Grey';
elseif Lcol==4
handles.prej{11,2}='Black';
end
set(handles.uitable2,'data',handles.prej)
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu26_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu26 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

23) This object is uitable2 or Inputs – previous jobs table as in Figure 4.31. Product color and material grade of previous jobs by 11 machines will be shown here. The function of “function out=str2numABCD(in2)” is transforming of string (A, B, C,

and D) to numbers. The in2 variables are represented as material grades from A to D and in21 represents material grades in numeric form. Because the computer cannot compiles the value with characters form, the four conditions was developed. If in2 is A, in21 is 1. If in2 is B, in21 is 2. If in2 is C, in21 is 3. And if in2 is D, in21 is 4. At the same way, the function of “function in31=str2numcolor(in3)” is transforming of string (Clear, White, Grey, and Black) to numbers. The in3 variables are represented as product color from Clear to Black and in31 represents product color in numeric form. If in3 is Clear, in31 is 1. If in3 is White, in31 is 2. If in3 is Grey, in31 is 3. And if in3 is Black, in31 is 4.

```
% --- Executes when entered data in editable cell(s) in uitable2.
function uitable2_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to uitable2 (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%   Indices: row and column indices of the cell(s) edited
%   PreviousData: previous data for the cell(s) edited
%   EditData: string(s) entered by the user
%   NewData: EditData or its converted form set on the Data property.
Empty if Data was not changed
%   Error: error string when failed to convert EditData to
appropriate value for Data
% handles    structure with handles and user data (see GUIDATA)

function out=str2numABCD(in2)
in2
if strcmp(in2,'A')
    in21=1;
elseif strcmp(in2,'B')
    in21=2;
elseif strcmp(in2,'C')
    in21=3;
elseif strcmp(in2,'D')
    in21=4;
end
out=in21;
```

```
function in31=str2numcolor(in3)
if strcmp(in3,'Clear')
    in31=1;
elseif strcmp(in3,'White')
    in31=2;
elseif strcmp(in3,'Grey')
    in31=3;
elseif strcmp(in3,'Black')
    in31=4;
end
```

Part 4: The scripts of "Step 3: Click START to run program "

This part composed by the objects as in Figure 4.32. For more details of functions and scripts, please see the description as follows.

Figure 4.32 START button

1) This object is pushbutton3 which is called START button as in Figure 4.32. This object operates to calculate the results of injection molding machine selection program. The concerned data and functions involve with uitable1, uitable2, and MCselection.fis. This operation will be count the evaluation time with function "tic" and "toc" at the initial and end of calculation part scripts. The "Please wait…" message box will appear during calculation process with the function of "h1 = waitbar(0,'Please wait...')". The scripts of "a = readfis('MCselection')" are the opening of MCselection.fis. The concerned data will be referred again by "num=handles.num; str=handles.str; str2=handles.prej" from uitable1 and uitable2. The scripts of "[row1,col1]=size(num)" returns the size of matrix num in separate variables row1 and col1 for calculation with MCselection.fis. The system will calculate row-by-row by the function of "steps = row1". From the script "n=evalfis(num(ii,:), a)", it indicates n as the output of MCselection.fis. And the categories of n are included of 6 levels which are referred by "level=0:1:6". Each level is called MGr (Machine group).

In case of MGr is 1, the scripts of "in2=str(ii,5); in3=str(ii,6); nin2=str2numABCD(in2); nin3=str2numcolor(in3)" mean that in2 represents the string of material grade variables at column 5, in3 represents the string of product color variables at column 6, nin2 represents in2 that has transformed to numeric value (i.e. A is 1, B is 2, C is 3 and D is 4) and nin3 represents in3 that has transformed to numeric value (i.e. Clear is 1, White is 2, Grey is 3 and Black is 4). The product color of previous job in machine M1 and M2 were defined to be c2(1) and c2(2) in numerical value (i.e. Clear is 1, White is 2, Grey is 3 and Black is 4) by functions of “c2(1)=str2numcolor(str2{1,2})” and “c2(2)=str2numcolor(str2{2,2})” respectively. The script of “c3=abs(c2-nin3)” means the output of c2 minus nin3 is absolute value of c3. Then, c3 will be sorted in ascending order by the function of “[c3,indc3]=sort(c3)” for selecting the machine that returns the lowest c3. This function returns array of indices “indc3” where indc3 is an m-by-n matrix. If c3 is zero (0), it indicates that c2 and nin3 have the same value or the same product color. But, if M1 and M2 have the same c3 or “if c3(1)==c3(2)”, material grade will then be considerated. This process is just similar with product color consideration. The scripts of “c1(1)=str2numABCD(str2{1,1}); c1(2)=str2numABCD(str2{2,1})” are the transformation of string to numeric value of material grades A, B, C and D. The right machine can be defined by sorting of c3 with “c3=abs(c1-nin2); [c2,indc2]=sort(c3)”. The first row of array indc2 will be selected by follows “mach(ii)=indc2(1)”. If not, c3(1) is not same as c3(2), the first row of array indc3 will be selected instead by follows “mach(ii)=indc3(1)”. Please note that mach(ii) is the selected machine number. If the selected machine is number 2 or M2, mach(ii) is 2.

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tic
h1 = waitbar(0,'Please wait...');

a = readfis('MCselection');
num=handles.num;     %Input product/uitable1
str=handles.str;    %Input product/uitable1
str2=handles.prej;  %Previous Jobs Table
```

```
[row1,col1]=size(num);

steps = row1;

for ii=1:row1
 waitbar(ii / steps)
n=evalfis(num(ii,:), a);

level=0:1:6;
MGr=sum(n>level);
group0(ii)=MGr;
if MGr==1
in2=str(ii,5);
in3=str(ii,6);
nin2=str2numABCD(in2);
nin3=str2numcolor(in3);

         c2(1)=str2numcolor(str2{1,2});
         c2(2)=str2numcolor(str2{2,2});
         c3=abs(c2-nin3);
         [c3,indc3]=sort(c3);

 if c3(1)==c3(2)
         c1(1)=str2numABCD(str2{1,1});
         c1(2)=str2numABCD(str2{2,1});
         c3=abs(c1-nin2);
         [c2,indc2]=sort(c3);
         mach(ii)=indc2(1);   %%%%%%
 else
    mach(ii)=indc3(1);
 end
```

In case of MGr is 2, there are almost scripts same as MGr is 1. Except, the product color of previous job in machine M3, M4 and M5 were defined to be c2(1), c2(2) and c2(3) in numerical value (i.e. Clear is 1, White is 2, Grey is 3 and Black is 4) by functions of “c2(1)=str2numcolor(str2{3,2}), c2(2)=str2numcolor(str2{4,2}) and c2(3)=str2numcolor(str2{5,2})” respectively. The script of “c3=abs(c2-nin3)” means the output of c2 minus nin3 is absolute value of c3. Then, c3 will be sorted in ascending order by the function of “[c3,indc3]=sort(c3)” for selecting the machine that returns the lowest c3. This function returns array of indices “indc3” where indc3 is an m-by-n matrix. If c3 is zero (0), it indicates that c2 and nin3 have the same value or the same product color. But, if M3, M4 and M5 have the same c3 or “if c3(1)==c3(2)”, material grade will then be considerated. This process is just similar with product color consideration. The scripts of “c1(1)=str2numABCD(str2{3,1}), c1(2)=str2numABCD(str2{4,1}) and c1(3)=str2numABCD(str2{5,1})” are the transformation of string to numeric value of material grades A, B, C and D. The right machine can be defined by sorting of c3 with “c3=abs(c1-nin2); [c2,indc2]=sort(c3)”. The row number 1 of array indc2 will be selected. If the selected indc2 is row 1, program will show number of machine as 1 and plus with 2 to be 3. Thus, if the selected indc2 array is row 2 or 3, the selected machine will be 4 or 5 by follows “mach(ii)=indc2(1)+2”. But, if c3(1) is not equal to c3(2), the first row of array indc3 will be selected instead. And program will show number of machine as 3, that is calculated by row 1 plus with 2 to be 3, by follows “mach(ii)=indc3(1)+2”.

```
elseif MGr==2
in2=str(ii,5);
in3=str(ii,6);
nin2=str2numABCD(in2);
nin3=str2numcolor(in3);
         c2(1)=str2numcolor(str2{3,2});
         c2(2)=str2numcolor(str2{4,2});
         c2(3)=str2numcolor(str2{5,2});
         c3=abs(c2-nin3);
         [c3,indc3]=sort(c3);
 if c3(1)==c3(2)
         c1(1)=str2numABCD(str2{3,1});
         c1(2)=str2numABCD(str2{4,1});
         c1(3)=str2numABCD(str2{5,1});
         [c2,indc2]=sort(c1-nin2);
         mach(ii)=indc2(1)+2;
 else
         mach(ii)=indc3(1)+2;
 end
```

In case of MGr is 3, there are almost scripts same as MGr is 1. Except, the product color of previous job in machine M6 and M7 were defined to be c2(1) and c2(2) in numerical value (i.e. Clear is 1, White is 2, Grey is 3 and Black is 4) by functions of “c2(1)=str2numcolor(str2{6,2})” and “c2(2)=str2numcolor(str2{7,2})” respectively. The scripts of “c1(1)=str2numABCD(str2{6,1})” and “c1(2)=str2numABCD(str2{7,1})” are the transformation of string to numeric value of material grades A, B, C and D when if M10 and M11 have the same c3 or “if c3(1)==c3(2)”. The right machine can be defined by sorting of c3 with “c3=abs(c1-nin2); [c2,indc2]=sort(c3)”. The row number 1 of array indc2 will be selected. If the selected indc2 is row 1, program will show number of machine as 1 and plus with 5 to be 6. Thus, if the selected indc2 array is row 2, the selected machine will be 7 by follows “mach(ii)=indc2(1)+5”. But, if c3(1) is not equal to c3(2), the first row of array indc3 will be selected instead. And program will show number of machine as 6, that is calculated by row 1 plus with 5 to be 6, by follows “mach(ii)=indc3(1)+5”.

```
elseif MGr==3
in2=str(ii,5);
in3=str(ii,6);
nin2=str2numABCD(in2);
nin3=str2numcolor(in3);
         c2(1)=str2numcolor(str2{6,2});
         c2(2)=str2numcolor(str2{7,2});
         c3=abs(c2-nin3);
         [c3,indc3]=sort(c3);

 if c3(1)==c3(2)
         c1(1)=str2numABCD(str2{6,1});
         c1(2)=str2numABCD(str2{7,1});
         c3=abs(c1-nin2);
         [c2,indc2]=sort(c3);
         mach(ii)=indc2(1)+5;
 else
         mach(ii)=indc3(1)+5;
 end
```

In case of MGr is 4 and 5, there is just one machine in each group, so the target machine can be achieved immediately. After in2 and in 3 were defined by the scripts of " in2=str(ii,5)" and "in3=str(ii,6)", machine number 8 or M8 will be selected by "mach(ii)=8". And machine number 9 or M9 will be selected from the scripts of "mach(ii)=9" in case of MGr=5.

```
elseif MGr==4
          in2=str(ii,5);
          in3=str(ii,6);
          mach(ii)=8;

elseif MGr==5
          in2=str(ii,5);
          in3=str(ii,6);
          mach(ii)=9;
```

For the last machine group, MGr is 6, there are almost scripts same as MGr is 1 and 3. Except, the product color of previous job in machine M10 and M11 were defined to be c2(1) and c2(2) in numerical value (i.e. Clear is 1, White is 2, Grey is 3 and Black is 4) by functions of "c2(1)=str2numcolor(str2{10,2})" and "c2(2)=str2numcolor(str2{11,2})" respectively. The scripts of "c1(1)=str2numABCD(str2{10,1})" and "c1(2)=str2numABCD(str2{11,1})" are the transformation of string to numeric value of material grades A, B, C and D when if M10 and M11 have the same c3 or "if c3(1)==c3(2)". The right machine can be defined by sorting of c3 with "c3=abs(c1-nin2); [c2,indc2]=sort(c3)". The row number 1 of array indc2 will be selected. If the selected indc2 is row 1, program will show number of machine as 1 and plus with 9 to be 10. Thus, if the selected indc2 array is row 2, the selected machine will be 11 by follows "mach(ii)=indc2(1)+9". But, if c3(1) is not equal to c3(2), the first row of array indc3 will be selected instead. And program will show number of machine as 10, that is calculated by row 1 plus with 9 to be 10, by follows "mach(ii)=indc3(1)+9". The evaluation time of this program will stop at "toc" located in the end of these scripts.

```
elseif MGr==6
in2=str{ii,5};
```

```
in3=str(ii,6);
nin2=str2numABCD(in2);
nin3=str2numcolor(in3);
         c2(1)=str2numcolor(str2{10,2});
         c2(2)=str2numcolor(str2{11,2});

         c3=abs(c2-nin3);
         [c3,indc3]=sort(c3);
if c3(1)==c3(2)
         c1(1)=str2numABCD(str2{10,1});
         c1(2)=str2numABCD(str2{11,1});
         c3=abs(c1-nin2);
         [c2,indc2]=sort(c3);
         mach(ii)=indc2(1)+9;
 else
         mach(ii)=indc3(1)+9;
 end

end
toc
end
```

In order to provide the references of 11 machine names (M1 to M11), the set of namemach are used to refer the name of each machine in table3 or uitable4 by the scripts of “mach=mach'; namemach={'TOSHIBA IS30EPN-1Y'; 'JSW J30 ED'; 'JSW J50 EII'; 'NIGATA NN50E'; 'TOSHIBA IS50EP-1.5A'; 'NIIGATA'; 'NIIGATA'; 'NIIGATA'; 'NIIGATA'; 'JSW J55 EII'; 'NIIGATA'; 'NIIGATA';}”. In uitable4 or Result table, there are 4 columns which display the information of the results from calculation (Figure 4.33). The information in column 1, 2, 3 and 4 have defined by the scripts of “table3{ii,1}='The selected machine is'”, “table3{ii,2}= ['M' num2str(mach(ii))]”, “table3{ii,3}= namemach{mach(ii)}”, and “table3{ii,4}=group0(ii)” respectively. And the scripts of “set(handles.uitable4,'data',table3)” are the setting of table3 is uitable4.

```
mach=mach';
namemach={'TOSHIBA IS30EPN-1Y';
'JSW J30 ED';
```

```
'JSW  J50 EII';
'NIGATA NN50E';
'TOSHIBA IS50EP-1.5A';
'NIIGATA';
'NIIGATA';
'NIIGATA';
'JSW  J55 EII';
'NIIGATA';
'NIIGATA';
};
for ii=1:length(mach)
         table3{ii,1}='The selected machine is';
         table3{ii,2}=['M' num2str(mach(ii))];
         table3{ii,3}=namemach{mach(ii)};
         table3{ii,4}=group0(ii);
end
set(handles.uitable4,'data',table3)
close(h1)
```

Results

	1	2	3	4
1	The selected machine is	M1	TOSHIBA IS30EPN-1Y	1
2	The selected machine is	M1	TOSHIBA IS30EPN-1Y	1
3	The selected machine is	M3	JSW J50 EII	2
4	The selected machine is	M6	NIIGATA	3
5	The selected machine is	M8	NIIGATA	4
6	The selected machine is	M9	JSW J55 EII	5
7	The selected machine is	M9	JSW J55 EII	5
8	The selected machine is	M10	NIIGATA	6
9	The selected machine is	M10	NIIGATA	6
10	The selected machine is	M10	NIIGATA	6

Figure 4.33 Uitable4 or results table

4.2.3 Scripts of MCselection.fis file

After the creating of fuzzy inference system in 4.1.1.1 has performed, MATLAB will generate codes in MCselection.fis file as in Figure 4.34. Sets of script of [System] are the information of our fuzzy inference system which are the compositions of Name (file name), Type (fuzzy inference system style), Version, NumInputs (number of inputs), NumOutputs (number of output), NumRules (number of rules), AndMethod (method of ‘AND’ interaction) , OrMethod (method of ‘OR’ interaction), ImpMethod (method of implication), AggMethod (method of aggregation), DefuzzMethod (method of defuzzification).

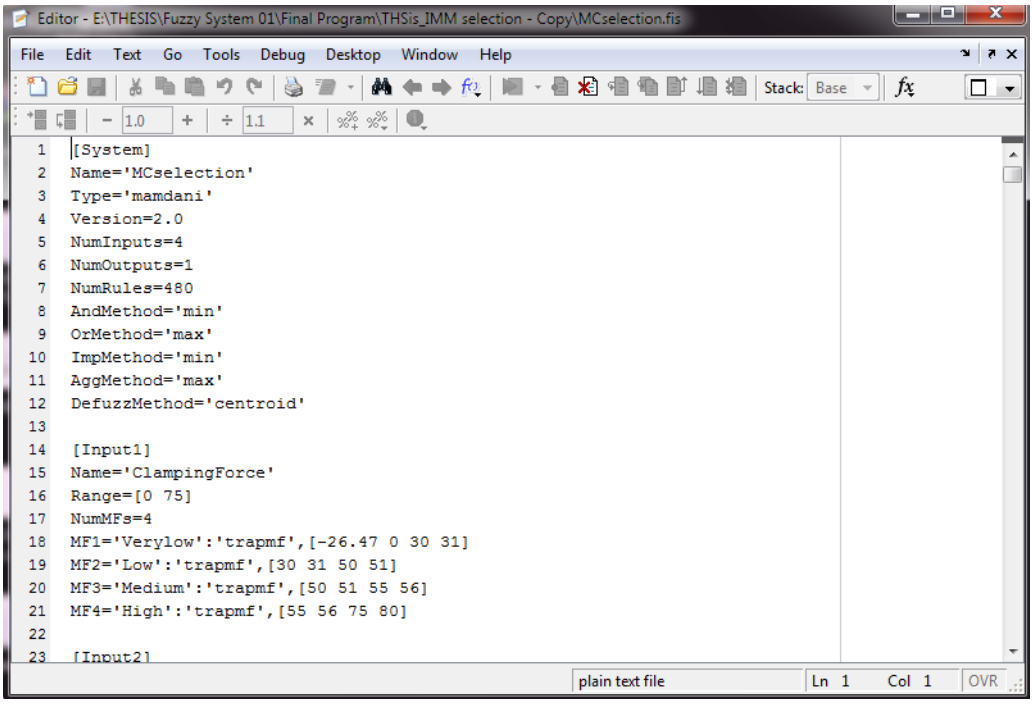


Figure 4.34 MCselection.fis in MATLAB editor

```
[System]
Name='MCselection'
Type='mamdani'
Version=2.0
NumInputs=4
NumOutputs=1
NumRules=480
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
```

```
DefuzzMethod='centroid'
```

This part contains sets of script of [Input1], or ClampingForce variable. which are included of Name (variable name), Range (range of input), NumMFs (number of membership functions). And the scripts of “MF1='Verylow':'trapmf',[-26.47 0 30 31]” are membership function name, membership function type, and input membership function params of MF1. by the same way to MF2, MF3, and MF4.

```
[Input1]
Name='ClampingForce'
Range=[0 75]
NumMFs=4
MF1='Verylow':'trapmf',[-26.47 0 30 31]
MF2='Low':'trapmf',[30 31 50 51]
MF3='Medium':'trapmf',[50 51 55 56]
MF4='High':'trapmf',[55 56 75 80]
```

This part contains sets of script of [Input2], or TiebarX variable. which are included of Name (variable name), Range (range of input), NumMFs (number of membership functions). And the scripts of “MF1='Verylow':'trapmf',[-5 0 260 261]” are membership function name, membership function type, and input membership function params of MF1. by the same way to MF2, MF3, and MF4.

```
[Input2]
Name='TiebarX'
Range=[0 350]
NumMFs=4
MF1='Verylow':'trapmf',[-5 0 260 261]
MF2='Low':'trapmf',[260 261 300 301]
MF3='Medium':'trapmf',[300 301 310 311]
MF4='High':'trapmf',[310 311 350 360]
```

This part contains sets of script of [Input3], or TiebarY variable. which are included of Name (variable name), Range (range of input), NumMFs (number of membership functions). And the scripts of “MF1='Verylow':'trapmf',[-5 0 260 261]” are membership function name, membership function type, and input membership function params of MF1. by the same way to MF2, MF3, MF4 and MF5.

```
[Input3]
Name='TiebarY'
Range=[0 380]
NumMFs=5
MF1='Verylow':'trapmf',[-5 0 260 261]
MF2='Low':'trapmf',[260 261 300 301]
MF3='Medium':'trapmf',[300 301 310 311]
MF4='High':'trapmf',[310 311 350 351]
MF5='Veryhigh':'trapmf',[350 351 380 400]
```

This part contains sets of script of [Input4], or Shotwt variable. which are included of Name (variable name), Range (range of input), NumMFs (number of membership functions). And the scripts of “MF1='Veryverylow':'trapmf',[-5 0 22 23]” are membership function name, membership function type, and input membership function params of MF1. by the same way to MF2, MF3, MF4, MF5 and MF6.

```
[Input4]
Name='Shotwt'
Range=[0 99]
NumMFs=6
MF1='Veryverylow':'trapmf',[-5 0 22 23]
MF2='Verylow':'trapmf',[22 23 27 28]
MF3='Low':'trapmf',[27 28 44 45]
MF4='Medium':'trapmf',[44 45 55 56]
MF5='High':'trapmf',[55 56 59 60]
MF6='Veryhigh':'trapmf',[59 60 99 105]
```

This part contains sets of script of [Output1], or MCgroup variable. which are included of Name (variable name), Range (range of input), NumMFs (number of membership functions). And the scripts of “MF1='G1':'trimf',[0 0.4998 1]” are membership function name, membership function type, and input membership function params of MF1. by the same way to MF2, MF3, MF4, and MF5.

```
[Output1]
Name='MCgroup'
Range=[0 6]
NumMFs=6
MF1='G1':'trimf',[0 0.4998 1]
MF2='G2':'trimf',[1 1.5 2]
MF3='G3':'trimf',[2 2.5 3]
MF4='G4':'trimf',[3 3.5 4]
MF5='G5':'trimf',[4 4.5 5]
MF6='G6':'trimf',[5 5.5 6]
```

These codes as below are codes of 384 fuzzy if-then rules which are used in MCselection.fis. The numbers in first 4 columns indicate the membership function number of each input variable. The number in column after "," indicates output variable or MCgroup. The number in "( )" indicates the interaction of each fuzzy if-then rules. Finally, the last number of rules are weight of rules.

```
[Rules]
1 1 1 1, 1 (1) : 1
1 1 1 2, 1 (1) : 1
1 1 1 3, 1 (1) : 1
1 1 1 4, 1 (1) : 1
1 1 1 5, 1 (1) : 1
1 1 1 6, 1 (1) : 1
1 1 2 1, 1 (1) : 1
1 1 2 2, 1 (1) : 1
1 1 2 3, 1 (1) : 1
1 1 2 4, 1 (1) : 1
1 1 2 5, 1 (1) : 1
1 1 2 6, 1 (1) : 1
1 1 3 1, 1 (1) : 1
1 1 3 2, 1 (1) : 1
1 1 3 3, 1 (1) : 1
1 1 3 4, 1 (1) : 1
1 1 3 5, 1 (1) : 1
1 1 3 6, 1 (1) : 1
1 1 4 1, 1 (1) : 1
1 1 4 2, 1 (1) : 1
1 1 4 3, 1 (1) : 1
1 1 4 4, 1 (1) : 1
1 1 4 5, 1 (1) : 1
1 1 4 6, 1 (1) : 1
1 2 1 1, 1 (1) : 1
1 2 1 2, 1 (1) : 1
1 2 1 3, 1 (1) : 1
1 2 1 4, 1 (1) : 1
1 2 1 5, 1 (1) : 1
1 2 1 6, 1 (1) : 1
```

```
1 2 2 1, 1 (1) : 1
1 2 2 2, 1 (1) : 1
1 2 2 3, 1 (1) : 1
1 2 2 4, 1 (1) : 1
1 2 2 5, 1 (1) : 1
1 2 2 6, 1 (1) : 1
1 2 3 1, 1 (1) : 1
1 2 3 2, 1 (1) : 1
1 2 3 3, 1 (1) : 1
1 2 3 4, 1 (1) : 1
1 2 3 5, 1 (1) : 1
1 2 3 6, 1 (1) : 1
1 2 4 1, 1 (1) : 1
1 2 4 2, 1 (1) : 1
1 2 4 3, 1 (1) : 1
1 2 4 4, 1 (1) : 1
1 2 4 5, 1 (1) : 1
1 2 4 6, 1 (1) : 1
1 3 1 1, 2 (1) : 1
1 3 1 2, 2 (1) : 1
1 3 1 3, 2 (1) : 1
1 3 1 4, 2 (1) : 1
1 3 1 5, 2 (1) : 1
1 3 1 6, 2 (1) : 1
1 3 2 1, 2 (1) : 1
1 3 2 2, 2 (1) : 1
1 3 2 3, 2 (1) : 1
1 3 2 4, 2 (1) : 1
1 3 2 5, 2 (1) : 1
1 3 2 6, 2 (1) : 1
1 3 3 1, 2 (1) : 1
1 3 3 2, 2 (1) : 1
1 3 3 3, 2 (1) : 1
1 3 3 4, 2 (1) : 1
1 3 3 5, 2 (1) : 1
1 3 3 6, 2 (1) : 1
1 3 4 1, 6 (1) : 1
1 3 4 2, 6 (1) : 1
1 3 4 3, 6 (1) : 1
1 3 4 4, 6 (1) : 1
1 3 4 5, 6 (1) : 1
1 3 4 6, 6 (1) : 1
1 4 1 1, 6 (1) : 1
1 4 1 2, 6 (1) : 1
1 4 1 3, 6 (1) : 1
1 4 1 4, 6 (1) : 1
1 4 1 5, 6 (1) : 1
1 4 1 6, 6 (1) : 1
1 4 2 1, 6 (1) : 1
1 4 2 2, 6 (1) : 1
1 4 2 3, 6 (1) : 1
1 4 2 4, 6 (1) : 1
1 4 2 5, 6 (1) : 1
1 4 2 6, 6 (1) : 1
1 4 3 1, 6 (1) : 1
1 4 3 2, 6 (1) : 1
1 4 3 3, 6 (1) : 1
1 4 3 4, 6 (1) : 1
```

```
1 4 3 5, 6 (1) : 1
1 4 3 6, 6 (1) : 1
1 4 4 1, 6 (1) : 1
1 4 4 2, 6 (1) : 1
1 4 4 3, 6 (1) : 1
1 4 4 4, 6 (1) : 1
1 4 4 5, 6 (1) : 1
1 4 4 6, 6 (1) : 1
2 1 1 1, 2 (1) : 1
2 1 1 2, 2 (1) : 1
2 1 1 3, 2 (1) : 1
2 1 1 4, 2 (1) : 1
2 1 1 5, 2 (1) : 1
2 1 1 6, 2 (1) : 1
2 1 2 1, 3 (1) : 1
2 1 2 2, 3 (1) : 1
2 1 2 3, 3 (1) : 1
2 1 2 4, 3 (1) : 1
2 1 2 5, 3 (1) : 1
2 1 2 6, 3 (1) : 1
2 1 3 1, 3 (1) : 1
2 1 3 2, 3 (1) : 1
2 1 3 3, 3 (1) : 1
2 1 3 4, 3 (1) : 1
2 1 3 5, 3 (1) : 1
2 1 3 6, 3 (1) : 1
2 1 4 1, 6 (1) : 1
2 1 4 2, 6 (1) : 1
2 1 4 3, 6 (1) : 1
2 1 4 4, 6 (1) : 1
2 1 4 5, 6 (1) : 1
2 1 4 6, 6 (1) : 1
2 2 1 1, 2 (1) : 1
2 2 1 2, 2 (1) : 1
2 2 1 3, 2 (1) : 1
2 2 1 4, 2 (1) : 1
2 2 1 5, 2 (1) : 1
2 2 1 6, 2 (1) : 1
2 2 2 1, 2 (1) : 1
2 2 2 2, 2 (1) : 1
2 2 2 3, 2 (1) : 1
2 2 2 4, 2 (1) : 1
2 2 2 5, 2 (1) : 1
2 2 2 6, 2 (1) : 1
2 2 3 1, 3 (1) : 1
2 2 3 2, 3 (1) : 1
2 2 3 3, 3 (1) : 1
2 2 3 4, 3 (1) : 1
2 2 3 5, 3 (1) : 1
2 2 3 6, 4 (1) : 1
2 2 4 1, 6 (1) : 1
2 2 4 2, 6 (1) : 1
2 2 4 3, 6 (1) : 1
2 2 4 4, 6 (1) : 1
2 2 4 5, 6 (1) : 1
2 2 4 6, 6 (1) : 1
2 3 1 1, 4 (1) : 1
2 3 1 2, 4 (1) : 1
```

```
2 3 1 3, 4 (1) : 1
2 3 1 4, 4 (1) : 1
2 3 1 5, 4 (1) : 1
2 3 1 6, 4 (1) : 1
2 3 2 1, 4 (1) : 1
2 3 2 2, 4 (1) : 1
2 3 2 3, 4 (1) : 1
2 3 2 4, 4 (1) : 1
2 3 2 5, 4 (1) : 1
2 3 2 6, 4 (1) : 1
2 3 3 1, 3 (1) : 1
2 3 3 2, 3 (1) : 1
2 3 3 3, 3 (1) : 1
2 3 3 4, 3 (1) : 1
2 3 3 5, 3 (1) : 1
2 3 3 6, 3 (1) : 1
2 3 4 1, 6 (1) : 1
2 3 4 2, 6 (1) : 1
2 3 4 3, 6 (1) : 1
2 3 4 4, 6 (1) : 1
2 3 4 5, 6 (1) : 1
2 3 4 6, 6 (1) : 1
2 4 1 1, 6 (1) : 1
2 4 1 2, 6 (1) : 1
2 4 1 3, 6 (1) : 1
2 4 1 4, 6 (1) : 1
2 4 1 5, 6 (1) : 1
2 4 1 6, 6 (1) : 1
2 4 2 1, 6 (1) : 1
2 4 2 2, 6 (1) : 1
2 4 2 3, 6 (1) : 1
2 4 2 4, 6 (1) : 1
2 4 2 5, 6 (1) : 1
2 4 2 6, 6 (1) : 1
2 4 3 1, 6 (1) : 1
2 4 3 2, 6 (1) : 1
2 4 3 3, 6 (1) : 1
2 4 3 4, 6 (1) : 1
2 4 3 5, 6 (1) : 1
2 4 3 6, 6 (1) : 1
2 4 4 1, 6 (1) : 1
2 4 4 2, 6 (1) : 1
2 4 4 3, 6 (1) : 1
2 4 4 4, 6 (1) : 1
2 4 4 5, 6 (1) : 1
2 4 4 6, 6 (1) : 1
3 1 1 1, 5 (1) : 1
3 1 1 2, 5 (1) : 1
3 1 1 3, 5 (1) : 1
3 1 1 4, 5 (1) : 1
3 1 1 5, 5 (1) : 1
3 1 1 6, 5 (1) : 1
3 1 2 1, 5 (1) : 1
3 1 2 2, 5 (1) : 1
3 1 2 3, 5 (1) : 1
3 1 2 4, 5 (1) : 1
3 1 2 5, 5 (1) : 1
3 1 2 6, 5 (1) : 1
```

```
3 1 3 1, 5 (1) : 1
3 1 3 2, 5 (1) : 1
3 1 3 3, 5 (1) : 1
3 1 3 4, 5 (1) : 1
3 1 3 5, 5 (1) : 1
3 1 3 6, 5 (1) : 1
3 1 4 1, 5 (1) : 1
3 1 4 2, 5 (1) : 1
3 1 4 3, 5 (1) : 1
3 1 4 4, 5 (1) : 1
3 1 4 5, 5 (1) : 1
3 1 4 6, 5 (1) : 1
3 2 1 1, 5 (1) : 1
3 2 1 2, 5 (1) : 1
3 2 1 3, 5 (1) : 1
3 2 1 4, 5 (1) : 1
3 2 1 5, 5 (1) : 1
3 2 1 6, 5 (1) : 1
3 2 2 1, 5 (1) : 1
3 2 2 2, 5 (1) : 1
3 2 2 3, 5 (1) : 1
3 2 2 4, 5 (1) : 1
3 2 2 5, 5 (1) : 1
3 2 2 6, 5 (1) : 1
3 2 3 1, 5 (1) : 1
3 2 3 2, 5 (1) : 1
3 2 3 3, 5 (1) : 1
3 2 3 4, 5 (1) : 1
3 2 3 5, 5 (1) : 1
3 2 3 6, 5 (1) : 1
3 2 4 1, 6 (1) : 1
3 2 4 2, 6 (1) : 1
3 2 4 3, 6 (1) : 1
3 2 4 4, 6 (1) : 1
3 2 4 5, 6 (1) : 1
3 2 4 6, 6 (1) : 1
3 3 1 1, 5 (1) : 1
3 3 1 2, 5 (1) : 1
3 3 1 3, 5 (1) : 1
3 3 1 4, 5 (1) : 1
3 3 1 5, 5 (1) : 1
3 3 1 6, 5 (1) : 1
3 3 2 1, 5 (1) : 1
3 3 2 2, 5 (1) : 1
3 3 2 3, 5 (1) : 1
3 3 2 4, 5 (1) : 1
3 3 2 5, 5 (1) : 1
3 3 2 6, 5 (1) : 1
3 3 3 1, 5 (1) : 1
3 3 3 2, 5 (1) : 1
3 3 3 3, 5 (1) : 1
3 3 3 4, 5 (1) : 1
3 3 3 5, 5 (1) : 1
3 3 3 6, 5 (1) : 1
3 3 4 1, 6 (1) : 1
3 3 4 2, 6 (1) : 1
3 3 4 3, 6 (1) : 1
3 3 4 4, 6 (1) : 1
```

```
3 3 4 5, 6 (1) : 1
3 3 4 6, 6 (1) : 1
3 4 1 1, 6 (1) : 1
3 4 1 2, 6 (1) : 1
3 4 1 3, 6 (1) : 1
3 4 1 4, 6 (1) : 1
3 4 1 5, 6 (1) : 1
3 4 1 6, 6 (1) : 1
3 4 2 1, 6 (1) : 1
3 4 2 2, 6 (1) : 1
3 4 2 3, 6 (1) : 1
3 4 2 4, 6 (1) : 1
3 4 2 5, 6 (1) : 1
3 4 2 6, 6 (1) : 1
3 4 3 1, 6 (1) : 1
3 4 3 2, 6 (1) : 1
3 4 3 3, 6 (1) : 1
3 4 3 4, 6 (1) : 1
3 4 3 5, 6 (1) : 1
3 4 3 6, 6 (1) : 1
3 4 4 1, 6 (1) : 1
3 4 4 2, 6 (1) : 1
3 4 4 3, 6 (1) : 1
3 4 4 4, 6 (1) : 1
3 4 4 5, 6 (1) : 1
3 4 4 6, 6 (1) : 1
4 1 1 1, 6 (1) : 1
4 1 1 2, 6 (1) : 1
4 1 1 3, 6 (1) : 1
4 1 1 4, 6 (1) : 1
4 1 1 5, 6 (1) : 1
4 1 1 6, 6 (1) : 1
4 1 2 1, 6 (1) : 1
4 1 2 2, 6 (1) : 1
4 1 2 3, 6 (1) : 1
4 1 2 4, 6 (1) : 1
4 1 2 5, 6 (1) : 1
4 1 2 6, 6 (1) : 1
4 1 3 1, 6 (1) : 1
4 1 3 2, 6 (1) : 1
4 1 3 3, 6 (1) : 1
4 1 3 4, 6 (1) : 1
4 1 3 5, 6 (1) : 1
4 1 3 6, 6 (1) : 1
4 1 4 1, 6 (1) : 1
4 1 4 2, 6 (1) : 1
4 1 4 3, 6 (1) : 1
4 1 4 4, 6 (1) : 1
4 1 4 5, 6 (1) : 1
4 1 4 6, 6 (1) : 1
4 2 1 1, 6 (1) : 1
4 2 1 2, 6 (1) : 1
4 2 1 3, 6 (1) : 1
4 2 1 4, 6 (1) : 1
4 2 1 5, 6 (1) : 1
4 2 1 6, 6 (1) : 1
4 2 2 1, 6 (1) : 1
4 2 2 2, 6 (1) : 1
```

```
4 2 2 3, 6 (1) : 1
4 2 2 4, 6 (1) : 1
4 2 2 5, 6 (1) : 1
4 2 2 6, 6 (1) : 1
4 2 3 1, 6 (1) : 1
4 2 3 2, 6 (1) : 1
4 2 3 3, 6 (1) : 1
4 2 3 4, 6 (1) : 1
4 2 3 5, 6 (1) : 1
4 2 3 6, 6 (1) : 1
4 2 4 1, 6 (1) : 1
4 2 4 2, 6 (1) : 1
4 2 4 3, 6 (1) : 1
4 2 4 4, 6 (1) : 1
4 2 4 5, 6 (1) : 1
4 2 4 6, 6 (1) : 1
4 3 1 1, 6 (1) : 1
4 3 1 2, 6 (1) : 1
4 3 1 3, 6 (1) : 1
4 3 1 4, 6 (1) : 1
4 3 1 5, 6 (1) : 1
4 3 1 6, 6 (1) : 1
4 3 2 1, 6 (1) : 1
4 3 2 2, 6 (1) : 1
4 3 2 3, 6 (1) : 1
4 3 2 4, 6 (1) : 1
4 3 2 5, 6 (1) : 1
4 3 2 6, 6 (1) : 1
4 3 3 1, 6 (1) : 1
4 3 3 2, 6 (1) : 1
4 3 3 3, 6 (1) : 1
4 3 3 4, 6 (1) : 1
4 3 3 5, 6 (1) : 1
4 3 3 6, 6 (1) : 1
4 3 4 1, 6 (1) : 1
4 3 4 2, 6 (1) : 1
4 3 4 3, 6 (1) : 1
4 3 4 4, 6 (1) : 1
4 3 4 5, 6 (1) : 1
4 3 4 6, 6 (1) : 1
4 4 1 1, 6 (1) : 1
4 4 1 2, 6 (1) : 1
4 4 1 3, 6 (1) : 1
4 4 1 4, 6 (1) : 1
4 4 1 5, 6 (1) : 1
4 4 1 6, 6 (1) : 1
4 4 2 1, 6 (1) : 1
4 4 2 2, 6 (1) : 1
4 4 2 3, 6 (1) : 1
4 4 2 4, 6 (1) : 1
4 4 2 5, 6 (1) : 1
4 4 2 6, 6 (1) : 1
4 4 3 1, 6 (1) : 1
4 4 3 2, 6 (1) : 1
4 4 3 3, 6 (1) : 1
4 4 3 4, 6 (1) : 1
4 4 3 5, 6 (1) : 1
4 4 3 6, 6 (1) : 1
```

```
4 4 4 1, 6 (1) : 1
4 4 4 2, 6 (1) : 1
4 4 4 3, 6 (1) : 1
4 4 4 4, 6 (1) : 1
4 4 4 5, 6 (1) : 1
4 4 4 6, 6 (1) : 1
```

4.2.4 Scripts of static text objects

All objects of static text that have shown on the program screen have their scripts. User can changes the properties about text background color, text colors or all him/her needs by editing these scripts. The example of text10 static text object is here as follows:

```
% --- Executes during object creation, after setting all properties.
function text10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called
```

4.3 Injection molding machine selection system user guide

The procedures of using IMM_Selection02.m are as follows:

Step 1: Open MATLAB2010a program. Double-click to select file name “IMM_Selection02.m” in Current Folder window. And click  to run IMM_Selection02.m file as in Figure 4.35.

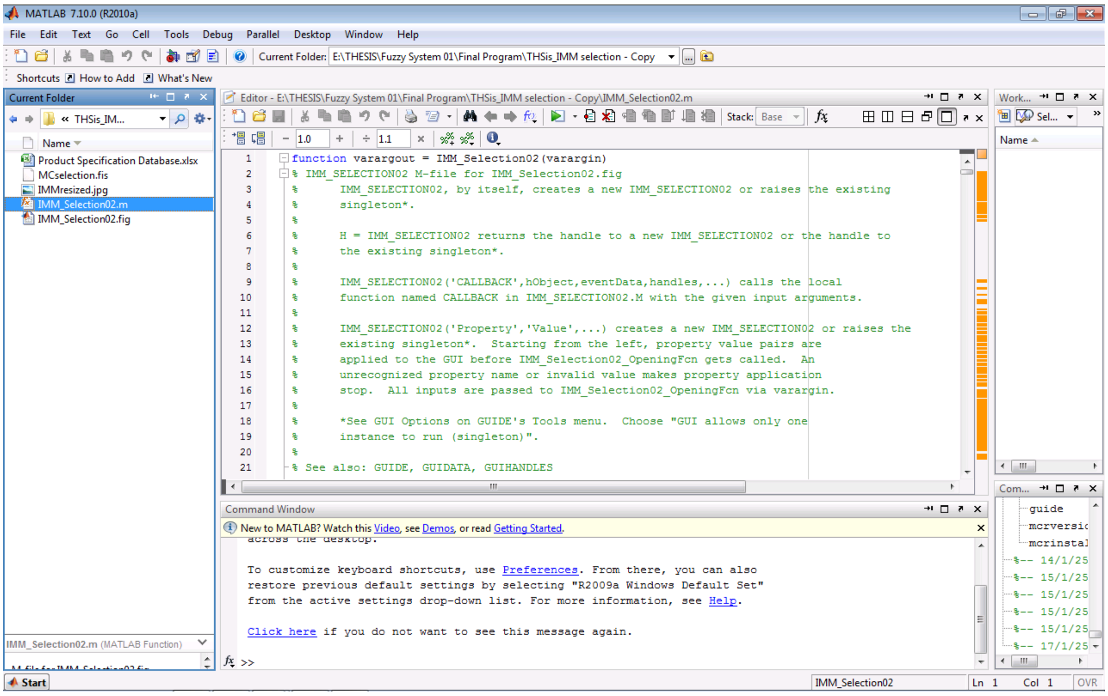


Figure 4.35 Open file IMM_Selection02.m

Step 2: MATLAB shows the main window that is the user interface of the proposed program.

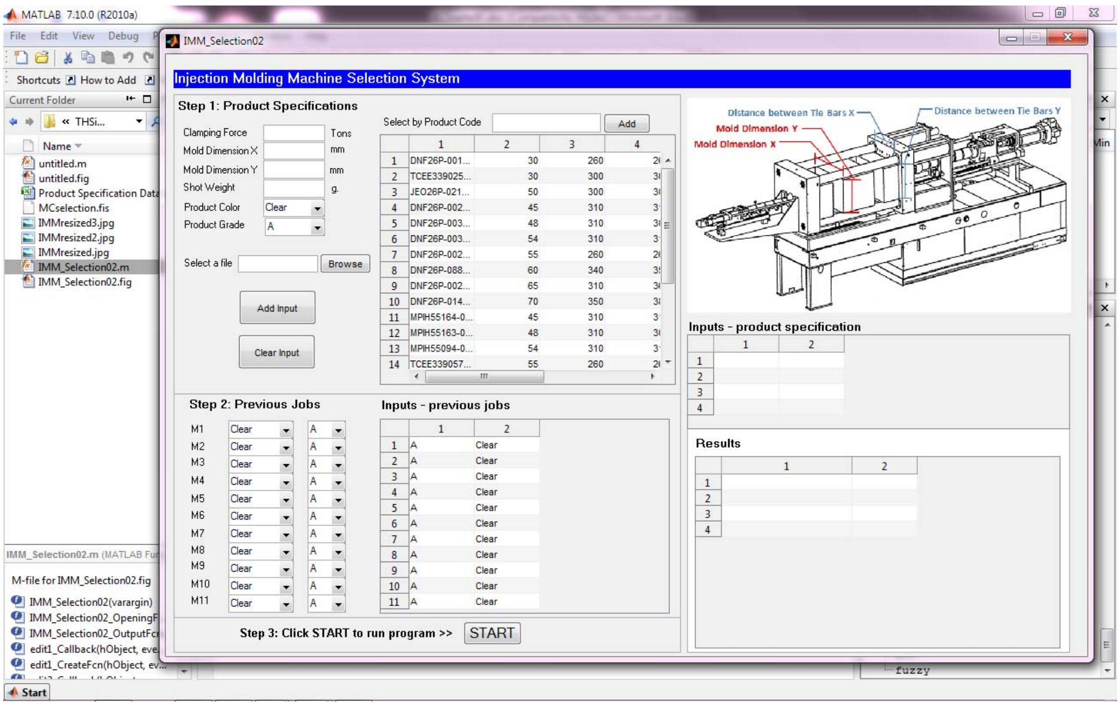
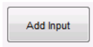


Figure 4.36 IMM_Selection02.m main window

Step 3: Fills all product specifications into the program. There are three methods for filling out the information that are:

1) Type all input data. For example, type data of Clamping Force of product as 40 tons, Mold Dimension X and Y as 290 and 310 mm, Shot Weight as 30 g, Product Color as Grey and Product Grade as A as in Figure 4.37. Then, push 

button. All input variables will be shown in table of Inputs – product specification as in Figure 4.38.

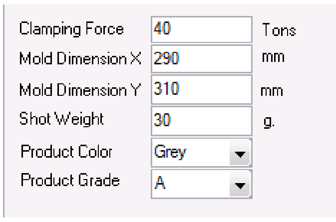


Figure 4.37 Input variables

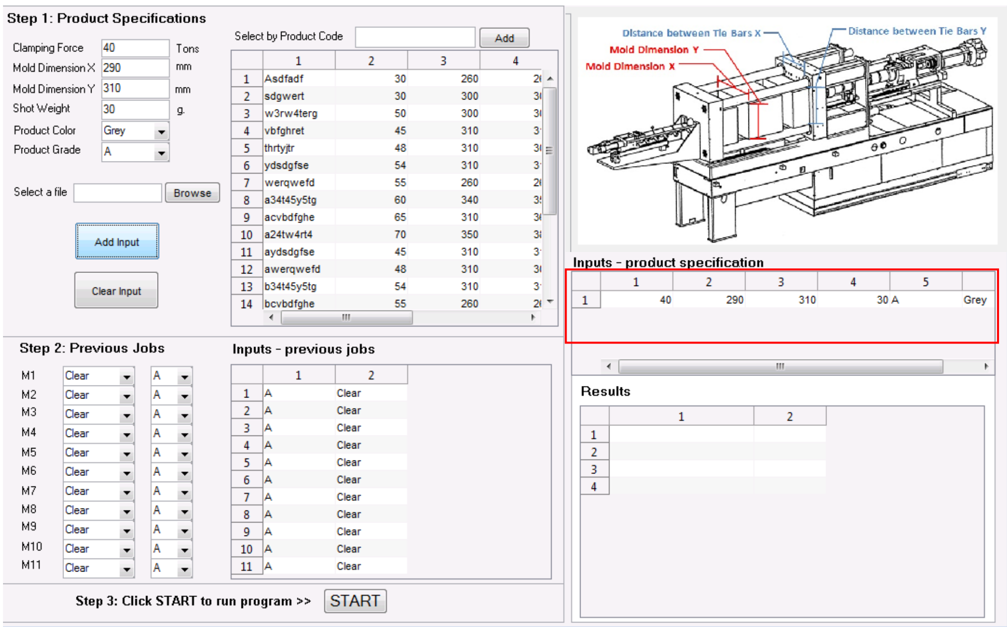


Figure 4.38 Input data from filling out the form

2) Upload product specification data file (.xlsx). This file contains data of Clamping Force, Mold Dimension X, Mold Dimension Y, Product Color and Product Grade like method 1). For example, click ‘Browse’, select file names “data.xlsx”. Then, click ‘Open’. All data within this file will be displayed in table of Inputs – product specification. Please see Figure 4.39 and 4.40.

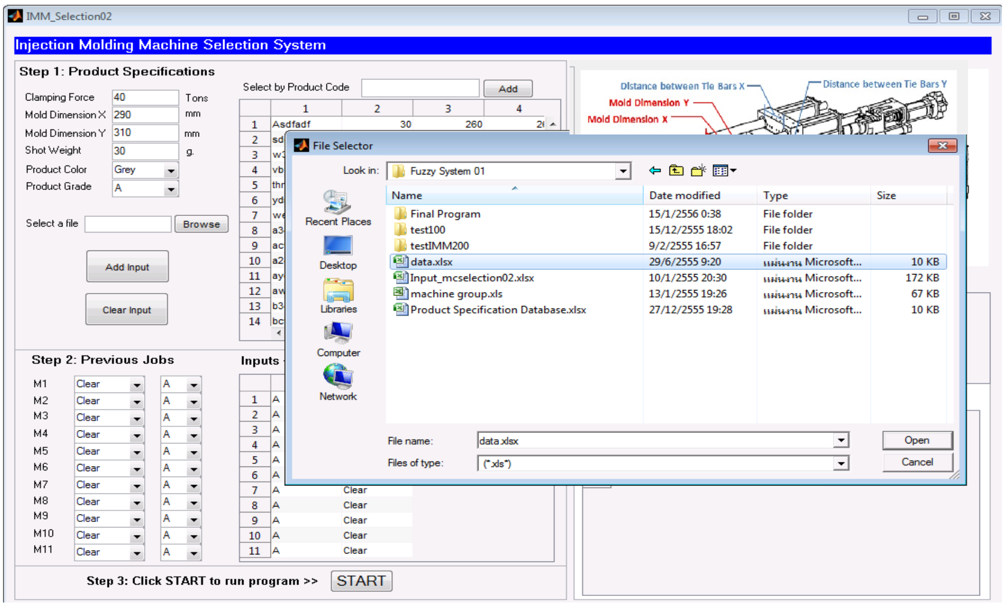


Figure 4.39 Open "data.xlsx" file

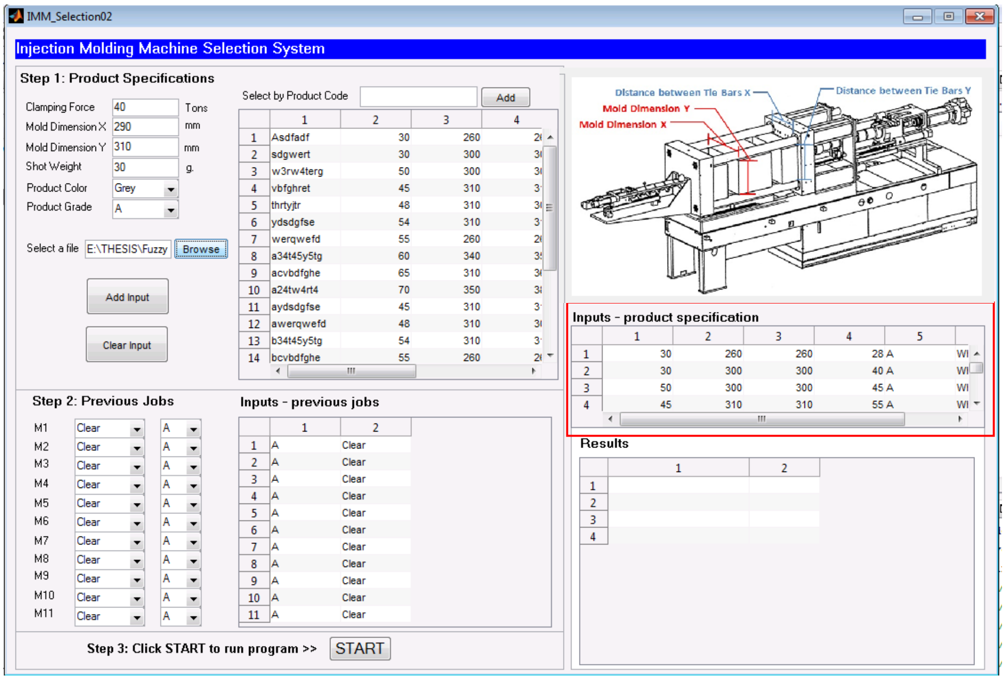


Figure 4.40 Input data from inserting data file

3) Type the name of required product code. If user cannot remember that name, the data in the bellowed table is the references. The data in table are composed of Clamping Force, Mold Dimension X, Mold Dimension Y, Product Color and Product Grade same as method 1) and 2). After that, push 'Add' button, data of the

selected product code will be shown in table of Inputs – product specification. Please see Figure 4.41 and 4.42.

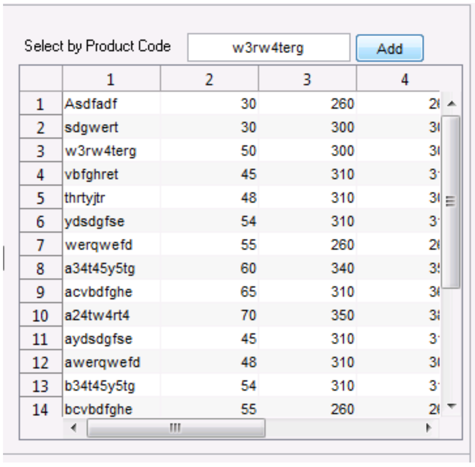


Figure 4.41 Product specification database

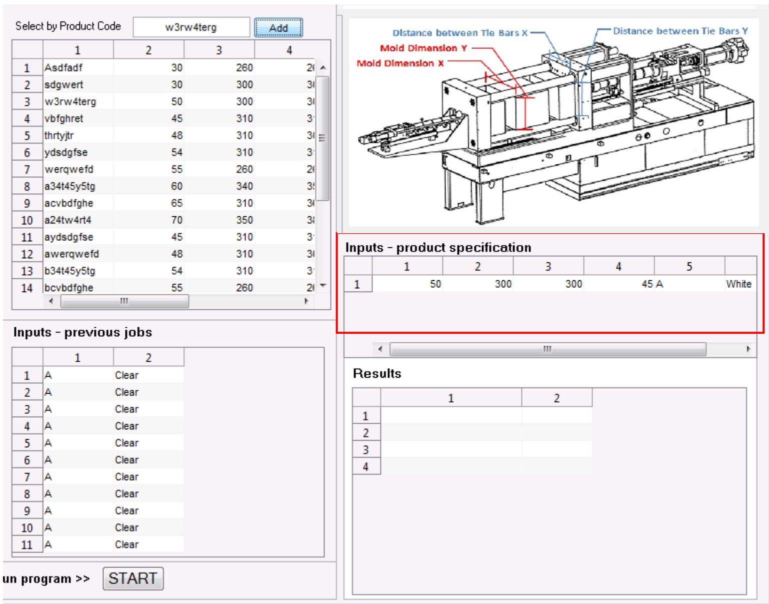


Figure 4.42 Input data from searching product code

Step 4: If user needs to delete or modify input data in the table of Inputs – product specification, click ‘Clear Input’ (Figure 4.43) to delete all data in that table. Please be reminded that user cannot modify data particularly.

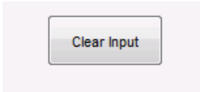


Figure 4.43 Clear Input button

Step 5: Set the value of Product Color and Product Grade of the latest jobs of 11 machines. These data will appear in Inputs – previous jobs immediately. Validate all input variables. Please see Figure 4.44.

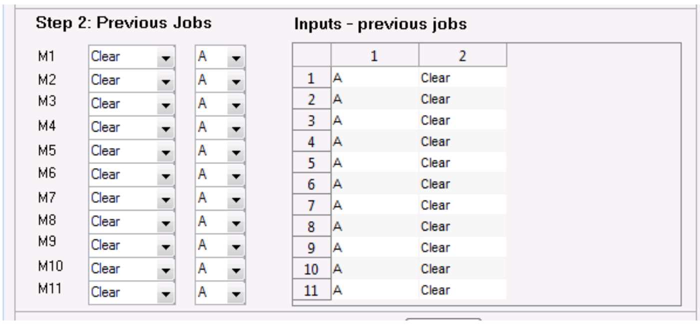


Figure 4.44 Table of Inputs - previous jobs

Step 6: Click 'START' button as in Figure 4.45 to run the program.

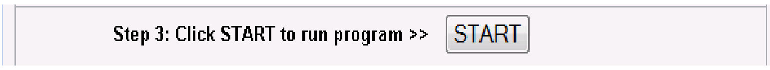


Figure 4.45 START button

Step 7: The selected injection molding machine will be displayed in table of Results. For example, the selected machine is M3 names JSW J50 Ell which is classified in machine group 2 as in Figure 4.46, mean the product code "w3rw4terg" should be assigned to machine M3.

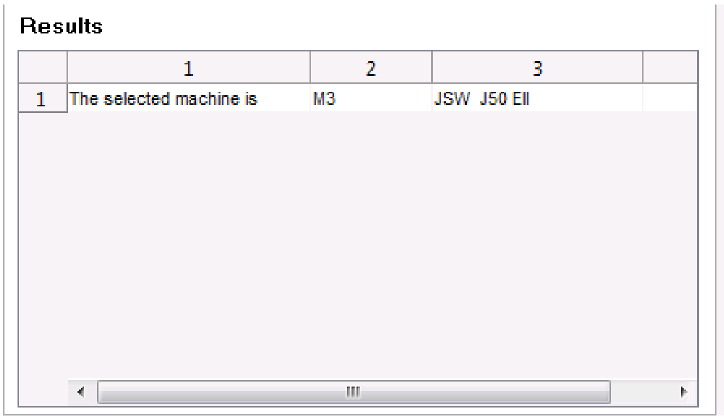


Figure 4.46 Results table

4.4 Validation and verification of the proposed system

The end of the development process, validation and verification (V&V) of the proposed system, named as Injection Molding Machine Selection System, are necessary. According to IEEE Std 1012-2004, validation is the evaluation process to determine whether our system satisfies user requirements. And verification is the process of evaluating the system to determine whether this developed system satisfies the required conditions.

4.2.1 Objective of V&V

The objectives of V&V are the discovering errors or mistakes via early detection and correction of software. And, the ability of implementation in actual production should be assessed to enhance process and product risk management.

4.2.2 V&V technique

The process of V&V is more powerful by applying of the combination static testing and dynamic testing to verify and analyze the validity of system. Static testing is concerned with demonstration and analysis the concerned documents during system development such as user requirement, development diagram and system prototype. Dynamic testing is system testing with test data to identify errors.

4.2.3 System testing by test cases

This defect testing aims to determine all errors and show the program meets its specification with repeatable, systematic and recordable. Random testing is not enough. Hence, the selecting of test sets that cover the range of behaviors of the program and able to representative of real uses is good strategy. The defect testing process can be divided to four steps as follows:

1) Design test cases – To design both input and output data for testing.

2) Prepare test data – To prepare data that concerns driving of program.

3) Run program with test data – To simulate using of program.

4) Compare results to test cases – To verify that program meets its specification.

Finally, this study has prepared checklists that conformed to verification and validation standard. All checklists are shown in Table 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 and 4.7. These checklists covered all operations; start to end of using program. And the examples of test cases are shown in Table 4.8, 4.9, 4.10, 4.11, 4.12, 4.13 and 4.14.

Table 4.1 Test case of the opening of IMM_Selection02.m

Test Case ID	TCIMMS_Funct_01		
Description	Test the opening of IMM_Selection02.m		
Applicable for	MATLAB R2010a, Microsoft Excel 2007		
Requirement	-		
Initial Conditions	MATLAB R2010a have installed on Microsoft Windows.		

Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
1	Open the MATLAB R2010a	Pass	-
2	Verify that MATLAB Current Folder window is showing the folder that program is kept.	Pass	-
3	Verify that the current folder has an m-file names IMM_Selection02.m	Pass	-
4	Verify that the current folder has an m-file names IMM_Selection02.fig	Pass	-
5	Verify that the current folder has an m-file names MCselection.fis	Pass	-
6	Verify that the current folder has an m-file names IMMresized03.jpg	Pass	-
7	Verify that the current folder has an m-file names Product Specification Database.xlsx	Pass	-
8	After double-clicking at IMM_Selection02.m is applied, verify that IMM_Selection02.m is displayed in Editor window.	Pass	-
9	Verify that interface of IMM_Selection02 is displayed after clicking at run button without errors.	Pass	-

Table 4.2 Test case of the interface screen and default values

Test Case ID	TCIMMS_Funct_02		
Description	Test the interface screen and default values		
Applicable for	MATLAB R2010a, Microsoft Excel 2007		
Requirement	-		
Initial Conditions	The IMM_selection02.m can be opened smoothly.		
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
1	Verify that Clamping Force, Mold Dimension X, Mold Dimension Y and Shot Weight are empty.	Pass	-
2	Verify that default value of Product Color is Clear.	Pass	-
3	Verify that default value of Product Grade is A.	Pass	-
4	Verify that Select a file box is empty and Browse button is displayed.	Pass	-
5	Verify that Add Input and Clear Input below the Select a file box are displayed.	Pass	-
6	Verify that Select by Product Code box is empty and Add button beside the Product Code box is displayed.	Pass	-
7	Verify that table below the Product Code box is displayed the data from Product Specification Database.xlsx	Pass	-
8	Verify that the picture names IMMresized03 is displayed over the table of Inputs-product specification.	Pass	-
9	Verify that the table of Inputs-product specification is empty.	Pass	-

Table 4.2 Test case of the interface screen and default values (Cont.)

Test Case ID	TCIMMS_Funct_02		
Description	Test the interface screen and default values		
Applicable for	MATLAB R2010a, Microsoft Excel 2007		
Requirement	-		
Initial Conditions	The IMM_selection02.m can be opened smoothly.		
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
10	Verify that default values of Previous Jobs for M1 to M11 are displayed with Clear and A.	Pass	-
11	Verify that data in table of Inputs-previous jobs are A and Clear.	Pass	-
12	Verify that START button is displayed below the table of Inputs-previous jobs.	Pass	-
13	Verify that the table of Results is empty.	Pass	-

Table 4.3 Test case of the operations of entering data

Test Case ID	TCIMMS_Funct_03		
Description	Test the operations of entering data		
Applicable for	MATLAB R2010a, Microsoft Excel 2007		
Requirement	-		
Initial Conditions	The system displays the right default values and interface.		
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
1	Verify that the boxes of Clamping Force, Mold Dimension X, Mold Dimension Y and Shot Weight can be entered.	Pass	-
2	Verify that “Please input Clamping Force between 30 to 75” message box will appear if user enters value less than 30 into the Clamping Force box.	Pass	-
3	Verify that “Please input Clamping Force between 30 to 75” message box will appear if user enters value more than 75 into the Clamping Force box.	Pass	-
4	Verify that “Please input Clamping Force between 30 to 75” message box will appear if user lefts Clamping Force box in blank.	Pass	-
5	Verify that “Please input Mold Dimension X between 260 to 350” message box will appear if user enters value less than 260 into the Mold Dimension X box.	Pass	-

Table 4.3 Test case of the operations of entering data (Cont.)

Test Case ID	TCIMMS_Funct_03		
Description	Test the operations of entering data		
Applicable for	MATLAB R2010a, Microsoft Excel 2007		
Requirement	-		
Initial Conditions	The system displays the right default values and interface.		
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
6	Verify that “Please input Mold Dimension X between 260 to 350” message box will appear if user enters value more than 350 into the Mold Dimension X box.	Pass	-
7	Verify that “Please input Mold Dimension X 30 to 75” message box will appear if user lefts Mold Dimension X box in blank.	Pass	-
6	Verify that “Please input Mold Dimension Y between 260 to 380” message box will appear if user enters value less than 260 into the Mold Dimension Y box.	Pass	-
7	Verify that “Please input Mold Dimension Y between 260 to 380” message box will appear if user enters value more than 380 into the Mold Dimension Y box.	Pass	-
8	Verify that “Please input Mold Dimension Y 260 to 380” message box will appear if user lefts Mold Dimension Y box in blank.	Pass	-

Table 4.3 Test case of the operations of entering data (Cont.)

Test Case ID	TCIMMS_Funct_03		
Description	Test the operations of entering data		
Applicable for	MATLAB R2010a, Microsoft Excel 2007		
Requirement	-		
Initial Conditions	The system displays the right default values and interface.		
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
9	Verify that “Please input Shot Weight between 1 to 100” message box will appear if user enters value less than 1 into the Shot Weight box.	Pass	-
10	Verify that “Please input Shot Weight between 1 to 100” message box will appear if user enters value more than 100 into the Shot Weight box.	Pass	-
11	Verify that “Please input Shot Weight between 1 to 100” message box will appear if user lefts Shot Weight box in blank.	Pass	-
12	Verify that the drop boxes of Product Color and Product Grade can be selected the choices.	Pass	-
13	Verify that Add Input button can be clicked.	Pass	-
14	Verify that clicking of Add Input button can add all data of Clamping Force, Mold Dimension X, Mold Dimension Y, Shot Weight, Product Color and Product Grade to the Inputs-product specification table.	Pass	-
15	Verify that Clear Input button can be clicked.	Pass	-

Table 4.3 Test case of the operations of entering data (Cont.)

Test Case ID		TCIMMS_Funct_03	
Description		Test the operations of entering data	
Applicable for		MATLAB R2010a, Microsoft Excel 2007	
Requirement		-	
Initial Conditions		The system displays the right default values and interface.	
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
16	Verify that the clicking of Clear Input button can remove all data in the Inputs-product specification table.	Pass	-
17	Verify that data that displayed in Inputs-product specification table have 6 columns.	Pass	-

Table 4.4 Test case of the operations of entering data from file

Test Case ID		TCIMMS_Funct_04	
Description		Test the operations of entering data from file	
Applicable for		MATLAB R2010a, Microsoft Excel 2007	
Requirement		-	
Initial Conditions		The target file is located in any place of hard drive. The target file is .xlsx only.	
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
1	Verify that Browse button can be clicked to select the file named “data.xlsx”.	Pass	-
2	Verify that the required file from Browse button is .xlsx file only.	Pass	-

Table 4.4 Test case of the operations of entering data from file (Cont.)

Test Case ID		TCIMMS_Funct_04	
Description		Test the operations of entering data from file	
Applicable for		MATLAB R2010a, Microsoft Excel 2007	
Requirement		-	
Initial Conditions		The target file is located in hard drive and is .xlsx only.	
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
3	Verify that Add Input button can be clicked.	Pass	-
4	Verify that clicking of Add Input button can add all data from data.xlsx file to the Inputs-product specification table.	Pass	-
5	Verify that Clear Input button can be clicked.	Pass	-
6	Verify that the clicking of Clear Input button can remove all data in the Inputs-product specification table.	Pass	-
7	Verify that data that displayed in data.xlsx have 6 columns.	Pass	-
8	Verify that the first column of data.xlsx is Clamping Force.	Pass	-
9	Verify that the first column of data.xlsx is Mold Dimension X.	Pass	-
10	Verify that the first column of data.xlsx is Mold Dimension Y.	Pass	-
11	Verify that the first column of data.xlsx is Shot Weight.	Pass	-
12	Verify that the first column of data.xlsx is Product Grade.	Pass	-
13	Verify that the first column data.xlsx is Product Color.	Pass	-

Table 4.5 Test case of the operations of entering data by searching from database

<table>
<tr><td colspan="2">Test Case ID</td><td colspan="3">TCIMMS_Funct_05</td></tr>
<tr><td colspan="2">Description</td><td colspan="3">Test the operations of entering data by searching from database</td></tr>
<tr><td colspan="2">Applicable for</td><td colspan="3">MATLAB R2010a, Microsoft Excel 2007</td></tr>
<tr><td colspan="2">Requirement</td><td colspan="3">-</td></tr>
<tr><td colspan="2">Initial Conditions</td><td colspan="3">The database file is located in the same folder of IMM_selection02.m. The database file is .xlsx only.</td></tr>
<tr><td>Step</td><td>Task & Expected Result</td><td>Actual Result</td><td colspan="2">Detail
(If Expected Result and Actual Result are different.)</td></tr>
<tr><td>1</td><td>Verify that the latest version of Product Specification Database.xlsx is displayed in table.</td><td>Pass</td><td colspan="2">-</td></tr>
<tr><td>2</td><td>Verify that data that displayed in Inputs-product specification table have 7 columns.</td><td>Pass</td><td colspan="2">-</td></tr>
<tr><td>3</td><td>Verify that the first column of Product Specification Database table is Product Code.</td><td>Pass</td><td colspan="2">-</td></tr>
<tr><td>4</td><td>Verify that the second column of Product Specification Database table is Clamping Force.</td><td>Pass</td><td colspan="2">-</td></tr>
<tr><td>5</td><td>Verify that the third column of Product Specification Database table is Mold Dimension X.</td><td>Pass</td><td colspan="2">-</td></tr>
<tr><td>6</td><td>Verify that the fourth column of Product Specification Database table is Mold Dimension Y.</td><td>Pass</td><td colspan="2">-</td></tr>
<tr><td>7</td><td>Verify that the fifth column of Product Specification Database table is Shot Weight.</td><td>Pass</td><td colspan="2">-</td></tr>
<tr><td>8</td><td>Verify that the sixth column of Product Specification Database table is Product Grade.</td><td>Pass</td><td colspan="2">-</td></tr>
</table>

Table 4.5 Test case of the operations of entering data by searching from database (Cont.)

Test Case ID	TCIMMS_Funct_05		
Description	Test the operations of entering data by searching from database		
Applicable for	MATLAB R2010a, Microsoft Excel 2007		
Requirement	-		
Initial Conditions	The database file is located in the same folder of IMM_selection02.m. The database file is .xlsx only.		
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
9	Verify that the seven column of Product Specification Database table is Product Color.	Pass	-
10	Verify that Product Code can be copy and paste in the box of Search by Product Code.	Pass	-

Table 4.5 Test case of the operations of entering data by searching from database (Cont.)

Test Case ID	TCIMMS_Funct_05		
Description	Test the operations of entering data by searching from database		
Applicable for	MATLAB R2010a, Microsoft Excel 2007		
Requirement	-		
Initial Conditions	The database file is located in the same folder of IMM_selection02.m. The database file is .xlsx only.		
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
11	Verify that all data of Clamping Force, Mold Dimension X, Mold Dimension Y, Product Grade and Product Color will be display after clicking ‘Add’.	Pass	-
12	Verify that data that displayed in Inputs-product specification table have 7 columns.	Pass	-
13	Verify that the first column of Inputs-product specification table is Clamping Force.	Pass	-
14	Verify that the second column of Inputs-product specification table is Mold Dimension X.	Pass	-
15	Verify that the third column of Inputs-product specification table is Mold Dimension Y.	Pass	-
16	Verify that the fourth column of Inputs-product specification table is Shot Weight.	Pass	-
17	Verify that the fifth column of Inputs-product specification table is Product Grade.	Pass	-
18	Verify that the sixth column of Inputs-product specification table is Product Color.	Pass	-

Table 4.6 Test case of the entering data of previous jobs

Test Case ID	TCIMMS_Funct_06		
Description	Test the entering data of previous jobs		
Applicable for	MATLAB R2010a, Microsoft Excel 2007		
Requirement	-		
Initial Conditions	The system only shows the default previous job data of Product Color and Product Grade for all machine.		
Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
1	Verify that the data in table of Inputs-previous jobs are changed immediately by following Product Grade and Product Color value on left side.	Pass	-
2	Verify that the first column in table of Inputs-previous jobs is Product Grade.	Pass	-
3	Verify that the second column in table of Inputs-previous jobs is Product Color.	Pass	-
4	Verify that there are four choice of A, B, C and D in Product Grade.	Pass	-
5	Verify that there are four choices of Clear, White, Grey and Black in Product Color	Pass	-

Table 4.7 Test case of the running of IMM_Selection02.m

Test Case ID	TCIMMS_Funct_07		
Description	Test the running of IMM_Selection02.m		
Applicable for	MATLAB R2010a, Microsoft Excel 2007		
Requirement	-		
Initial Conditions	There is not any data in the table of Results.		

Step	Task & Expected Result	Actual Result	Detail (If Expected Result and Actual Result are different.)
1	Verify that table of Inputs-product specification is added enough data before running.	Pass	-
2	Verify that "Please wait…" message box will be displayed after clicking of START button.	Pass	-
3	Verify that system will not display any result in Results table if there is not any data in the table of Inputs-product specification.	Pass	-
4	Verify that system runs smoothly and does not return any error message in command window.	Pass	-
5	Verify that the first column of Results table displays as "The selected machine is".	Pass	-
6	Verify that the second column of Results table is the selected machine (M1 to M11).	Pass	-
7	Verify that the third column of Results table is the name of selected machine.	Pass	-
8	Verify that the fourth column of Results table is the number of machine group.	Pass	-
9	Verify that system always returns evaluation time in command window.	Pass	-

Array of values

Table 4.8 shows test cases which are arranged into array of values. These values (input) are Clamping Force, Mold Dimension X, Mold Dimension Y and Shot Weight. The output value is number of machine group. The objective of this testing is to validate the machine group selection process. All 31,104 cases will be tested by the system and then compare actual output with the expected results. Beside of 4 columns of Clamping Force, Mold Dimension X, Mold Dimension Y and Shot Weight are used to be input data, 2 columns of Product Grade and Product Color are also used to be test data but these data do not need to be varied.

Table 4.8 Test case of the validation of machine group selection rules

Test Case ID	TCIMMS_Funct_08					
Description	Test the machine group selection rules					
Applicable for	MATLAB R2010a, Microsoft Excel 2007					
Requirement	-					
Initial Conditions	Test data is .xlsx file.					
Scenario	Array of values				Expected Result	Pass / Fail
	Clamping Force (Ton)	Mold Dimension X (mm)	Mold Dimension Y (mm)	Shot Weight (g)		
1	0	0	0	0	1	Pass
2	0	0	0	23	1	Pass
3	0	0	0	28	1	Pass
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
31102	75	350	380	55	6	Pass
31103	75	350	380	59	6	Pass
31104	75	350	380	99	6	Pass

Table 4.9 shows rules of machine selection within machine group 1. This group has two of machines that are M1 and M2. The input data of evaluation process use material color and grade of target products and previous job of each machine. The outputs are the most appropriated machines for that target products. In accordance with the system interface, the table below represents target product data by product specification.

Table 4.9 Test case of the validation of machine selection rules for machine group 1

<table>
<tr><td colspan="2">Test Case ID</td><td colspan="7">TCIMMS_Funct_09</td></tr>
<tr><td colspan="2">Description</td><td colspan="7">Test the machine number selection rules of machine group 1</td></tr>
<tr><td colspan="2">Applicable for</td><td colspan="7">MATLAB R2010a, Microsoft Excel 2007</td></tr>
<tr><td colspan="2">Requirement</td><td colspan="7">-</td></tr>
<tr><td colspan="2">Initial Conditions</td><td colspan="7">These rules were not included in Fuzzy Inference System</td></tr>
<tr><td rowspan="3">Scenario</td><td colspan="2">Product Specification</td><td colspan="4">Previous Jobs</td><td rowspan="3">Expected Results</td><td rowspan="3">Pass / Fail</td></tr>
<tr><td rowspan="2">Product Color</td><td rowspan="2">Material Grade</td><td colspan="2">Material Color</td><td colspan="2">Material Grade</td></tr>
<tr><td>M1</td><td>M2</td><td>M1</td><td>M2</td></tr>
<tr><td>1</td><td>Cl</td><td>A</td><td>Cl</td><td>Cl</td><td>A</td><td>A</td><td>M1</td><td>Pass</td></tr>
<tr><td>2</td><td>Cl</td><td>A</td><td>Cl</td><td>Cl</td><td>A</td><td>B</td><td>M1</td><td>Pass</td></tr>
<tr><td>3</td><td>Cl</td><td>A</td><td>Cl</td><td>Cl</td><td>A</td><td>C</td><td>M1</td><td>Pass</td></tr>
<tr><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td></tr>
<tr><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td></tr>
<tr><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td></tr>
<tr><td>4094</td><td>Bl</td><td>D</td><td>Bl</td><td>Bl</td><td>D</td><td>B</td><td>M2</td><td>Pass</td></tr>
<tr><td>4095</td><td>Bl</td><td>D</td><td>Bl</td><td>Bl</td><td>D</td><td>C</td><td>M2</td><td>Pass</td></tr>
<tr><td>4096</td><td>Bl</td><td>D</td><td>Bl</td><td>Bl</td><td>D</td><td>D</td><td>M1</td><td>Pass</td></tr>
</table>

Remarks: Cl is Clear color, Bl is Black color.

The table below (Table 4.10) is included of test rules of machine selection within machine group 2. This group is composed of machine M3, M4 and M5. Thus, the numbers of test rules are quite increased to 65,536.

Table 4.10 Test case of the validation of machine selection rules for machine group 2

Test Case ID	TCIMMS_Funct_10
Description	Test the machine number selection rules of machine group 2
Applicable for	MATLAB R2010a, Microsoft Excel 2007
Requirement	-
Initial Conditions	These rules were not included in Fuzzy Inference System

Scenario	Product Specification		Previous Jobs						Expected Results	Pass / Fail
	Product Color	Material Grade	Material Color			Material Grade				
			M3	M4	M5	M3	M4	M5		
1	Cl	A	Cl	Cl	Cl	A	A	A	M3	Pass
2	Cl	A	Cl	Cl	Cl	A	A	B	M3	Pass
3	Cl	A	Cl	Cl	Cl	A	A	C	M3	Pass
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
65534	Bl	D	Bl	Bl	Bl	D	D	B	M5	Pass
65535	Bl	D	Bl	Bl	Bl	D	D	C	M5	Pass
65536	Bl	D	Bl	Bl	Bl	D	D	D	M3	Pass

Remarks: Cl is Clear color, Bl is Black color.

Table 4.11 is included of machine selection test rules within machine group 3. This group is composed of machine M6 and M7. Thus, the numbers of test rules are same as machine group 1 (4,096 rules).

Table 4.11 Test case of the validation of machine selection rules for machine group 3

Test Case ID	TCIMMS_Funct_11
Description	Test the machine number selection rules of machine group 3
Applicable for	MATLAB R2010a, Microsoft Excel 2007
Requirement	-
Initial Conditions	These rules were not included in Fuzzy Inference System

Scenario	Product Specification		Previous Jobs				Expected Results	Pass / Fail
	Product Color	Material Grade	Material Color		Material Grade			
			M6	M7	M6	M7		
1	Cl	A	Cl	Cl	A	A	M6	Pass
2	Cl	A	Cl	Cl	A	B	M6	Pass
3	Cl	A	Cl	Cl	A	C	M6	Pass
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
4094	Bl	D	Bl	Bl	D	B	M7	Pass
4095	Bl	D	Bl	Bl	D	C	M7	Pass
4096	Bl	D	Bl	Bl	D	D	M6	Pass

Remarks: Cl is Clear color, Bl is Black color.

Some groups of machine have only on machine in the list. Machine group 4 and 5 were separated by the different machine specification. As a result, every test rules of these machine group return the same machine number as in Table 4.12 and Table 4.13.

Table 4.12 Test case of the validation of machine selection rules for machine group 4

Test Case ID	TCIMMS_Funct_12
Description	Test the machine number selection rules of machine group 4
Applicable for	MATLAB R2010a, Microsoft Excel 2007
Requirement	-
Initial Conditions	These rules were not included in Fuzzy Inference System

Scenario	Product Specification		Previous Jobs		Expected Results	Pass / Fail
	Product Color	Material Grade	Material Color	Material Grade		
			M8	M8		
1	Cl	A	Cl	A	M8	Pass
2	Cl	A	Cl	A	M8	Pass
3	Cl	A	Cl	A	M8	Pass
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
254	Bl	D	Bl	D	M8	Pass
255	Bl	D	Bl	D	M8	Pass
256	Bl	D	Bl	D	M8	Pass

Remarks: Cl is Clear color, Bl is Black color.

Table 4.13 Test case of the validation of machine selection rules for machine group 5

Test Case ID	TCIMMS_Funct_13
Description	Test the machine number selection rules of machine group 5
Applicable for	MATLAB R2010a, Microsoft Excel 2007
Requirement	-
Initial Conditions	These rules were not included in Fuzzy Inference System

Scenario	Product Specification		Previous Jobs		Expected Results	Pass / Fail
	Product Color	Material Grade	Material Color	Material Grade		
			M9	M9		
1	Cl	A	Cl	A	M9	Pass
2	Cl	A	Cl	A	M9	Pass
3	Cl	A	Cl	A	M9	Pass
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
254	Bl	D	Bl	D	M9	Pass
255	Bl	D	Bl	D	M9	Pass
256	Bl	D	Bl	D	M9	Pass

Remarks: Cl is Clear color, Bl is Black color.

The last machine group is 6 which are composed by 4,096 test rules to selecting of M10 or M11.

Table 4.14 Test case of the validation of machine selection rules for machine group 6

Test Case ID	TCIMMS_Funct_14
Description	Test the machine number selection rules of machine group 6
Applicable for	MATLAB R2010a, Microsoft Excel 2007
Requirement	-
Initial Conditions	These rules were not included in Fuzzy Inference System

Scenario	Product Specification		Previous Jobs				Expected Results	Pass / Fail
	Product Color	Material Grade	Material Color		Material Grade			
			M10	M11	M10	M11		
1	Cl	A	Cl	Cl	A	A	M10	Pass
2	Cl	A	Cl	Cl	A	B	M10	Pass
3	Cl	A	Cl	Cl	A	C	M10	Pass
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
4094	Bl	D	Bl	Bl	D	B	M11	Pass
4095	Bl	D	Bl	Bl	D	C	M11	Pass
4096	Bl	D	Bl	Bl	D	D	M11	Pass

Remarks: Cl is Clear color, Bl is Black color.

CHAPTER V
COMPUTATIONAL TIME AND DEFECTIVE IMPROVEMENT

This chapter represented the performance analysis of injection molding machine selection system that influenced on injection molding process improvement. The calculation was used to estimate the number of reduced defectives if the right machine was assigned to a job. The contents included all concepts of the orientation of type and quantity of defects in a plastic industry case study, using of pareto graph to select the target defects, the selection of defects that relate to our proposed system's criteria, process of system application and estimate the reduced defectives.

5.1 Comparison of a proposed system and a traditional method

The traditional machine selection step, the machine selection step will be started after the planner received orders from the sales department. This process will be finished when the appropriated machine was identified. Table 5.1 shows the machine selection process for one job (one task). According to understand the elapsed time phenomenon of the current machine selection system, this research has included the machine selection process observation and gathering the time using for each job in the unit of second. Then, the data of time spent for machine selection of 30 example jobs are shown in Table 5.2. The normal probability distribution can be used to explain the distribution of the data in Figure 5.1.

Table 5.1 Elapsed time of machine selection processes for one product

<table>
<tr><th rowspan="2">No.</th><th colspan="2">Traditional Method</th><th colspan="2">Proposed System</th></tr>
<tr><th>Operations</th><th>Time (second)</th><th>Operations</th><th>Time (second)</th></tr>
<tr><td>1</td><td>Receive task description and check</td><td>20</td><td>Receive task description and check</td><td>20</td></tr>
<tr><td rowspan="3">2</td><td rowspan="3">Access the databases of product specifications</td><td rowspan="3">22</td><td>Open program</td><td>25</td></tr>
<tr><td>Add product specification data to the system</td><td>7</td></tr>
<tr><td>Add all previous job data to the system</td><td>34</td></tr>
<tr><td>3</td><td>Compare task description and machine’s specification which is included of:
- Clamping force
- Distance between tie bars
- Shot weight
- Product color of previous job
- Material grade of previous job</td><td>105</td><td>Run the system</td><td>< 1</td></tr>
<tr><td></td><td>Total Time</td><td>147</td><td></td><td>87</td></tr>
</table>

The distribution of evaluation time of the planner is approximately normally distributed with a mean of 63.45 seconds (1.35 minutes) and a standard deviation of 26.79 seconds (0.27 minutes). Figure 5.1 shows evaluation time phenomenon in a normal distribution by determining 30 samples (n = 30). Therefore, the machine selection by human causes the uncertainty time that can be noticed by a standard deviation value (26.79 seconds).

Table 5.2 Elapsed time and probability of a machine selection process of current system

Evaluation Time (seconds)	Probability	Z-value	Mean ($\bar{x}$), (seconds)	Standard Deviation (SD)
19.99	0.003995	-1.62218	63.45	26.79
23	0.004763	-1.50984		
26	0.005605	-1.39787		
28.99	0.006511	-1.28627		
31	0.00715	-1.21125		
35	0.008472	-1.06196		
46.7	0.012246	-0.62527		
48	0.012608	-0.57675		
49	0.012874	-0.53943		
49	0.012874	-0.53943		
50.3	0.0132	-0.4909		
51.6	0.013502	-0.44238		
52.8	0.013758	-0.3976		
54	0.013992	-0.35281		
56	0.014325	-0.27816		
59	0.014686	-0.16619		
65	0.014865	0.057752		
66	0.014823	0.095076		
73	0.013974	0.356341		
76.2	0.013297	0.475776		
77	0.013103	0.505635		
78	0.012849	0.542959		
82	0.011717	0.692253		
87	0.01012	0.878871		
95	0.007445	1.17746		

Table 5.2 Elapsed time and probability of machine selection process of a current system (Cont.)

Evaluation Time (second)	Probability	Z-value	Mean ($\bar{x}$), (second)	Standard Deviation (SD)
96	0.00712	1.214784	63.45	26.79
102	0.005289	1.438725		
105	0.004474	1.550696		
105	0.004474	1.550696		
116	0.002176	1.961256		

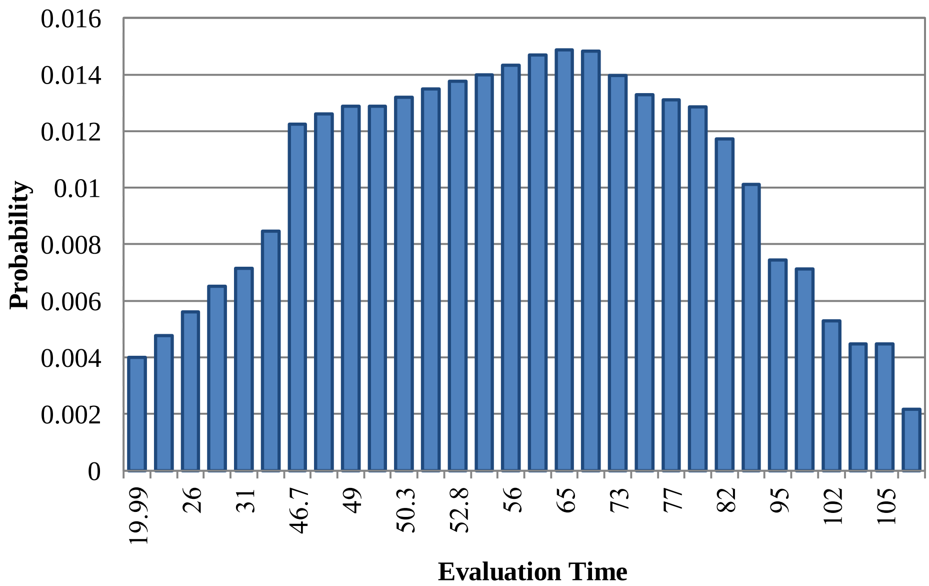


Figure 5.1 The normal probability distribution graph of machine selection time by current system

Next step, this research will demonstrate how the proposed system can enhance the production planner operation. The simulation method will be used to simulate the machine selection process. The elapsed time of execution will be measured to compare the performance of two methods. Finally, if this proposed system was proved to the feasibility of implementation, a user manual will be constructed.

Example

The actual data of July 2012 had been used to simulate test running and measure the computational time by the proposed system. After running of the system, all elapsed times of job have displayed on command window of MATLAB as in Figure 5.2. All results have shown in Table 5.3.

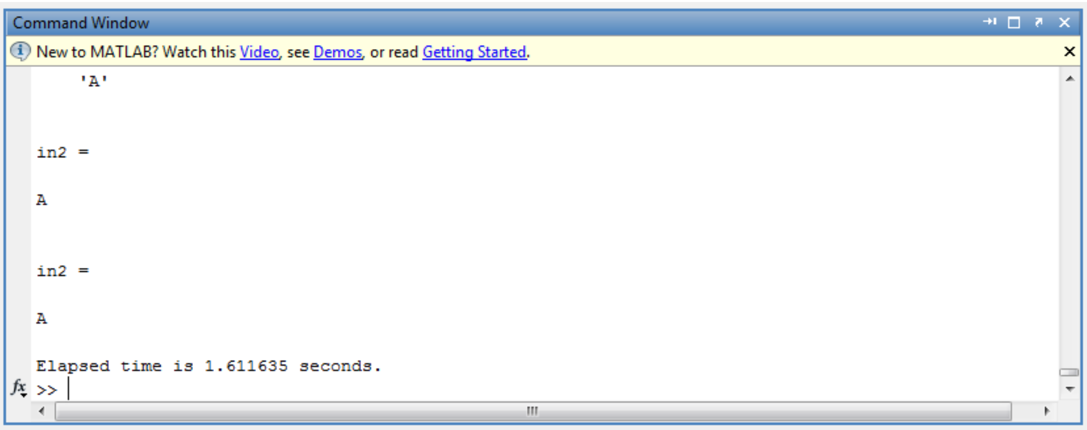


Figure 5.2 An elapsed time that has been measured by the proposed system

Table 5.3 The elapsed time recording of data from July 2012

Job No.	Count Time (sec.)	Δ	Job No.	Count Time (sec.)	Δ
1	1.0748	1.0748	11	1.2156	0.0115
2	1.0949	0.0201	12	1.2288	0.0132
3	1.1132	0.0183	13	1.2394	0.0106
4	1.1275	0.0143	14	1.2490	0.0096
5	1.1440	0.0165	15	1.2612	0.0122
6	1.1584	0.0144	16	1.2701	0.0089
7	1.1693	0.0109	17	1.2792	0.0091
8	1.1825	0.0132	18	1.2868	0.0076
9	1.1933	0.0108	19	1.2963	0.0095
10	1.2041	0.0108	20	1.3039	0.0076

Table 5.3 The elapsed time recording of data from July 2012 (Cont.)

Job No.	Count Time (sec.)	Δ	Job No.	Count Time (sec.)	Δ
21	1.3135	0.0096	38	1.4631	0.0097
22	1.3211	0.0076	39	1.4731	0.0100
23	1.3282	0.0071	40	1.4831	0.01
24	1.3350	0.0068	41	1.4920	0.0089
25	1.3422	0.0072	42	1.5012	0.0092
26	1.3493	0.0071	43	1.5116	0.0104
27	1.3560	0.0067	44	1.5201	0.0085
28	1.3667	0.0107	45	1.5289	0.0088
29	1.3774	0.0107	46	1.5375	0.0086
30	1.3873	0.0099	47	1.5464	0.0089
31	1.3965	0.0092	48	1.5549	0.0085
32	1.4051	0.0086	49	1.5650	0.0101
33	1.4144	0.0093	50	1.5736	0.0086
34	1.4230	0.0086	51	1.5845	0.0109
35	1.4336	0.0106	52	1.5931	0.0086
36	1.4448	0.0112	53	1.6017	0.0086
37	1.4534	0.0086	54	1.6116	0.0099

From Table 5.3, total elapsed times or computational times by our proposed system for 54 jobs input is 1.6116 second. Job no. 1 has elapsed time higher than other jobs caused by data loading process at the initial stage of system running. The average elapsed time of all jobs (except job no.1) is 0.0101 second. Meanwhile, an elapsed time average of current system is 63.45 second per a job. It indicates that the proposed system reduced elapsed times of the machine selection significantly. Besides, the elapsed time by the planner would double if the second job had been started. Therefore, the proposed system could maintain its computation time. The elapsed time of the proposed system is still similar with the machine selection process of the first job.

5.2 Type and quantity of defective in a plastic industry case study

The observation and collecting data have been carried out for five months, during July 2012 - November 2012, from the plastic industry case study. The collections of data are the defectives from each machine which were separated by type of defects. The scope of the study was confined to 11 injection molding machines (35 machines of total). They are the same set of machines as the system development phase. All concerned data were contained in Table 5.4. From this table, 20 types of defect could be found from all defectives. These are flash, short shot, black dot, broken, gate NG, crack, warping, burn mark, oily, oily Mark, bubble, scratch, sink mark, flow mark, discolor, weld line, dakon, silver, over cut and other. The characteristics and root-causes of 19 types have referred by the statements in chapter 2 but the type of 'other'. The defect type of 'other' is different from those 19 types and this could not be identified the properties and root-causes obviously, therefore they were called 'other'. The highest total defective is silver and the next are short shot, black dot, sink mark, other, scratch, weld line, over cut, oil mark, bubble, oily, broken, burn mark, dakon, discolor, flash, flow mark, gate NG, warping and crack respectively. Their quantities varied in a wide range depending on the probability of occurrence. We found that short shot was the highest amount in July and September, around 39.43% and 58.42% of total defectives. In August, silver was the first by 48.62% and there were the majority of black dot around 40.35% and 68.34% in October and November. For the second of all, the percentages of defectives were quite different from the third. Therefore, if the first or second just would be reduced in every months, they could affect to total defectives obviously.

Table 5.4 Defects that produced from 11 machines on July 2012 to January 2013

Defects	July	August	September	October	November	Total
Silver	17437	7529	1156	952	15	27089
Short shot	18252	919	1790	1126	133	22220
Black dot	3081	2370	29	3086	1183	9749
Other	1638	1974	32	476	40	4160
Sink mark	2113	674	54	401	65	3307
Scratch	655	471	0	33	0	1159
Weld line	325	645	0	20	0	990
Over cut	513	45	0	385	7	950
Oily mark	487	0	0	400	0	887
Bubble	111	28	0	699	0	838
Oily	623	22	0	0	0	645
Broken	50	165	0	3	280	498
Burn mark	233	252	3	0	0	488
Dakon	251	221	0	10	0	482
Discolor	350	33	0	0	0	383
Flash	109	37	0	27	0	173
Flow mark	20	39	0	30	0	89
Gate NG	0	60	0	0	0	60
Warping	36	0	0	0	0	36
Crack	4	0	0	0	8	12
Total	46288	15484	3064	7648	1731	74215

5.3 The selection of defects for the defect reduction by pareto graph

At this section, we will exhibit more clearly of the amount of defective of all defects. The number of defective parts varied from 0% to 68%, thus it was not necessary to improve all defect types because it would be difficult and time-consuming. The decreasing of defect that having huge proportion is better solution based on defect causes controllability perspective and highly returns. These ideas contributed to the application of pareto graph for selecting of defects. They were having potential to be improved by the proposed system. The pareto graphs represent the cumulative percentages of each defect in July, August, September, October and November as in Figure 5.3, Figure 5.4, Figure 5.5, Figure 5.6 and Figure 5.7 respectively.

In the next step, defects that having cumulative percentage at least 80% would be selected to use by the proposed system testing. The amount of reduced defective parts would be collected. Infers to Figure 5.3, in July, the sum of short shot plus silver could gain up to 83.76%. Thus, these defects had to be investigated the residual defective part from the test. In August (Figure 5.4), silver, black dot, other and short shot were chosen with 82.61%. There were five defects have been selected in October (Figure 5.7): black dot, short shot, silver, bubble and other, the total was 82.88%. But silver and black dot quantities have been collected greater than another defects in September (Figure 5.5); cumulative percentage exceeded to 96.15%. As in November (Figure 5.7), only two defects (e.g. black dot and broken) were chosen because the cumulative percentage of them reached to 84.52%. All results as mentioned have contained in Table 5.5.

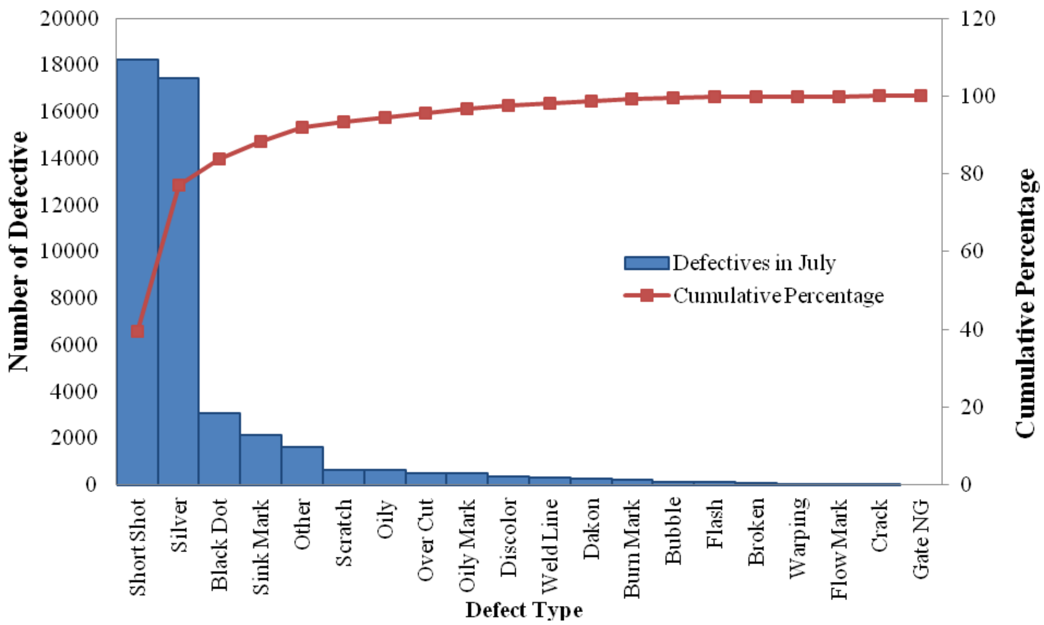


Figure 5.3 Pareto graph of defectives in July 2012

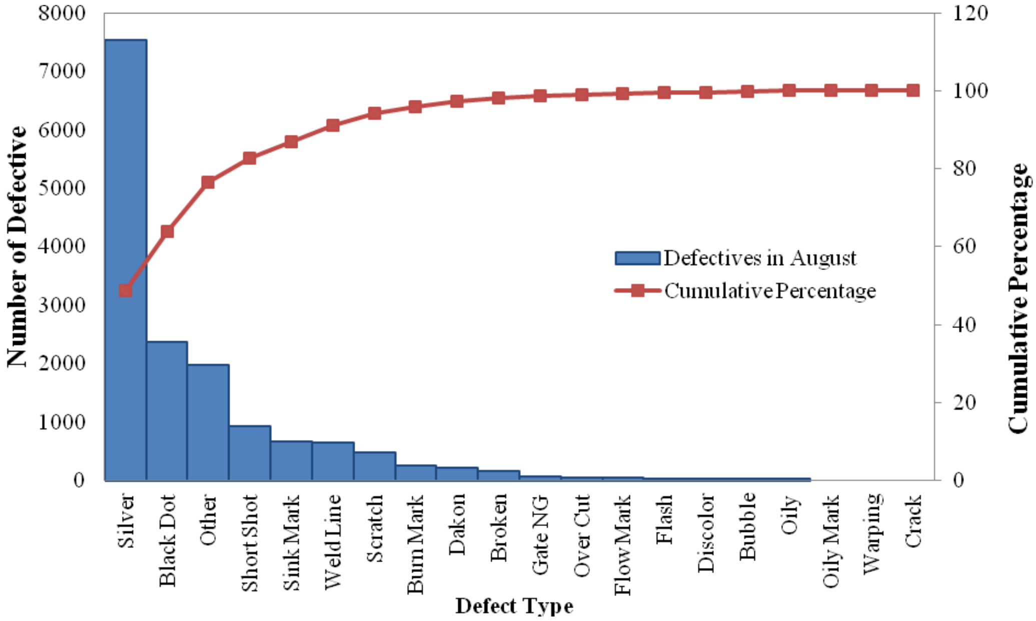


Figure 5.4 Pareto graph of defectives in August 2012

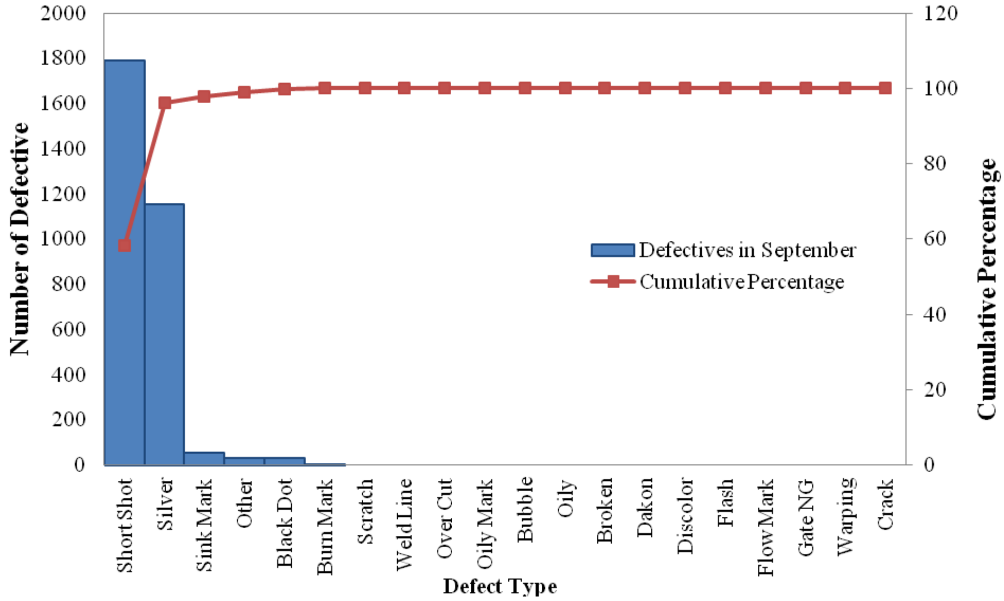


Figure 5.5 Pareto graph of defectives in September 2012

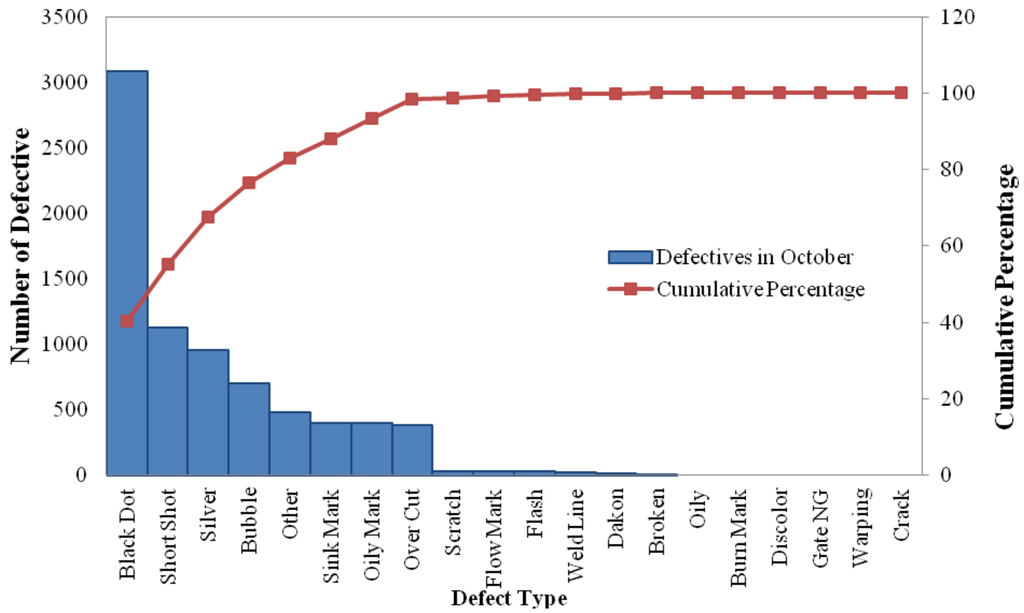


Figure 5.6 Pareto graph of defectives in October 2012

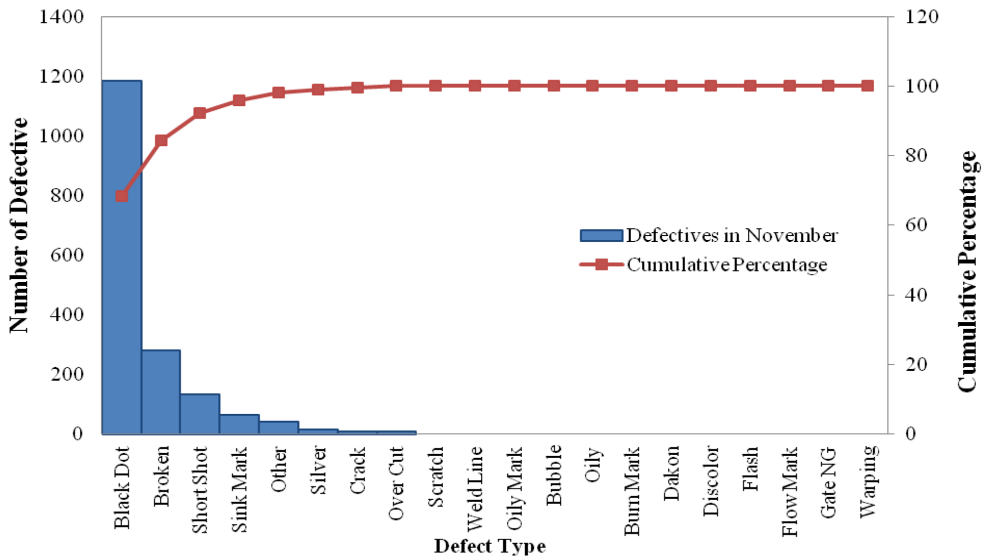


Figure 5.7 Pareto graph of defectives in November 2012

Table 5.5 Summary of percentage and cumulative percentage of the selected defects

Months	Defects	Percentage (%)	Cumulative Percentage (%)
July	Short shot Silver Black dot	39.43 37.67 6.66	83.76
August	Silver Black dot Others Short shot	48.62 15.31 12.75 5.94	82.61
September	Short shot Silver	58.42 37.73	96.15
October	Black dot Short shot Silver Bubble Others	40.35 14.72 12.45 9.14 6.22	82.88
November	Black dot Broken	68.34 16.16	84.52

5.4 The defects that relate to the proposed system's selection criteria

In according to this study has not applied on the real production in plastic manufacturer case study, the residual defective parts have obtained by calculation. After applying the proposed system, the residual defective parts are the differences of actual defective parts and the reduced parts by the new machine assignment: reduced defective parts related to the selection criteria of the proposed system.

From the literature review, this research has synthesized the relation of defect causes and troubleshooting approaches as in Table 5.6. Based on the table, all defect types referred to the Table 5.5 and have added common causes and point of troubleshooting which have categorized into three types; mold, machine and material.

For example, black streaking (black dot) caused by the contamination of melting plastic in the barrel. The point of troubleshooting should focuses on raw material preparation: material color of a new job and the last job should be the same on a machine. Therefore, this defect relates to the product color criteria of our proposed system. We can assume that this black streaking part will be reduced to zero if the new machine have assigned to this job. However, the probability of having black streaking in such a right machine always is. Thus, the residual defective part cannot be zero but they have to plus with the product of the total defectives from the machines have not reassigned and probability of having defect from the machines that have not reassigned. The details of calculation will be illustrated in the next topic forward.

Based on Table 5.6, the defects that can be improved by our proposed system are summarized in Table 5.7, separated by month. Monthly, short shot and black dot have relation to our selection criteria. Therefore, these defects can be improved by the proposed system but silver, bubble, broken and other will not be improved.

Table 5.6 The defects in the injection molding influenced by machine selection criteria in the case study

Defects	Common Causes	Point of Troubleshooting			Reference
		Man	Machine	Material	
Black streaking	Contaminated plastication system Improper drying	-	-	✓	Gordon (2010)
Sink marks	Underpacking Mold malfunction Poor part design	✓	-	-	Gordon (2010)
Silver streaking	Moisture trapped in material Melt temperature too high Condensation in mold Poor pressurization Water leak in mold	✓	✓	✓	Gordon (2010) Bown (1979)
Short shots	Cavity not filling properly Insufficient melt volume Moisture in material Mold release agent used High-pressure drop in mold Poor mold design	-	✓	✓	Gordon (2010) Bown (1979)

Table 5.6 The defects in the injection molding influenced by machine selection criteria in the case study (Cont.)

Defects	Common Causes	Point of Troubleshooting			Reference
		Man	Machine	Material	
Bubble	Moisture Entrained air Not typical material flow pattern	-	-	✓	Gordon (2010) Bown (1979)
Broken	Processing temperatures too high Contaminants/fines present Improper drying Wet resin Improperly sized cylinder Poor mold design Too much regrind	-	✓	✓	Gordon (2010) Bown (1979)

Table 5.7 Defects that are affected by the proposed system

Months	Target Defects
July	Short shot Black dot
August	Black dot Short shot
September	Black dot
October	Black dot Short shot
November	Black dot

5.5 The system application approach and the residual defective quantity estimation

The proposed system had tested with actual data from industry case study. This study had used the machines and defective parts data from July 2012 as example. The outputs of the system were the new machines which provided more suitability and lower defective parts. This part describes the detail of six major steps that are pre-data preparation, input data preparation, system running, conversion of output machine to actual machine, comparison between the defectives that produced by actual machines and the proposed system, calculation of the probability of having defective parts from machines that had not reassigned and the estimation of the residual defective parts after using the proposed system and performance measurement.

5.5.1 Pre-data preparation

Based on the same set of machines as the proposed system's machines, we have selected jobs that had short shot and black dot.

Note To show all hiding data rows, this file has not to be filtered mode, if not, go to 'Data' menu and click 'Filter' button to inactive it in case of *.xlsx file.

5.5.2 Input data preparation

The product code of the job that have had short shot and black dot parts to find out product specification in the search box on the screen or fill out or insert data (*.xlsx) file. In case of inserting data file, all jobs should be ascended up by clamping force before using. This step helps user operates easily and returns proper results. Table 5.8 and Figure 5.8 show the example of sorted input data file by clamping force: these jobs have had short shot parts. Then, material grade and product color data of the latest batches that were collected from 11 machines in June (refers to Table 5.9) must be set in Previous jobs table (Figure 5.9).

Table 5.8 Test data of July

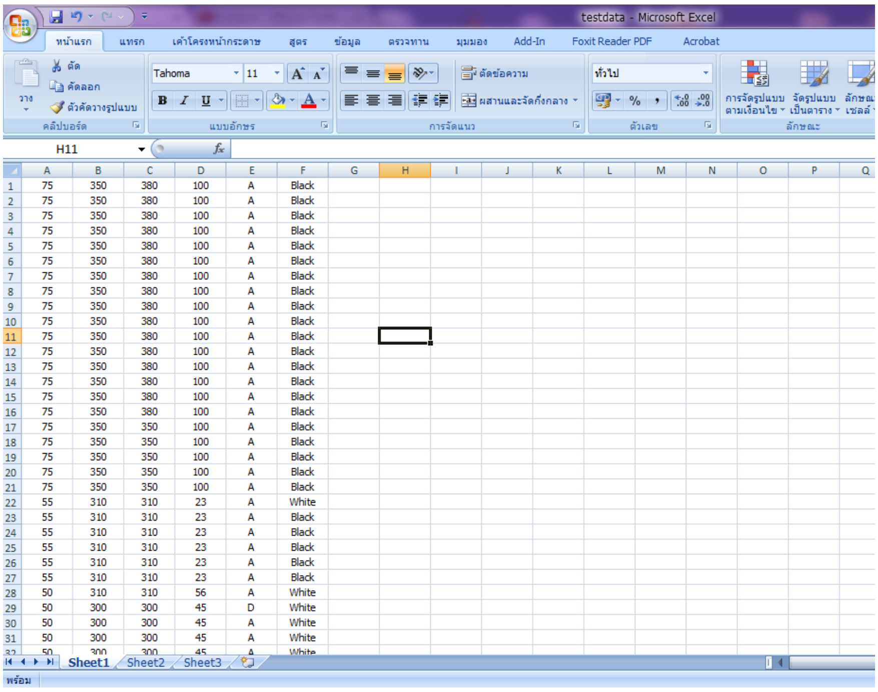
Clamping Force	Distance between TiebarX	Distance between TiebarY	Shot Weight	Material Grade	Product Color
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	380	100	A	Black
75	350	350	100	A	Black
75	350	350	100	A	Black
75	350	350	100	A	Black
75	350	350	100	A	Black

Table 5.8 Test data of July (Cont.)

Clamping Force	Distance between TiebarX	Distance between TiebarY	Shot Weight	Material Grade	Product Color
75	350	350	100	A	Black
55	310	310	23	A	White
55	310	310	23	A	Black
55	310	310	23	A	Black
55	310	310	23	A	Black
55	310	310	23	A	Black
55	310	310	23	A	Black
50	310	310	56	A	White
50	300	300	45	D	White
50	300	300	45	A	White
50	300	300	45	A	White
50	300	300	45	A	White
50	300	300	45	A	Black
50	300	300	45	A	White
50	310	310	56	A	White
50	300	300	45	A	White
50	300	300	45	A	White
50	300	300	45	A	White
50	300	300	45	A	White
50	300	300	45	A	White
50	300	300	45	A	Black
50	300	300	45	A	Black
50	300	300	45	A	Black
50	300	300	45	A	Black
50	310	310	56	C	Clear
50	300	300	45	C	Black
50	310	310	56	C	Clear
50	310	310	56	C	Clear
35	300	300	45	C	Black
30	310	310	23	A	White
30	260	260	28	A	White
30	260	260	28	A	White
30	260	260	28	A	White

Table 5.8 Test data of July (Cont.)

Clamping Force	Distance between TiebarX	Distance between TiebarY	Shot Weight	Material Grade	Product Color
30	260	260	28	A	Black



	A	B	C	D	E	F
1	75	350	380	100	A	Black
2	75	350	380	100	A	Black
3	75	350	380	100	A	Black
4	75	350	380	100	A	Black
5	75	350	380	100	A	Black
6	75	350	380	100	A	Black
7	75	350	380	100	A	Black
8	75	350	380	100	A	Black
9	75	350	380	100	A	Black
10	75	350	380	100	A	Black
11	75	350	380	100	A	Black
12	75	350	380	100	A	Black
13	75	350	380	100	A	Black
14	75	350	380	100	A	Black
15	75	350	380	100	A	Black
16	75	350	380	100	A	Black
17	75	350	350	100	A	Black
18	75	350	350	100	A	Black
19	75	350	350	100	A	Black
20	75	350	350	100	A	Black
21	75	350	350	100	A	Black
22	55	310	310	23	A	White
23	55	310	310	23	A	Black
24	55	310	310	23	A	Black
25	55	310	310	23	A	Black
26	55	310	310	23	A	Black
27	55	310	310	23	A	Black
28	50	310	310	56	A	White
29	50	300	300	45	D	White
30	50	300	300	45	A	White
31	50	300	300	45	A	White
32	50	300	300	45	A	White

Figure 5.8 Test data of July 2012 in excel form

Table 5.9 Previous data of the latest batches produced by 11 machines

No. of Machine	June		July		Aug		Sep		Oct	
1	A	Clear	A	Clear	C	Black	C	Black	C	Black
2	A	Clear	A	Clear	A	Clear	A	Clear	A	Clear
3	C	Black	A	Black	C	Black	C	Black	C	Black
4	C	Clear	C	Clear	C	Clear	C	Clear	C	Clear
5	D	White	A	White	D	White	C	Clear	C	Clear
6	A	Black	A	White	D	Black	D	Black	D	Black
7	C	Black	C	Clear	C	Black	C	Black	A	White

Table 5.9 Previous data of the latest batches produced by 11 machines (Cont.)

No. of Machine	June		July		Aug		Sep		Oct	
1	A	Clear	A	Clear	C	Black	C	Black	C	Black
8	A	Clear	A	Clear	D	Black	D	Black	C	Clear
9	C	Grey	A	Black	A	Clear	A	Clear	A	Black
10	D	Clear	A	Black	A	Black	A	Black	A	Black
11	D	Clear	D	Clear	D	Clear	D	Clear	C	White

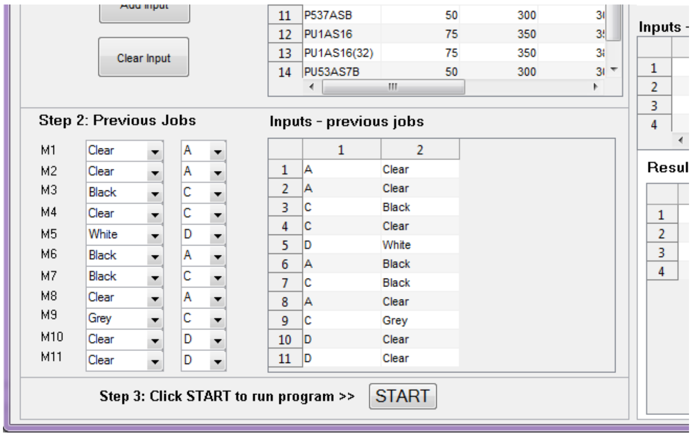


Figure 5.9 Product specification and previous jobs data have filled up on the system

5.5.3 System running

After product specifications were added to Inputs – product specification table, click 'START' to run program.

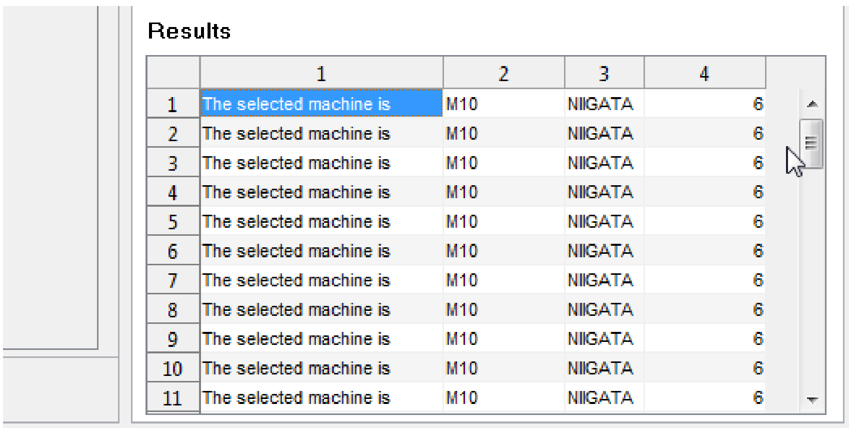


Figure 5.10 Machine selection results in Results table

5.5.4 Conversion of output machines to actual machines

In order to broad applicable in most industries, the proposed system represents all 11 machines with machine number 1 to 11. In an individual manner, users have to match the proposed system's machines to theirs. Table 5.10 is the reference of conversion the proposed system's machines to actual machines of our case study.

Table 5.10 The reference of proposed system's machines and case study's machines

Machine Number (The proposed system)	Machine Number (Actual)	Machine Name
M1	M5	TOSHIBA IS30EPN-1Y
M2	M6	JSW J30 ED
M3	M33	NIIGATA
M4	M34	NIIGATA
M5	M35	NIIGATA
M6	M1	JSW J50 EII
M7	M4	TOSHIBA IS50EP-1.5A
M8	M3	NIGATA NN50E
M9	M2	JSW J55 EII
M10	M31	NIIGATA
M11	M32	NIIGATA

5.5.5 Comparison between the defectives that produced by the actual machines and the proposed system

The defectives comparison between actual machines and the proposed system aims to define all parameters to estimate total defectives after using the proposed system. The number of defectives of short shot in July, number of reduced defectives, number of defectives from the machines that have not reassigned in July and total number of defectives from July to November must be defined. If the system matched a new machine to a job, it means that job produced by the reassigned machine. Meanwhile, the reassignment by the system can provide another machine with more suitable condition; in this case, we assume that defectives will be reduced to zero and the number of reduced defectives can be known. Table 5.11 shows results of selecting machine by the proposed system. In the reduced defectives column, short shot part is not reduced but 16 pieces of black dot can be eliminated from 2,979 pieces.

Example

Case 1: Product code is EE57LFTC604812, operating date is 3/7/12, the selected machine (actual production) is M1, selected machine (proposed system) is M1, defective parts (actual) is included of 82 pieces of short shot and 9 pieces of black dot.

Product code EE57LFTC604812 had operated on M1 (Actual production) and 82 pieces of short shot had occurred. From Table 5.11, this states that applying inadequately clamping force influences on short shot parts formation, so the statement was used to investigate of reducing short shot parts in this job. After running of the proposed system, M1 had still selected for this job. Thus, machine conditions were not changed both clamping force and material color from the last batch. The 82 pieces of short shot were not removed and 9 pieces of black dot as well.

Case 2: Product code is F26P1LFDN-00136, operating date is 5/7/12, the selected machine (actual production) is M2, selected machine (proposed system) is M35, defective parts (actual) is included of 65 pieces of short shot and 4 pieces of black dot.

Product code F26P1LFDN-00136 had operated on M2 (Actual production) and 65 pieces of short shot are occurred. After running of the proposed system, M35 had selected for this job. Due to this job requires 30 tons of clamping force for injection, M2 has 55 tons of maximum clamping force and M35 has 50 tons of maximum clamping force. Therefore, both machines were having enough capacity to handle it. The reduction of short shot parts could not be assumed. We assumed that 82 pieces of short shot are not removed. In opposite, product color of this job is white while the last injection in June of M2 (the proposed system calls M9) used grey material. The proposed system recommended the better machine with M35 (or M5) which had operated with white material in June. Thus, we assumed that 4 pieces of black dot would be reduced to 0.

Table 5.11 Results of the machine selection and the defect reduction by the proposed system in July

Date	Product Code	Selected Machine (Actual Production)	Selected Machine (Proposed system)		Defectives (Actual)		Reduced Defectives	
			Before Conversion	After Conversion	Short Shot	Black Dot	Short Shot	Black Dot
3/7/12	EE57LFTC604812	M1	M6	M1	82	9	-	-
5/7/12	F26P1LFDN-00136	M2	M5	M35	65	4	-	4
5/7/12	F26P1LFDN-00128	M34	M5	M35	60	-	-	-
5/7/12	P537ASB	M35	M5	M35	25	-	-	-
6/7/12	F26P1LFDN-00136	M2	M3	M33	129	35	-	-
6/7/12	P537ASB	M35	M5	M35	24	14	-	-
7/7/12	F26P1LFDN-00136	M2	M3	M33	160	13	-	-
7/7/12	P537ASB	M35	M5	M35	38	13	-	-
9/7/12	F26P1LFDN-00136	M2	M3	M33	53	12	-	12
9/7/12	PU1AS16(32)	M31	M10	M31	147	-	-	-
9/7/12	EE57LFTC510712	M34	M3	M33	6	-	-	-
9/7/12	P537ASB	M35	M5	M35	4	30	-	-
10/7/12	EE57LFTC604812	M2	M6	M1	5	-	-	-
10/7/12	PU1AS16(32)	M31	M10	M31	54	-	-	-
10/7/12	P537ASB	M35	M5	M35	-	17	-	-
11/7/12	PU1AS16(32)	M31	M10	M31	3	-	-	-
11/7/12	P537ASB	M35	M5	M35	119	45	-	-
12/7/12	00-54016	M2	M9	M2	-	20	-	-
12/7/12	PU1AS16(32)	M31	M10	M31	421	-	-	-
12/7/12	P537ASB	M35	M5	M35	60	171	-	-
13/7/12	PU1AS16(32)	M31	M10	M31	152	-	-	-
13/7/12	P537ASB	M35	M5	M35	32	50	-	-
14/7/12	PU1AS16(32)	M31	M10	M31	295	-	-	-
14/7/12	P537ASB	M35	M5	M35	367	116	-	-
15/7/12	PU1AS16(32)	M31	M10	M31	334	-	-	-
16/7/12	PU1AS16(32)	M31	M10	M31	509	-	-	-
16/7/12	PU53AS7B	M35	M3	M33	15	-	-	-
17/7/12	P5511410098	M2	M9	M2	8	-	-	-
17/7/12	PU1AS16(32)	M31	M10	M31	283	-	-	-

Table 5.11 Results of the machine selection and the defect reduction by the proposed system in July (Cont.)

Date	Product Code	Selected Machine	Selected Machine (Proposed system)		Defectives (Actual)		Reduced Defectives	
		(Actual Production)	Before Conversion	After Conversion	Short Shot	Black Dot	Short Shot	Black Dot
17/7/12	PU53AS7B	M35	M3	M33	34	-	-	-
18/7/12	P5511410098	M2	M9	M2	111	-	-	-
18/7/12	PU1AS16(32)	M31	M10	M31	728	-	-	-
18/7/12	PU53AS7B	M35	M3	M33	118	-	-	-
19/7/12	P5511410098	M2	M9	M2	89	-	-	-
19/7/12	PU1AS16(32)	M31	M10	M31	675	-	-	-
19/7/12	PU53AS7B	M35	M3	M33	26	-	-	-
20/7/12	P5511410098	M2	M9	M2	1	-	-	-
20/7/12	PU1AS16(32)	M31	M10	M31	197	-	-	-
21/7/12	PU1AS16(32)	M31	M10	M31	1,882	-	-	-
23/7/12	PU1AS16(32)	M31	M10	M31	1,942	-	-	-
24/7/12	P5511410098	M2	M9	M2	34	-	-	-
24/7/12	PU1AS16(32)	M31	M10	M31	561	-	-	-
25/7/12	PU1AS16(32)	M31	M10	M31	664	-	-	-
25/7/12	EE33LFTC802011	M34	M3	M33	2	-	-	-
26/7/12	IH55LFMP066-02	M1	M7	M4	-	501	-	-
26/7/12	PU1AS16	M31	M10	M31	274	-	-	-
26/7/12	EE33LFTC802001	M35	M3	M33	15	-	-	-
27/7/12	IH55LFMP066-02	M1	M7	M4	-	157	-	-
27/7/12	PU1AS16	M31	M10	M31	2,611	-	-	-
28/7/12	EE33LFTC802031	M1	M7	M4	-	10	-	-
28/7/12	PU1AS16	M31	M10	M31	1,218	-	-	-
30/7/12	PU1AS16	M31	M10	M31	2,192	-	-	-
30/7/12	R411LFTH24263C	M33	M3	M33	11	-	-	-
31/7/12	PU1AS16	M31	M10	M31	-	1,762	-	-
Total					16,835	2,979	0	16

Table 5.12 Results of the machine selection and the defect reduction by the proposed system in August

Date	Product Code	Selected Machine (Actual Production)	Selected Machine (Proposed system)		Defectives (Actual)		Reduced Defectives	
			Before Conversion	After Conversion	Short Shot	Black Dot	Short Shot	Black Dot
1/8/12	EE57LFTC510712	M2	M9	M2	5	-	-	-
2/8/12	EE57LFTC510712	M2	M9	M2	2	-	-	-
3/8/12	EE57LFTC510712	M2	M9	M2	6	-	-	-
3/8/12	R411LFTH23826C	M1	M7	M4	4	-	-	-
6/8/12	EE57LFTC510712	M2	M9	M2	17	-	-	-
6/8/12	R411LFTH23826C	M1	M7	M4	-	42	-	42
7/8/12	PD1AS29L	M31	M10	M31	-	26	-	-
7/8/12	EE57LFTC510712	M2	M9	M2	14	-	-	-
15/8/12	E26P0LFBJ-02063	M4	M6	M1	-	16	-	-
7/8/12	R411LFTH23826C	M1	M7	M4	-	32	-	32
7/8/12	R411LFTH23857B	M34	M6	M1	97	-	-	-
8/8/12	P12AS8AL	M31	M10	M31	-	10	-	-
8/8/12	EE57LFTC510712	M2	M9	M2	1	-	-	-
9/8/12	P12AS8AL	M31	M10	M31	-	8	-	-
9/8/12	PD1AS28AL	M31	M10	M31	-	6	-	-
9/8/12	F26P1LFDN-00145	M34	M1	M5	37	-	-	-
10/8/12	F26P1LFDN-00134	M01	M7	M4	2	-	-	-
10/8/12	F26P1LFDN-00136	M4	M5	M35	13	26	-	-
10/8/12	U41PC5B	M31	M11	M32	6	-	-	-
10/8/12	EE33LFTC905315	M1	M7	M4	-	50	-	50
11/8/12	F26P1LFDN-00136	M4	M5	M35	15	1	-	-
11/8/12	U41PC5B	M31	M11	M32	2	-	-	-
11/8/12	EE33LFTC905315	M1	M7	M4	-	90	-	90
13/8/12	K41PC5B	M31	M11	M32	-	19	-	19
13/8/12	R411LFTH23829C	M34	M3	M33	34	-	-	-
14/8/12	563DR-W	M35	M5	M35	-	19	-	-
14/8/12	EE33LFTC905315	M1	M7	M4	-	34	-	34
14/8/12	R411LFTH23829C	M34	M3	M33	59	-	-	-
15/8/12	P-984DZ7A	M2	M9	M2	-	14	-	-
15/8/12	P-984DZ7B	M2	M9	M2	-	14	-	-
15/8/12	EE33LFTC905315	M1	M7	M4	-	10	-	10
15/8/12	EE33LFTC905511	M1	M7	M4	-	10	-	10
16/8/12	D159AS	M31	M10	M31	8	-	-	-
16/8/12	EE33LFTC905511	M1	M7	M4	11	10	-	10

Table 5.12 Results of the machine selection and the defect reduction by the proposed system in August (Cont.)

Date	Product Code	Selected Machine (Actual Production)	Selected Machine (Proposed system)		Defectives (Actual)		Reduced Defectives	
			Before Conversion	After Conversion	Short Shot	Black Dot	Short Shot	Black Dot
17/8/12	EE33LFTC905315	M1	M7	M4	15	16	-	10
17/8/12	5625Z3	M34	M7	M4	11	-	-	-
18/8/12	F26P1LFDN-0095	M35	M10	M31	2	-	-	-
21/8/12	F26P1LFDN-0095	M35	M10	M31	-	3	-	-
22/8/12	F26P1LFDN-00128	M34	M5	M35	40	762	-	-
23/8/12	PU1AS59	M31	M10	M31	1	-	-	-
24/8/12	F26P1LFDN-00128	M34	M5	M35	14	7	-	-
24/8/12	KN66JF8A	M32	M11	M32	19	18	-	-
25/8/12	PD1AS28AL	M31	M10	M31	-	16	-	-
25/8/12	KN66JF8A	M32	M11	M32	146	-	-	-
28/8/12	PU1AS28AL	M31	M10	M31	28	-	-	-
28/8/12	F26P1LFDN-00095	M35	M11	M32	-	11	-	-
28/8/12	KN66JF8A	M32	M11	M32	45	5	-	-
29/8/12	R411LFTH23828F	M35	M8	M3	20	-	-	-
30/8/12	R411LFTH23828F	M35	M8	M3	2	-	-	-
Total					676	1,275	0	313

Table 5.13 Results of the machine selection and the defect reduction by the proposed system in September

Date	Product Code	Selected Machine (Actual Production)	Selected Machine (Proposed system)		Defectives (Actual)		Reduced Defectives	
			Before Conversion	After Conversion	Short Shot	Black Dot	Short Shot	Black Dot
1/9/12	PU1AS16	M31	M10	M31	925	-	-	-
3/9/12	PU1AS16	M31	M10	M31	100	-	-	-
3/9/12	R411LFTH23829C	M34	M3	M33	49	-	-	-
4/9/12	PU1AS16	M31	M10	M31	8	-	-	-
4/9/12	PU1AS16	M31	M10	M31	665	-	-	-
Total					1,747	0	0	0

Table 5.14 Results of the machine selection and the defect reduction by the proposed system in October

Date	Product Code	Selected Machine (Actual Production)	Selected Machine (Proposed system)		Defectives (Actual)		Reduced Defectives	
			Before Conversion	After Conversion	Short Shot	Black Dot	Short Shot	Black Dot
1/10/12	P-203AC451	M31	M11	32	-	6	-	6
2/10/12	P-203AC451	M31	M11	32	-	13	-	13
3/10/12	P-203AC451	M31	M11	32	-	6	-	6
4/10/12	EE57LFTC605511	M33	M3	33	-	30	-	-
5/10/12	PU-4AS3A	M32	M11	32	11	-	-	-
5/10/12	EE57LFTC605511	M33	M3	33	8	248	-	-
6/10/12	PU-4AS3A	M32	M11	32	-	601	-	-
8/10/12	PU-4AS3A	M32	M11	32	4	223	-	-
9/10/12	PU-4AS3A	M32	M11	32	-	20	-	-
11/10/12	PU-4AS3A	M32	M11	32	40	-	-	-
11/10/12	PD12AS8	M32	M11	32	55	15	-	-
11/10/12	EE57LFTC605511	M33	M3	33	-	105	-	-
12/10/12	R411LFTH23829C	M2	M3	33	10	-	-	-
12/10/12	F26P1LFDN-00277	M31	M10	31	35	-	-	-
12/10/12	PD12AS8	M32	M11	32	-	239	-	-
12/10/12	EE57LFTC604812	M33	M7	4	270	-	-	-
13/10/12	R411LFTH23829C	M2	M3	33	14	-	-	-
13/10/12	PD12AS8	M32	M11	32	-	290	-	-
13/10/12	EE57LFTC604812	M33	M7	4	10	10	-	-
15/10/12	PD12AS8	M32	M11	32	-	40	-	-
15/10/12	P128AS	M32	M10	31	-	11	-	11
17/10/12	PU1AS16	M31	M10	31	60	2	-	-
18/10/12	1551P410098	M2	M9	2	53	-	-	-
18/10/12	PU1AS16	M31	M10	31	10	-	-	-
18/10/12	P15AS8	M32	M11	32	-	5	-	-
19/10/12	1551P410098	M2	M9	2	8	-	-	-
19/10/12	PU1AS16	M31	M10	31	2	-	-	-
19/10/12	P159AS	M32	M11	32	32	10	-	-
19/10/12	EE57LFTC604812	M33	M7	4	-	10	-	-
20/10/12	PU1AS16	M31	M10	31	-	13	-	-
20/10/12	P159AS	M32	M11	32	-	151	-	-
23/10/12	PU1AS16	M31	M10	31	49	-	-	-
23/10/12	P159AS	M32	M11	32	-	214	-	-

Table 5.14 Results of the machine selection and the defect reduction by the proposed system in October (Cont.)

Date	Product Code	Selected Machine (Actual Production)	Selected Machine (Proposed system)		Defectives (Actual)		Reduced Defectives	
			Before Conversion	After Conversion	Short Shot	Black Dot	Short Shot	Black Dot
24/10/12	PU1AS16	M31	M10	31	7	1	-	-
24/10/12	P159AS	M32	M11	32	6	55	-	-
25/10/12	PU1AS16	M31	M10	31	-	3	-	-
25/10/12	PU1AS59	M32	M10	31	6	-	-	-
25/10/12	R411LFTH23887B	M34	M8	3	-	51	-	-
26/10/12	PU1AS16	M31	M10	31	147	-	-	-
26/10/12	P129AS	M32	M11	32	58	93	-	-
26/10/12	R411LFTH23887B	M34	M8	3	-	5	-	-
27/10/12	P129AS	M32	M11	32	-	230	-	-
29/10/12	PU1AS16	M31	M10	31	13	3	-	-
29/10/12	P-203AC451	M31	M11	32	-	31	-	31
29/10/12	P129AS	M32	M11	32	-	126	-	126
29/10/12	R411LFTH23887B	M34	M8	3	-	5	-	-
30/10/12	P-203AC451	M31	M11	32	-	61	-	61
30/10/12	P129AS	M32	M11	32	-	10	-	-
30/10/12	D159AS	M32	M11	32	-	80	-	-
31/10/12	P-203AC451	M31	M11	32	-	70	-	70
Total					908	3,086	0	324

Table 5.15 Results of the machine selection and the defect reduction by the proposed system in November

Date	Product Code	Selected Machine (Actual Production)	Selected Machine (Proposed system)		Defectives (Actual)		Reduced Defectives	
			Before Conversion	After Conversion	Short Shot	Black Dot	Short Shot	Black Dot
1/11/12	P158AS	M31	M11	32	41	-	-	-
2/11/12	P158AS	M31	M11	32	-	35	-	35
3/11/12	P158AS	M31	M11	32	-	57	-	57
5/11/12	T022PL70RT-1	M2	M9	2	-	503	-	-
5/11/12	P158AS	M31	M11	32	2	5	-	5
5/11/12	P157AS	M31	M11	32	-	9	-	9

Table 5.15 Results of the machine selection and the defect reduction by the proposed system in November (Cont.)

Date	Product Code	Selected Machine (Actual Production)	Selected Machine (Proposed system)		Defectives (Actual)		Reduced Defectives	
			Before Conversion	After Conversion	Short Shot	Black Dot	Short Shot	Black Dot
6/11/12	5511410098	M1	M9	2	6	-	-	-
6/11/12	T022PL70RT-1	M2	M9	2	-	31	-	-
7/11/12	5621Z3-02	M2	M9	2	2	-	-	-
7/11/12	P157AS	M31	M11	32	2	-	-	31
9/11/12	A024(12)PC60RT-1	M2	M9	2	-	12	-	126
10/11/12	EE57LFTC605511	M33	M3	33	-	39	-	-
12/11/12	EE57LFTC605511	M33	M3	33	-	201	-	61
Total					53	892	6	106

5.5.6 Calculation of the probability of having defective parts from machines that had not reassigned

As the assumption of zero defective in the reassigned machine, the estimation should consider the amount of defectives that regularly occurs although the good machine was selected. The probability of defective parts is the answer. In this case, probability of short shot and black dot parts were calculated from the collections of data during five months.

Probability of having short shot parts from machines that had not reassigned

$$= \frac{\text{No. of short shot parts from the right machine in July}}{\text{Number of total short shot parts}}$$

and

Probability of having black dot parts from machines that had not reassigned

$$= \frac{\text{No. of black dot parts from the right machine in July}}{\text{Number of total black dot parts}}$$

From Table 5.11, 5.12, 5.13, 5.14 and 5.15, total short shot and black dot part could be calculated as follow;

= 16,835 + 676 + 1,747 + 908 + 53

= 20,219 pieces of short shot

and

= 2,979 + 1,275 + 0 + 3,086 + 892

= 8,232 pieces of black dot

Table 5.16 shows the number of short shot part from the machine have not reassigned (Actual = Proposed system) and the machine have reassigned (Actual ≠ Proposed system). The reduced parts by the proposed system were six pieces in November while the other months were zero. These zero part indicates that the proposed system can selected the machine very similar to those by the current system. For black dot parts investigation, Table 5.17, the reduced parts made by the proposed system are higher than short shot. These results support that the proposed system can be used to find out the more appropriate machines than the current system. However, the probability of having defective part had been calculated within these tables for more realistic of our approach. The calculation method will be displayed in the next topic.

Table 5.16 Monthly summary of probability of having short shot parts from machines that had not reassigned

Month	Defective parts (Actual = Proposed system)	Defective parts (Actual ≠ Proposed system)	Reduced defective parts by the proposed system	Probability
July	16,147	688	0	0.799
August	284	384	0	0.014
September	1,698	49	0	0.084
October	598	310	0	0.030
November	2	51	6	9.89×10^{-5}

Table 5.17 Monthly summary of probability of having black dot parts from machines that had not reassigned

Month	Defective parts (Actual = Proposed system)	Defective parts (Actual ≠ Proposed system)	Reduced defective parts by the proposed system	Probability
July	2,247	732	16	0.273
August	122	1,153	313	0.015
September	0	0	0	0
October	2,681	405	324	0.030
November	786	106	106	0.095

Table 5.18 and 5.19 show the percentage of jobs that the selected machine by the proposed system same as planner. There are the numbers of job that the proposed system could select similar with planner exceeded to 70.37%, 66%, 53.85%, 40.82% and 20% in July, October, November, August and September respectively. The high percentages of defective part and jobs when the reassigned machine same as the actual shows the proposed system worked well in July, October and November. And the low percentages do not indicate the bad performance of the proposed system but the proposed system assigned the better machines to those jobs at the same machine size. These machines provide the same of clamping force but lower risk of material contamination. Thus, short shot parts are not reduced like the black dot parts in August and September.

Table 5.18 Monthly summaries of percentages of jobs and defective parts of short shot produced from the reassigned machines same as the actual machines

Month	Defective parts (Actual = Proposed system)	All defective parts	Reduced defective parts by the proposed system	No. of jobs (Actual = Proposed system)	% Defective parts (Actual = Proposed system)	% Jobs (Actual = Proposed system)
July	16,147	16,835	0	38	95.91	70.37
August	284	676	0	20	42.01	40.82
September	1,698	1,747	0	4	97.20	20
October	598	908	0	33	65.86	66
November	2	53	6	7	3.77	53.85

Table 5.19 Monthly summaries of percentages of jobs and defective parts of black dot produced from the reassigned machines same as the actual machines

Month	Defective parts (Actual = Proposed system)	All defective parts	Reduced defective parts by the proposed system	No. of jobs (Actual = Proposed system)	% Defective parts (Actual = Proposed system)	% Jobs (Actual = Proposed system)
July	2,247	2,979	16	38	75.43	70.37
August	122	1,275	313	20	9.57	40.82
September	0	0	0	4	0	20
October	2,681	3,086	324	33	86.88	66
November	786	892	106	7	88.12	53.85

5.5.7 The estimation of the residual defective parts after using the proposed system and performance measurement

The residual defective parts can be calculated as follows;

= (No. of defective parts of short shot in July – No. of reduced defectives) + (No. of short shot parts from the right machine in July x Probability of having short shot parts from machines that had not reassigned)

= (16,835 – 0) + (16,147 x 0.799)

= 29,048.45 ≈ 29,048 pieces

Then, % reduction of defective part can be calculated as follows;

Example:

The % reduction of defective part of short shot in July

= [(No. of defective parts after using proposed system – No. of defective parts before using proposed system) / No. of defective parts before using proposed system] x 100

= [(29,048 – 16,835) / 16,835] x 100 = 72.55%

As the calculated result, the percentage of reduced short shot was positive that mean the new assignment could not reduce the percentage of short shot parts in July. The same results could be gained by other months. However, the percentage of short shot parts would be decreased around 11.32 in November. For the reason why a huge short shot part in July were increased after applying of the proposed system, because the probability of having short shot parts from machines that have not reassigned was higher than other months. For the black dot parts, the percentage of defective parts could be decreased to 24.47 and 3.48 in August and November. All results are summarized in Table 5.20.

Table 5.20 The summary of %reduction of short shot and black dot parts estimation after applying the proposed system from July to November 2012

Month	Short shot (pieces)				Black dot (pieces)			
	Before	After	Δ	%	Before	After	Δ	%
July	16,835	29,048	12,213	72.55	2,979	3,576	597	20.04
August	676	679	3	0.0044	1,275	963	-312	-24.47
September	1,747	1,890	143	8.19	0	0	0	0.00
October	908	925	17	1.87	3,086	3,635	549	17.79
November	53	47	-6	-11.32	892	861	-31	-3.48

From Table 5.18, after applying the probability to the calculation in some month, the defective part seemed to be increased. These results were affected by other causes which have not concerned to machine selection criteria. In other word, errors of selecting the machine are not only factor that influences on short shot or black dot formation. Hence, although good machine had been selected by our proposed system but some defective parts still occurred. Figure 5.11 and 5.12 are the causes of short shot and black dot parts in other factors (e.g. mold, man and method). These diagrams support the above statements. Therefore, the implementation of the proposed system, plastic manufacturers should consider these factors for modifying their system to cover all defects.

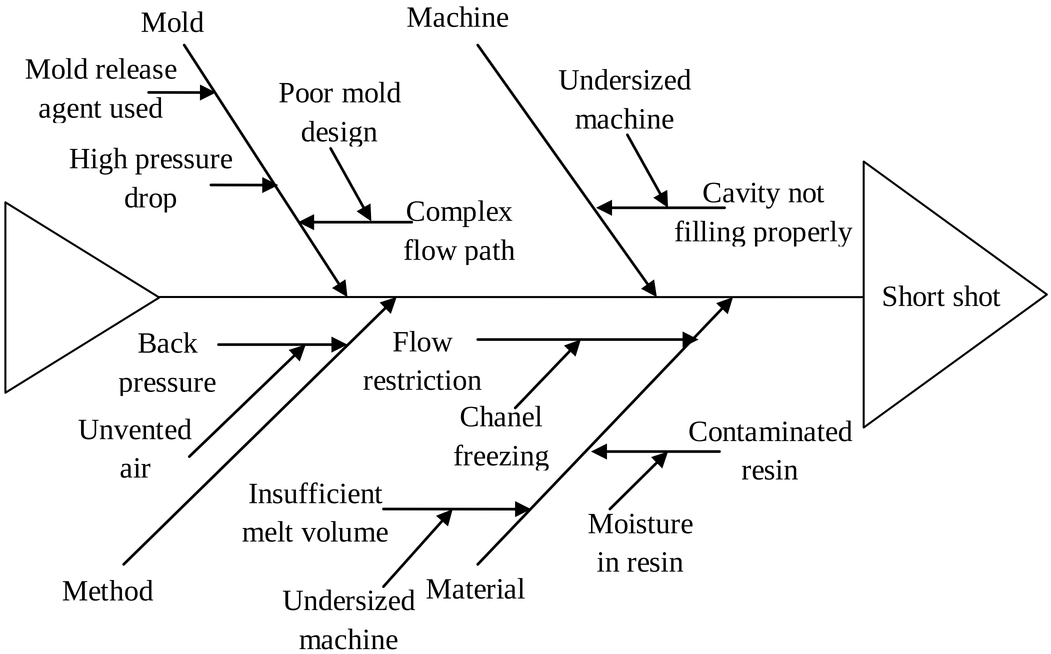


Figure 5.11 Cause and effect diagrams for short shot

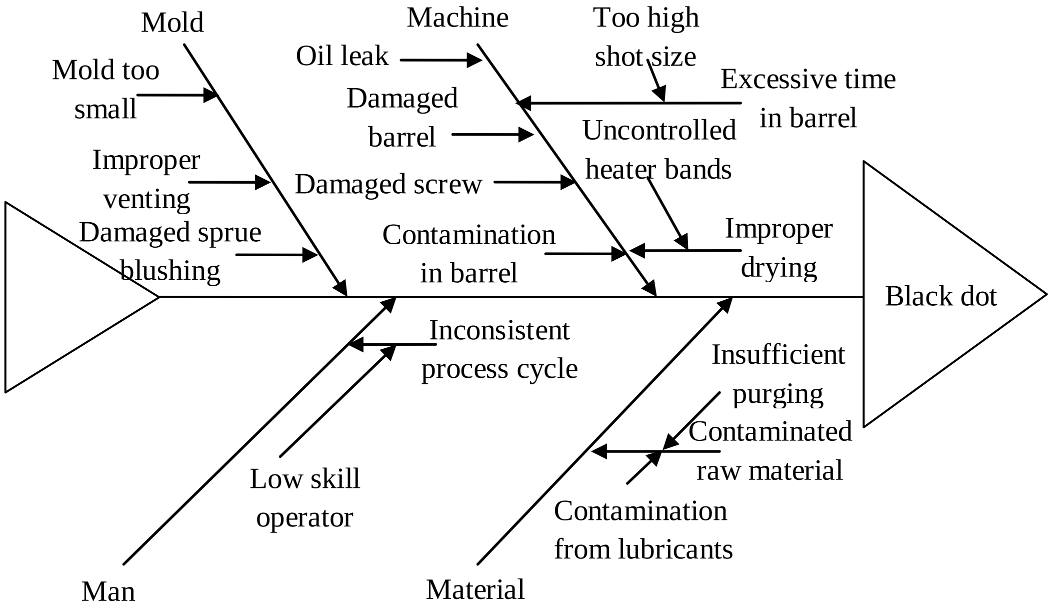


Figure 5.12 Cause and effect diagrams for black dot

5.6 Summary

As the research objectives, this study aims to develop the injection molding machine selection system on MATLAB and improve the efficiency of the machine selection method that influences on computational time and defect reduction as well. As the results, this chapter has contributed to the analysis of computational time reduction as elapsed time comparison between the current machine selection system and the injection molding machine selection system. Under the current system, the average elapsed time in machine selection process has exceeded to 63.45 seconds per a job and the elapsed time by the planner would be increased twice if the second job had been started. Moreover, this process also caused the uncertainty of time which has been exhibited by 26.79 of standard deviation value. On the other hand, the proposed system could maintain its computational time. Although more than 50 jobs had applied to the system, the overall computational time was just a few seconds. The perspective for defect reduction has also analyzed. The calculation had used to substitute actual testing in the plastic industry case study to estimate the residual defective parts after applying of the proposed system. In November, short shot parts may be reduced 11.32% if our system is applied. But there were short shot parts increasing in other months caused by the probability value of having short shot from machines that have not reassigned. Same as the black dot parts, there were defective parts increasing when the proposed system had applied in July and October. However, these defective parts seemed to be decreased to 24.47% and 3.48% in August and November respectively. Meanwhile, there was no change in September. These results show the possibility of more defect reduction if the reassigned machines are increased.
The percentages of jobs and defective parts that are selected by the proposed system are the same as a planner. It indicates that the proposed system could provide a logical machine selection method similar to the current system. But, there are other factors which do not involve to our selection criteria. In other word, errors of selecting the machine are not only a factor that influences on short shot or black dot formation. Our criteria are not enough to reduce all defects. Thus, although a good machine had been selected by this system but some defective parts still occurred. Therefore, the implementation of the proposed system, plastic manufacturers should investigate other concerned factors for modifying their system to cover all requirements.

CHAPTER VI
CONCLUSION AND RECOMMENDATION

This research has studied the development of the automated machine selection system by using fuzzy logic for the plastic injection industry. The major objectives aim to develop the injection molding machine selection system on MATLAB and also to improve the efficiency of the machine selection method in the plastic industry on computational time and defective reduction perspectives. Owing to the fact that effective machine-job assignment is the important issue in injection molding industry. There are common defects caused by inappropriate machine such as black streaking, burn mark, color fading, short shot, etc. Generally, the assignment of the right machine to the right job is done by the experienced planner or an expert. The lower skilled officer can risks inappropriate machine using. In according to handle this situation, having the decision support system is a good choice to enhance the current planning process and decrease human errors with high precision, repeatability, and time saving. Due to the property of fuzzy logic in practice, this study had applied it to develop the injection molding machine selection system.

The process began with the determination of observation and identification problem on shop-floor to collect existing planning operations flowchart, machine selection criteria (e.g. clamping force, distance between tie bars, short weight, product color, and material grade), machine information and specification (related to machine selection criteria), and also monthly defective quantity from July to November 2012. The 11 injection molding machines were used to construct fuzzy inference system. Their range are varied from 30 to 75 tons of machine size with different tie bars and short weight. After that, machine selection framework had been designed. The proposed system is composed of two major systems that are the Fuzzy Inference System (FIS) and System II. The first system aims to find out of the right machine group for jobs and another aims to identify the most suitable machine for job within a machine group. FIS system had constructed by four fuzzy sets of machine selection criteria which are clamping force, distance between tie bar X, distance between tie bar Y and short weight. Fuzzy sets of one output which is machine group had determined. These fuzzy sets of four inputs could generate 384 fuzzy rules. System II obtains the final answer. It is logic of selecting the most suitable machine by concerning of product color and material grade. MATLAB has been used as a system development platform and had been tested in basis of verification and validation of software.

The proposed system can improve the machine selection method obviously. Elapsed time of machine selection process has been reduced from 63.45 seconds to 0.01 second per job. Elapsed time by a planner would be increased twice if the second job had been started. Moreover, this process also caused the uncertainty of time which has been computed by 26.79 of standard deviation value. On the other hand, the proposed system could maintain its computational time. Although more than 50 jobs had applied to the system, the overall computational time was just a few seconds.

The perspective for defective reduction has also analyzed. The calculation approach had used to analyze the residual defective parts after applying of the proposed system instead of actual testing in a plastic industry case study. From defective historical data during July, August, September, October and November 2012, defects that having cumulative percentage at least 80% have been selected to test the system. In November, short shot parts may be reduced at 11.32% if our system was applied. However, there were short shot parts increasing in other months caused by the probability value of having short shot from machines that have not reassigned. Same as the black dot parts, there were defective parts increasing when the proposed system had applied in July and October. These defective parts seemed to be decreased to 24.47% and 3.48% in August and November respectively. Meanwhile, there was no changed in September. From the results that have mentioned, there is possibility of defect reduction if the reassigned machines were increased. Nevertheless, the increased parts shows that the errors of selecting the machine are not only factor that influences on short shot or black dot formation. Hence, good machine had been selected by our proposed system but some defective parts still occurred. Therefore, the implementation of the proposed system, plastic manufacturers should consider these factors for modification their system to cover all defects. The implementation in other plastic manufacturers should consider their machine selection operations and criteria particularly. Due to, the priority of their criteria are generally different and number of injection molding machine as well. Having not a lot of machines, some manufacturers can limit their fuzzy rules and number of system unit another one whilst have to carry on more huge system.

However, the results of the percentages of jobs and defective parts that the selected machine by the proposed system same as a planner indicated that the proposed system could provide machine selection logical similar to the current system in the plastic industry case study. The proposed system is the practical decision support tool for selecting injection molding machine in the plastic industry.

This study has been developed with the same sets of machine selection criteria since 2012. Our study had used only six priority selection criteria to be the case-study and illustrated fuzzy logic concept application for injection molding machine selection. There are some other criteria that still be used in actual production such as screw diameter, labor, etc. Thus, a user should consider updating of the software operations along with changing criteria. Fuzzy inference system toolbox provides the revising the software with practicable, realistic, visible graphical user interface and safer time. Unfortunately, broad applications may be limited caused by high investment of licensed MATLAB program.

REFERENCES

Akgun, A., Sezer, E.A., Nefeslioglu, H.A., Gokceoglu, C., Pradhan, B. (in press) "An easy-to-use MATLAB program (MamLand) for the assessment of landslide." Computers & Geosciences.

Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y. (2008) "A survey of scheduling problems with setup times or costs." European Journal of Operational Research 187(3): 985–1032.

Al-Hawari, T., Khrais, S., Al-Araidah, O., Al-Dwairi, A. F. (2010) "2D laser scanner selection using fuzzy logic." Expert Systems with Applications.

Arslan, M.C., Çatay, B., Budak, E. (2004) "A decision support system for machine tool selection." Journal of Manufacturing Technology Management 15(1): 101-109.

Bozzelli, J. (2007) "Sticking sprues or parts? lots of possible causes and solutions." Available at http://www.ptonline.com/articles/sticking-sprues-or-parts-lots-of-possible-causes-and-solutions, accessed on 3 April 2014.

Bown, J. (1979) "Injection molding of plastic components." Mcraw-Hill Book Company (UK) Ltd., United Kingdom.

Bozdana, A.T., Eyercioğlu, Ö. (2002) "Development of an expert system for the determination of injection." Journal of Materials Processing Technology 128: 113–122.

Byun, H.S., Lee, K.H. (2005) "A decision support system for the selection of a rapid prototyping process using the modified TOPSIS method." International Journal of Advanced Manufacturing Technology 26: 1338–1347.

Cao, D., Chen, M., Wan, G. (2005) "Parallel machine selection and job scheduling to minimize machine cost and job tardiness." Computer & Operations Research 32: 1995–2012.

Çimren, E., Çatay, B. and Budak, E. (2004) "Development of a machine tool selection system using analytic hierarchy process." In: 4th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering.

Chan, F.T.S., Swarnkar, R. (2006) "Ant colony optimization approach to a fuzzy goal programming model for a machine tool selection and operation allocation problem in an FMS." Robotics and Computer-Integrated Manufacturing, 22: 353–362.

Chen, J.F. (2006) "Minimization of maximum tardiness on unrelated parallel machines with process restrictions and setups." International Journal of Advanced Manufacturing Technology 29: 557-563.

Chtourou, H., Masmoudi, W., Maalej, A. (2005) "An expert system for manufacturing systems machine selection." Expert Systems with Application 28: 461–467.

Chu, T.C., Lin, Y.C. (2003) "A Fuzzy TOPSIS Method for Robot Selection." International Journal of Advanced Manufacturing Technology 21:284–290.

Chujo, H., Oka, H., Ikkai, Y., Komoda, N. (2005) "A real-time production scheduling method using attractor selection." Computational Intelligence for Modelling, Control and Automation, International Conference on Intelligent Agents, Web Technologies and Internet Commerce. 1, 511.

Dastidar, S.G., Nagi, R. (2005) "Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs." Computer & Operations Research 32, 2987-3005.

Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R. (1979) "Optimization and approximation in deterministic sequencing and scheduling: a survey." Annals of Discrete Mathematics 5, 287-326.

Gordon, M.J. (2010) "Total quality process control for injection molding." 2nd edition, A John Wiley & Sons, Inc., Hoboken, NJ, USA, 2010.

Guiffrida, A.L., Nagi, R. (1998) "Fuzzy set theory applications in production management research: a literature survey." Journal of Intelligent Manufacturing 9, 39-56.

Guldogan, E.U. (2011) "An integrated approach to machine selection and operation allocation problem." International Journal of Advanced Manufacturing Technology 55:797–805.

Hanna, A.S., Lotfallah, W.B. (1999) "A fuzzy logic approach to the selection of cranes." Automation in Construction 8: 597–608.

Harris, B., Lewis, F., Cook, D.J. (1998) "Machine planning for manufacturing: dynamic resource allocation and on-line supervisory control." Journal of Intelligent Manufacturing 9: 413-430.

Huang, J., Süer, G.A., Urs, S.B.R. (2011) "Genetic algorithm for rotary machine scheduling with dependent processing times." Journal of Intelligence Manufacturing.

IEEE. (2005) "IEEE standard for software verification and validation." Available at http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=9958, accessed on 16 January 2014.

İç, Y.T., Yurdakul, M. (2009) "Development of a decision support system for machining center selection." Expert Systems with Applications 36: 3505–3513.

Krogt, R., Little, J. (2009) "Optimising machine selection rules for sequence dependent setups with an application to cartoning." Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing 1148-1153.

Kulak, O., Durmuso˘glu, M.B. and Kahraman, C. (2005) "Fuzzy multi-attribute equipment selection based on information axiom." Journal of Materials Processing Technology 169:337–345.

Kwong, C.K., Smith, G.F., Lau, W.S. (1997) "Application of Case Based Reasoning in Injection Moulding." Journal ofMaterials Processing Technology 63: 463-467.

Lababidi, H.M.S., Baker, C.G.J. (1999) "An expert system for dryer selection using fuzzy logic." Computers and Chemical Engineering Supplement 691-694.

Li, P.C., Chen, G.H., Dai, L.C., Zhang, L. (2010) "Fuzzy logic-based approach for identifying the risk importance of human error." Safety Science 48: 902–913.

Lin, J., Lian, R.J. (2010) "Self-organizing fuzzy controller for injection molding machines." Journal of Process Control 20: 585–595.

Lin, Z.C., Yang, C.B. (1996) “Evaluation of machine selection by the AHP method.” Journal of Materials Processing Technology 57: 253-258.

Ma, J., Chen, S. Xu, Y. (2006) “Fuzzy logic from the viewpoint of machine intelligence.” Journal of Fuzzy Sets and Systems 157(5): 628-634.

Mahanty, R.N., Gupta D.P.B. (2007) “A fuzzy logic based fault classification approach using current samples only.” Electric Power Systems Research 77: 501–507.

Metaxiotis, K., Psarras, J., Samouilidis, E. (2003) “Integrating fuzzy logic into decision support systems: current research and future prospects.” Information Management & Computer Security 11(2): 53-59.

Mok, S.L., Kwong, C.K. (2002) “Application of artificial neural network and fuzzy logic in a case-based system for initial process parameter setting of injection molding.” Journal of Intelligent Manufacturing 13: 165-176.

Mula, J., Poler, R., Garcı´a-Sabater, J.P., Lario, F.C. (2006) “Models for production planning under uncertainty: a review.” International Journal of Production Economics 103: 271–285.

Nagarur, N., Vrat, P., Duongsuwan, W. (1997) “Production planning and scheduling for injection moulding of pipe fittings: A case study.” Internatinal Journal of Production Economics 53: 157–170.

Nagahanumaiah, Subburaj, K., Ravi, B. (2008) “Computer aided rapid tooling process selection and manufacturability evaluation for injection mold development.” Computers in Industry 59: 262–276.

Nakason, C. (2004) “Plastic processing” Prince of Songkla University, 3rd edition, Four Press, Bangkok. (In Thai)

Negnevitsky, M. (2005). “Artificial intelligence: a guide to intelligent systems.” 2nd edition. Pearson Education Limited, England.

Nowicki, E., Zdrzałka, S. (1990) “A survey of results for sequencing problems with controllable processing times.” Discrete Applied Mathematics 26(2-3): 271-287.

Önüt, S., Kara, S.S., Efendigil, T. (2008) “A hybrid fuzzy MCDM approach to machine tool selection.” Journal of Intelligence Manufacturing 19: 443–453.

Pravadalioglu, S. (2005) “Single-chip fuzzylogic controller design and an application on a permanent magnet dc motor.” Engineering Applications of Artificial Intelligence 18: 881–890.

Pötsch, G. Michaeli, W. (2008) “Injection molding.” Available at http://www.beck-shop.de/fachbuch/leseprobe/9783446406353_Excerpt_001.pdf, accessed on 24 August 2013.

Rao, R.V. (2007) “Machine Selection in a Flexible Manufacturing Cell.” Decision making in the manufacturing environment-using graph theory and fuzzy multiple attribute decision making methods. Springer London. Part 2, 139-148.

Sa-Nguansatra, P. (2010) “MATLAB Handbook” IDC Premier, Nonthaburi. (In Thai)

Sivanandam, S.N., Sumathi, S., Deepa, S.N. (2007) “Introduction to Fuzzy Logic using MATLAB.” Springer-Verlag Berlin, Heidelberg, Germany.

Shabtay, D., Steiner, G. (2007) “A survey of scheduling with controllable processing times.” Discrete Applied Mathematics 155(13): 1643 – 1666.

Shelesh-Nezhad, K., Siores, E. (1997) “An Intelligent System for Plastic Injection Molding Process Design.” Journal of Materials Processing Technology 63: 458-462.

Sterna, M. (2011) “A survey of scheduling problems with late work criteria.” Omega 39: 120-129.

Subramaniam, V., Lee, G. K., Ramesh, T., Hong, G. S. and Wong, Y. S. (2000) “Machine selection rules in a dynamic job shop.” International Journal of Advanced Manufacturing Technology 16(12): 902-908.

Subramaniam, V., Ramesh, T. and Raheja, A.S. (2002) “Exploiting the flexibility of multiple job routes in a Dynamic Job Shop.” Studies in informatics and control, 11(2).

Subramaniam, V., Lee, G.K., Hong, G.S., Wong, Y.S., and Ramesh, T. (2002) "Dynamic Selection of Dispatching Rules for Job Shop Scheduling." Production Planning and Control 11(1): 73-78.

Tabucanon et al. (1994) “Decision support system for multicriteria machine selection for flexible manufacturing systems.” Computer in Industry 25: 131-143.

Tanev, I.T, Uozumi, T., Morotome, Y. (2004) "Hybrid evolutionary algorithm-based real-world flexible job shop scheduling problem: application service provider approach." Applied Soft Computing 5: 87-100.

Tansel, Y., Yurdakul, M. (2009) "Development of a decision support system for machining center selection." Expert Systems with Applications 36: 3505–3513.

Tsai, J.P, Cheng, H.Y, Wang, S.Y, Kao, Y.C. (2010) "Multi-criteria decision making method for selection of machine tool." 2010 International Symposium on Computer Communication Control and Automation (3CA).

Tsekouras, G., Sarimveis, H., Raptis, C., Bafas, G. (2002) "A fuzzy logic approach for the classification of product qualitative characteristics." Computers and Chemical Engineering 26: 429–438.

Urs, S.B, (2005) "Scheduling rotary injection molding machine." Unpublished master's thesis, Ohio University.

Wang, T.Y., Shaw, C.F., Chen, Y.L. (2000) "Machine selection in flexible manufacturing cell: a fuzzy multiple attribute decision-making approach." International Journal of Production Research 38(9): 2079-2097.

Wang, Y.D., Chang, P.Q., Wu, Y.F., Hwang, S. J., Hwang, D. Y., Cheng, R.W. (2005) "Design and fabrication of an all-electric tiebarless injection molding machine." Proceedings of the IEEE International Conference on Mechatronics 783 – 787.

Wen, K.L. (2008) "A Matlab toolbox for grey clustering and fuzzy comprehensive evaluation." Advances in Engineering Software 39: 137–145.

Wong, B.K., Lai, V.S., (2011) "A survey of the application of fuzzy set theory in production and operations management: 1998–2009." International Journal of Production Economics 29(1): 157-168.

Yen, J., Langari, R., Zadeh, L.A. (1994) "Industrial Applications of Fuzzy Logic and Intelligent Systems." New York, USA.

Yurdakula, M.& Iç, Y. T. (2009) "Analysis of the benefit generated by using fuzzy numbers in a TOPSIS model developed for machine tool selection problems. Journal of Materials Processing Technology 209: 310–317.

Yusoff, S. M. M, Rohani, J. M., Hamid, W. H. W. & Ramly, E. (2004) A plastic injection molding process characterisation using experimental design technique: a case study. Jurnal Teknologi, 41(A): 1–16.

APPENDICES

APPENDIX A

SUGENO - STYLE FUZZY INFERENCE STRUCTTURE

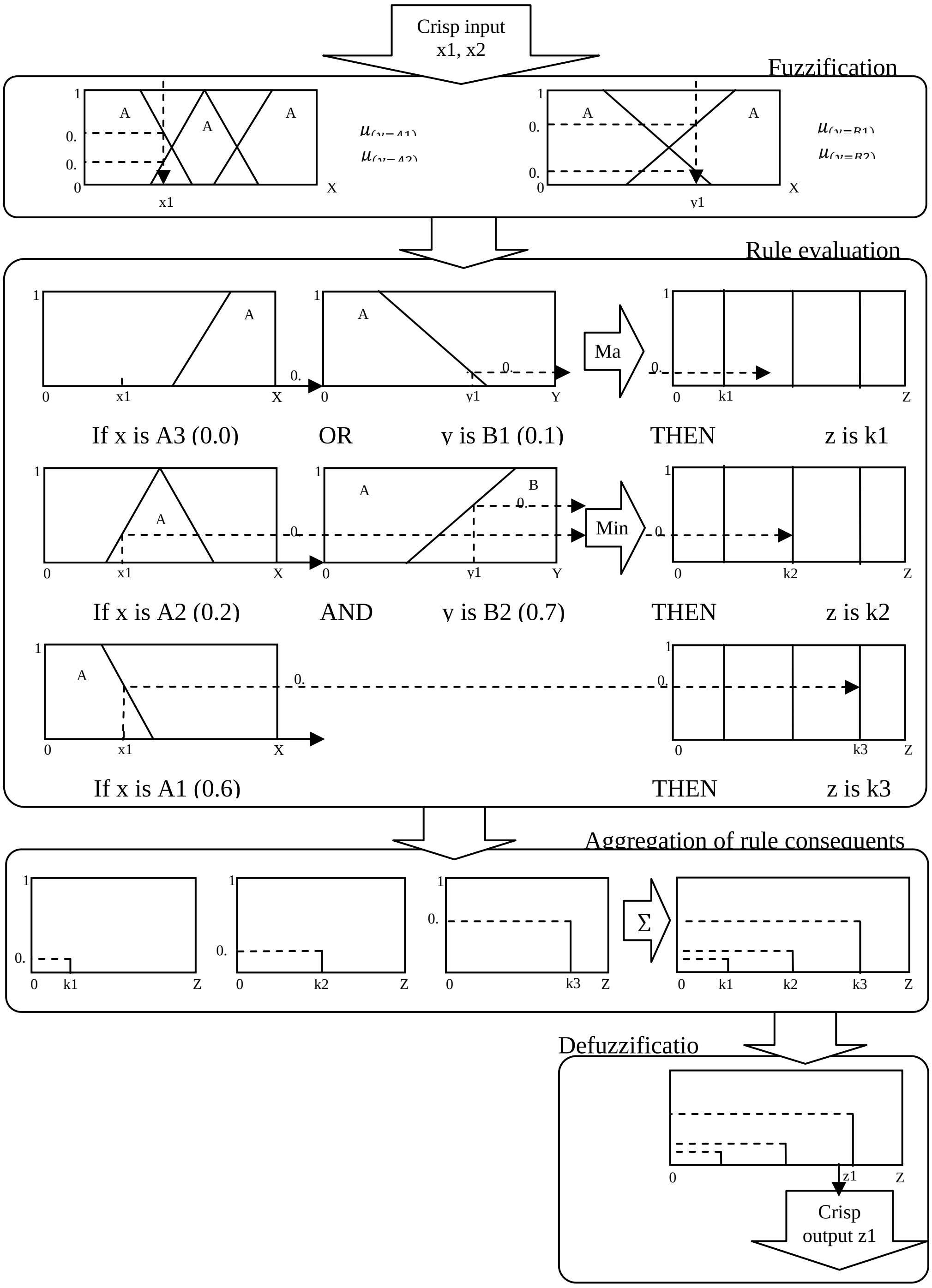


Figure A.1 Sugeno-style fuzzy inference structure (Negnevitsky, 2005)

APPENDIX B
INJECTION MOLDING MACHINE SPECIFICATION

This research had selected 11 of 35 injection molding machines from the plastic industry case study. These machine sizes are included of 30 to 650 tons. Table A.1 presents the collected data on all 35 injection molding machines from the plastic industry case study.

Table B.1 The 35 injection molding machines from the plastic industry case study

M/c No.	Machine	Clamping Force (Ton)	Distance Between Tie Bars (mm)	Shot Weight (g)
1	JSW J50 EII	50	310x310	56
2	JSW J55 EII	55	310x310	23
3	NIGATA NN50E	50	310x300	60
4	TOSHIBA IS50EP-1.5A	50	310x310	56
5	TOSHIBA IS30EPN-1Y	30	260x260	28
6	JSW J30 ED	30	260x260	28
7	MEIKI 650T.	650	945x945	2225
8	TOSHIBA 550T.	550	880x880	1400
9	JSW J350 EII	350	730x730	810
10	JSW J350 EII	350	730x730	810
11	JSW J280 EII	280	630x630	810
12	TOSHIBA IS 80 G-2A	80	375x375	105
13	TOSHIBA IS 80 ENP-5A	80	375x375	105
14	TOSHIBA IS100 E	100	410x410	150
15	TOSHIBA IS100 EN-3A	100	410x410	150
16	JSW J110 ELII	110	410x410	160
17	TOSHIBA IS100 G-3A	100	410x410	150
18	TOSHIBA IS130 G-5A	130	510x460	230
19	SUMITOMO SG150 C450M	150	510x510	191
20	TOSHIBA IS170 GN-7A	170	560x510	325
21	HAITIAN HTF 200X	200	510x510	375

Table B.1 The 35 injection molding machines from the plastic industry case study (Cont.)

M/c No.	Machine	Clamping Force (Ton)	Distance Between Tie Bars (mm)	Shot Weight (g)
22	TOSHIBA IS220 GN-10A	220	610x560	450
23	HAITIAN HTF 250X	250	570x570	487
24	TOSHIBA 650	650	950x950	2225
25	HWE 880 KN-F3	80	320x320	50
26	HWE 1200 KN-F3	120	395x365	150
27	HWE 1200 KN-F3	120	396x365	150
28	HWE 2380 KN-F4	240	525x520	480
29	NIIGATA	100	400x400	130
30	NIIGATA	100	400x400	130
31	NIIGATA	75	350x380	100
32	NIIGATA	75	350x380	100
33	NIIGATA	50	300x300	45
34	NIIGATA	50	300x300	45
35	NIIGATA	50	300x300	45

APPENDIX C
FUZZY RULES

There are all 384 of fuzzy inference rules in MCselection.fis file. All rules are as follows:

1. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G1) (1)
2. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G1) (1)
3. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G1) (1)
4. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G1) (1)
5. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G1) (1)
6. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G1) (1)
7. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G1) (1)
8. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G1) (1)
9. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G1) (1)
10. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G1) (1)
11. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G1) (1)

12. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G1) (1)
13. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G1) (1)
14. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G1) (1)
15. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G1) (1)
16. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G1) (1)
17. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G1) (1)
18. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G1) (1)
19. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G1) (1)
20. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G1) (1)
21. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G1) (1)
22. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G1) (1)
23. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G1) (1)
24. If (ClampingForce is Verylow) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G1) (1)
25. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G1) (1)
26. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G1) (1)
27. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G1) (1)

28. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G1) (1)
29. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G1) (1)
30. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G1) (1)
31. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G1) (1)
32. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G1) (1)
33. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G1) (1)
34. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G1) (1)
35. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G1) (1)
36. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G1) (1)
37. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G1) (1)
38. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G1) (1)
39. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G1) (1)
40. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G1) (1)
41. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G1) (1)
42. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G1) (1)
43. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G1) (1)

44. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G1) (1)
45. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G1) (1)
46. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G1) (1)
47. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G1) (1)
48. If (ClampingForce is Verylow) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G1) (1)
49. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G2) (1)
50. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G2) (1)
51. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G2) (1)
52. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G2) (1)
53. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G2) (1)
54. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G2) (1)
55. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G2) (1)
56. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G2) (1)
57. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G2) (1)
58. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G2) (1)
59. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G2) (1)

60. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G2) (1)
61. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G2) (1)
62. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G2) (1)
63. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G2) (1)
64. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G2) (1)
65. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G2) (1)
66. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G2) (1)
67. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
68. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
69. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
70. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
71. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
72. If (ClampingForce is Verylow) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
73. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
74. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G6) (1)
75. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G6) (1)

76. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G6) (1)
77. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G6) (1)
78. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
79. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
80. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G6) (1)
81. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G6) (1)
82. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G6) (1)
83. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G6) (1)
84. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
85. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
86. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G6) (1)
87. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G6) (1)
88. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G6) (1)
89. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G6) (1)
90. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
91. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)

92. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
93. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
94. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
95. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
96. If (ClampingForce is Verylow) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
97. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G2) (1)
98. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G2) (1)
99. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G2) (1)
100. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G2) (1)
101. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G2) (1)
102. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G2) (1)
103. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G3) (1)
104. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G3) (1)
105. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G3) (1)
106. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G3) (1)
107. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G3) (1)

108. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G3) (1)
109. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G3) (1)
110. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G3) (1)
111. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G3) (1)
112. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G3) (1)
113. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G3) (1)
114. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G3) (1)
115. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
116. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
117. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
118. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
119. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
120. If (ClampingForce is Low) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
121. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G2) (1)
122. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G2) (1)
123. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G2) (1)

124. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G2) (1)
125. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G2) (1)
126. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G2) (1)
127. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G2) (1)
128. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G2) (1)
129. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G2) (1)
130. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G2) (1)
131. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G2) (1)
132. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G2) (1)
133. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G3) (1)
134. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G3) (1)
135. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G3) (1)
136. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G3) (1)
137. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G3) (1)
138. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G4) (1)
139. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)

140. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
141. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
142. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
143. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
144. If (ClampingForce is Low) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
145. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G4) (1)
146. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G4) (1)
147. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G4) (1)
148. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G4) (1)
149. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G4) (1)
150. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G4) (1)
151. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G4) (1)
152. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G4) (1)
153. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G4) (1)
154. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G4) (1)
155. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G4) (1)

156. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G4) (1)
157. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G3) (1)
158. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G3) (1)
159. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G3) (1)
160. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G3) (1)
161. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G3) (1)
162. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G3) (1)
163. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
164. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
165. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
166. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
167. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
168. If (ClampingForce is Low) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
169. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
170. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G6) (1)
171. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G6) (1)

172. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G6) (1)
173. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G6) (1)
174. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
175. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
176. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G6) (1)
177. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G6) (1)
178. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G6) (1)
179. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G6) (1)
180. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
181. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
182. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G6) (1)
183. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G6) (1)
184. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G6) (1)
185. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G6) (1)
186. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
187. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)

188. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
189. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
190. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
191. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
192. If (ClampingForce is Low) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
193. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G5) (1)
194. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G5) (1)
195. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G5) (1)
196. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G5) (1)
197. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G5) (1)
198. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G5) (1)
199. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G5) (1)
200. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G5) (1)
201. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G5) (1)
202. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G5) (1)
203. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G5) (1)

204. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G5) (1)
205. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G5) (1)
206. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G5) (1)
207. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G5) (1)
208. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G5) (1)
209. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G5) (1)
210. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G5) (1)
211. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G5) (1)
212. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G5) (1)
213. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G5) (1)
214. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G5) (1)
215. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G5) (1)
216. If (ClampingForce is Medium) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G5) (1)
217. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G5) (1)
218. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G5) (1)
219. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G5) (1)

220. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G5) (1)
221. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G5) (1)
222. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G5) (1)
223. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G5) (1)
224. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G5) (1)
225. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G5) (1)
226. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G5) (1)
227. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G5) (1)
228. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G5) (1)
229. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G5) (1)
230. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G5) (1)
231. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G5) (1)
232. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G5) (1)
233. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G5) (1)
234. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G5) (1)
235. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)

236. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
237. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
238. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
239. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
240. If (ClampingForce is Medium) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
241. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G5) (1)
242. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G5) (1)
243. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G5) (1)
244. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G5) (1)
245. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G5) (1)
246. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G5) (1)
247. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G5) (1)
248. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G5) (1)
249. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G5) (1)
250. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G5) (1)
251. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G5) (1)

252. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G5) (1)
253. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G5) (1)
254. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G5) (1)
255. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G5) (1)
256. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G5) (1)
257. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G5) (1)
258. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G5) (1)
259. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
260. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
261. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
262. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
263. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
264. If (ClampingForce is Medium) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
265. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
266. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G6) (1)
267. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G6) (1)

268. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G6) (1)
269. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G6) (1)
270. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
271. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
272. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G6) (1)
273. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G6) (1)
274. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G6) (1)
275. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G6) (1)
276. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
277. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
278. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G6) (1)
279. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G6) (1)
280. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G6) (1)
281. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G6) (1)
282. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
283. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)

284. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
285. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
286. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
287. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
288. If (ClampingForce is Medium) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
289. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
290. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G6) (1)
291. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G6) (1)
292. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G6) (1)
293. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G6) (1)
294. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
295. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
296. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G6) (1)
297. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G6) (1)
298. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G6) (1)
299. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G6) (1)

300. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
301. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
302. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G6) (1)
303. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G6) (1)
304. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G6) (1)
305. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G6) (1)
306. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
307. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
308. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
309. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
310. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
311. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
312. If (ClampingForce is High) and (TiebarX is Verylow) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
313. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
314. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G6) (1)
315. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G6) (1)

316. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G6) (1)
317. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G6) (1)
318. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
319. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
320. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G6) (1)
321. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G6) (1)
322. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G6) (1)
323. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G6) (1)
324. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
325. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
326. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G6) (1)
327. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G6) (1)
328. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G6) (1)
329. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G6) (1)
330. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
331. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)

332. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
333. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
334. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
335. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
336. If (ClampingForce is High) and (TiebarX is Low) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
337. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
338. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G6) (1)
339. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G6) (1)
340. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G6) (1)
341. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G6) (1)
342. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
343. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
344. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G6) (1)
345. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G6) (1)
346. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G6) (1)
347. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G6) (1)

348. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
349. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
350. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G6) (1)
351. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G6) (1)
352. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G6) (1)
353. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G6) (1)
354. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
355. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
356. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
357. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
358. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
359. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
360. If (ClampingForce is High) and (TiebarX is Medium) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
361. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
362. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Verylow) then (MCgroup is G6) (1)
363. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Low) then (MCgroup is G6) (1)

364. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Medium) then (MCgroup is G6) (1)
365. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is High) then (MCgroup is G6) (1)
366. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Verylow) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
367. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
368. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Verylow) then (MCgroup is G6) (1)
369. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Low) then (MCgroup is G6) (1)
370. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Medium) then (MCgroup is G6) (1)
371. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is High) then (MCgroup is G6) (1)
372. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Low) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
373. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)
374. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Verylow) then (MCgroup is G6) (1)
375. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Low) then (MCgroup is G6) (1)
376. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Medium) then (MCgroup is G6) (1)
377. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is High) then (MCgroup is G6) (1)
378. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is Medium) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)
379. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Veryverylow) then (MCgroup is G6) (1)

380. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Verylow) then (MCgroup is G6) (1)
381. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Low) then (MCgroup is G6) (1)
382. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Medium) then (MCgroup is G6) (1)
383. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is High) and (Shotwt is High) then (MCgroup is G6) (1)
384. If (ClampingForce is High) and (TiebarX is High) and (TiebarY is High) and (Shotwt is Veryhigh) then (MCgroup is G6) (1)

APPENDIX D
PRODUCT SPECIFICATION DATABASE

Injection molding machine selection system always shows product specification database which is linked with Product Specification Database.xlsx. Table D.1 shows all data from that file.

Table D.1 Product specification database

Product Codes	Clamping Force	Distance between Tie Bar X	Distance between Tie Bar Y	Short Weight	Material Grade	Product Color
P5511410098	55	310	310	23	A	Black
P00-54016	55	310	310	23	A	White
1551P410098	55	310	310	23	A	Black
5621Z3-02	55	310	310	23	A	Black
5625Z3	50	310	310	56	C	Black
563DR-W	50	300	300	45	C	White
A024(12)PC60RT-1	55	310	310	23	A	Black
D159AS	75	350	350	100	A	Black
E26P0LFBJ-02063	50	310	310	56	A	Gray
EE33LFTC802001	50	300	300	45	C	Black
EE33LFTC802011	35	300	300	45	C	Black
EE33LFTC802031	50	310	310	56	C	Clear
EE33LFTC905315	50	310	310	56	C	Clear
EE33LFTC905511	50	310	310	56	C	Clear
EE57LFTC510712	50	300	300	45	A	Black
EE57LFTC604812	50	310	310	56	A	White
EE57LFTC605511	50	300	300	45	A	White
F26P1LFDN-00095	50	320	320	50	C	Gray
F26P1LFDN-00128	50	300	300	45	D	White
F26P1LFDN-00134	50	310	310	56	C	Clear

Table D.1 Product specification database (Cont.)

Product Codes	Clamping Force	Distance between Tie Bar X	Distance between Tie Bar Y	Short Weight	Material Grade	Product Color
F26P1LFDN-00136	30	310	310	23	A	White
F26P1LFDN-00145	30	260	260	28	C	Black
F26P1LFDN-00277	75	350	350	100	A	Grey
F26P1LFDN-0095	50	320	320	50	C	Gray
F26P1LFDN-136	30	260	260	28	A	White
IH55LFMP066-02	50	310	310	56	C	Clear
K41PC5B	75	350	380	100	C	Clear
KN66JF8A	75	350	380	100	D	Clear
P128AS	75	350	350	100	A	Grey
P129AS	75	350	350	100	A	White
P12A9L	75	350	350	100	A	Black
P12AS8AL	75	350	350	100	A	Gray
P157AS	75	350	380	100	C	White
P158AS	75	350	380	100	C	White
P159AS	75	350	350	100	A	White
P15AS8	75	350	350	100	A	White
P-203AC451	75	350	380	100	C	White
P537ASB	50	300	300	45	A	White
P5511410098	55	310	310	23	A	Black
P-984DZ7A	55	310	310	23	A	Clear
PD12AS8	75	350	350	100	A	Clear
PD1AS28AL	75	350	350	100	A	Gray
PD1AS29L	75	350	350	100	A	Black
PU1AS16	75	350	350	100	A	Black
PU1AS16(32)	75	350	380	100	A	Black
PU1AS28AL	75	350	350	100	A	Black
PU1AS59	75	350	350	100	A	Black
PU-4AS3A	75	350	380	100	C	White
PU53AS7B	50	300	300	45	A	Black
R411LFTH23826C	50	310	310	56	C	Clear
R411LFTH23828F	50	310	300	60	D	Black
R411LFTH23829C	50	300	300	45	C	Black

Table D.1 Product specification database (Cont.)

Product Codes	Clamping Force	Distance between Tie Bar X	Distance between Tie Bar Y	Short Weight	Material Grade	Product Color
R411LFTH23857B	50	310	310	56	D	Black
R411LFTH23887B	50	310	310	56	C	Clear
R411LFTH24263C	30	260	260	28	A	Black
T022PL70RT-1	55	310	310	23	A	Black
U41PC5B	75	350	380	100	C	Clear

BIOGRAPHY

NAME	Ms. Sirinut Suwannasri
DATE OF BIRTH	2 April 1985
PLACE OF BIRTH	Nakhon Nayok, Thailand
INSTITUTIONS ATTENDED	Suranaree University of Technology, 2003-2006 Bachelor of Science (Food Technology) Mahidol University, 2009-2013 Master of Engineering (Industrial Engineering)
HOME ADDRESS	191 M. 4 T. Pak Phli, Pak Phli District, Nakhon Nayok, 26130 Tel. 084-437-9876 E-mail : sirinut.swns@gmail.com
PUBLICATION / PRESENTATION	Sirinut Suwannasri, Thitikorn Limchimchol and Anchalee Dejpunthavanich, 2010. The Structural Model of Multi-Agent System Application on Production Planning in Plastic Injection Manufacturing, Proceedings of the National Conference on Industrial Engineering 2010, Bangkok, December 16-17, 2010. Sirinut Suwannasri, Ronnachai Sirovetnukul and Thitikorn Limchimchol, 2012. An Automated Plastic Injection

PUBLICATION / PRESENTATION

Molding Machine Selection System based on Fuzzy Logic using MATLAB, Proceedings of the International Conference on Business and Industrial Research 2012: ICBIR 2012, Bangkok, May 17-18, 2012.

Sirinut Suwannasri and Ronnachai Sirovetnukul, 2013, “The Defects Reduction in Injection Molding by Fuzzy Logic based Machine Selection System, Proceeding of the International Conference on Industrial Engineering and System 2013: ICIESM 2013, Barcelona, Spain, February 27-28, 2013.