**國 立 中 央 大 學**

**工 業 管 理 研 究 所**

**碩 士 論 文**

Applying Genetic Algorithm to Schedule Brewery Production

研 究 生：Bui Xuan Toan

指導教授：王啟泰 博士

中 華 民 國 一 ○ 六 年 二 月

Abstract

Scheduling is one of the most important problems in any manufacturing industry. Therefore, the problem has been studied extendedly. Since, the scheduling problem is classified as NP-hard problem, which means the time required for finding the optimal solution of the problem is grown exponentially with the size of the problem. Therefore, it is unrealistic to find optimal solution for the scheduling problem in the scene of the real world industrial case, even with today advanced computer system.

There are many heuristic algorithms have been proposal to solve the scheduling problem. They are beam search, local search technique, tabular search and Genetic Algorithm (GA), to name a few. In recent years, GA has become a noticeable candidate for solving the scheduling problem effectively. The idea of mimicking the evolutionary process is very interesting to researchers. And the recent advanced in heuristic GA has sparked more attention toward new research and application in the field of GA.

In Brewery industry, the fermentation process is the most crucial components of the whole manufacturing process. It will decide the quality, taste of the products as well as the productivity of the production line. Since, the fermentation time can take up to 41 days, and the requirement time is varying a lot between different types of beers, therefore finding a good scheduling solution to dealing with this complexity is crucial for beer manufacturers. This research will propose a GA to solve the scheduling problem in beer production. The proposed methodology will serve as a planning and analysis tool to utilize assets (tanks, filling lines) effectively, reduce congestion and synchronize the production process between the two production stages (liquid preparation and bottling).

**Keywords:** scheduling, lot sizing, brewery industry, two-stage production, GA.

Table of Contents

[Abstract i](#_Toc478563073)

[Table of Contents ii](#_Toc478563074)

[Table of Figures iv](#_Toc478563075)

[List of Tables vi](#_Toc478563076)

[CHAPTER 1. Introduction 1](#_Toc478563077)

[1.1 Motivations 1](#_Toc478563078)

[1.2 Objectives 2](#_Toc478563079)

[1.3 Research Framework 3](#_Toc478563080)

[CHAPTER 2. Literature Review 4](#_Toc478563081)

[2.1 Scheduling Problems in Industrial Management 4](#_Toc478563082)

[2.1.1 The Job-Shop Scheduling Problem 5](#_Toc478563083)

[2.1.2 The Flexible Job-Shop Scheduling Problem. 7](#_Toc478563084)

[2.1.3 The Integrated Operation Sequence and Resource Selection Problem: 8](#_Toc478563085)

[2.1.4 The Scheduling Problem in Soft Drink and Brewery Industry. 10](#_Toc478563086)

[2.2 GA Approach 13](#_Toc478563087)

[2.2.1 GA in General 13](#_Toc478563088)

[2.2.2 GA and Network Modeling 15](#_Toc478563089)

[CHAPTER 3. Brewery Production and Problem Description 21](#_Toc478563090)

[3.1 Beer Production Process 21](#_Toc478563091)

[3.2 Problem Description and Modeling 22](#_Toc478563092)

[CHAPTER 4. GA Approach 28](#_Toc478563093)

[4.1 Proposed Approach 28](#_Toc478563094)

[4.2 Proposed Time Line 29](#_Toc478563095)

[References 31](#_Toc478563096)

Table of Figures

[Figure 1: Typical scheduling problems (Zhang et al. 2006) 5](#_Toc484289276)

[Figure 2: Graph of job-shop scheduling problem (Adams et al. 1988) 6](#_Toc484289277)

[Figure 3: Operation components of jobs (Zhang et al. 2006) 8](#_Toc484289278)

[Figure 4: The precedence constraint of operation in each job (Baldo et al. 2014) 9](#_Toc484289279)

[Figure 5: The production process in brewery industry (Baldo et al. 2014) 10](#_Toc484289280)

[Figure 6: Production schedule situation in brewery industry (Baldo et al. 2014) 11](#_Toc484289281)

[Figure 7: GA/math programming approach (Toledo et al. 2014) 13](#_Toc484289282)

[Figure 8: Point to point search versus populated based search (Mitsuo et al. 2008) 14](#_Toc484289283)

[Figure 9: A graph G (Mitsuo et al. 2008) 15](#_Toc484289284)

[Figure 10: Adjacency matrix presentation of the graph G (Mitsuo et al. 2008) 16](#_Toc484289285)

[Figure 11: Variable-length encoding (Mitsuo et al. 2008) 17](#_Toc484289286)

[Figure 12: Fixed-length encoding (Mitsuo et al. 2008) 17](#_Toc484289287)

[Figure 13: Priority-based encoding technique (Mitsuo et al. 2008) 18](#_Toc484289288)

[Figure 14: The general structure of hybrid GA (Lin and Gen 2009) 19](#_Toc484289289)

[Figure 15: Applying local search to GA (Lin and Gen 2009) 20](#_Toc484289290)

[Figure 16: Beer production process (www.shutterstock.com) 22](#_Toc484289291)

[Figure 17: Two-stage production process 24](#_Toc484289292)

[Figure 18: Sabeco product set 28](#_Toc484289293)

[Figure 19: Proposed time line 32](#_Toc484289294)

List of Tables

[Table 1: The jobs, operations, precedence constraints 9](#_Toc484289295)

[Table 2: Sabeco product set. 27](#_Toc484289296)

[Table 3: Data input table 1. 28](#_Toc484289297)

[Table 4: Machines capacity table 29](#_Toc484289298)

[Table 5: A snapshot of processing time table 31](#_Toc484289299)

[Table 6: A snapshot of genes presentation 32](#_Toc484289300)

# Introduction

## Motivations

As the result of booming economy and drinking culture in Vietnam, the beer consumption in Vietnam has increased very fast in the past decade. In 2016 brewery consumption in Vietnam increased 10 percent compared to 2015 and 50 percent compared to 2010 (according to Vietnamese Ministry of Public Health). Now, Vietnam is the largest beer consumption in South East Asia, and the third largest beer consumption in Asia just after China and Japan (according to Vietnamese Ministry of Public Health). In 2016, Vietnamese consumers consumed 3.78 billion litter of beer. Beside huge consumption volume, the brewery market in Vietnam is still predicted growing fast. According to Euromonitor, the legal drinking age population in Vietnam will increase from 68.7 million in 2016 to 72.4 million by 2021, and the Vietnamese government is expecting that the growth rate of the economy is around 6.5 percent in the next 5 years. Therefore, Vietnamese brewery market is very huge and promising. Although, local brewery manufacturing company like Sabeco (the largest beer manufacturing company in Vietnam with 40.59 percent of the market share) produce huge amount of brewery products but the company still do not have an effective computerized scheduling function.

The last time I visited a Sabeco brewery manufacturing plant in my hometown, they have no computerized scheduling function. The plan receives order from the header quarter in Ho Chi Minh city and produce 3 shift per day if needed. The fact that, brewery industry is a huge industry in Vietnam and there still is no computerized scheduling function (at least with the local company like Sabeco). These two facts are the motivation for me to develop a scheduling function for the brewery industry. And the chosen method is GA because GA has become popular in recent years and it is considered as an effective algorithm for optimization problem in general as well as scheduling problem specifically.

## Objectives

The motivation for this research thesis comes from the real industrial beer manufacturing case in Ninh Thuan province in Vietnam. During a summer vacation in 2016, I have had a trip to visit a beer manufacturing plant in my hometown (Ninh Thuan provine, Vietnam). At that time, I realized that the plant manager needs a better production planning tool. Therefore, I have been thinking about combining the duty of writing a research thesis with solving the real world industrial case. At the result of this though, this research thesis ultimate goals is to build a scheduling function for brewery industry. A long with this major goal, the other objectives of this research thesis are as flowing:

* Give the reader comprehensive review about scheduling literature.
* Conduct a comprehensive review about GA in recent years.
* Give an interesting introduction about GA to the readers.
* Apply GA as a framework of the scheduling tool combining with local search technique to speed up the searching process.

Rising from those objectives, this research thesis needs to answer the flowing questions:

* What are the major scheduling problems in literature discussion as well as the major scheduling problem in brewery industry.
* What is GA?
* What are the major components of GA.
* Are there how many ideas about applying GA effectively.
* How to design an effective GA to solve the scheduling problem in the context of brewery industry.

## Research Framework

This thesis research concerning about applying GA to schedule brewery production. The thesis is organized into 5 chapters. Chapter 1 will discuss about the motivation and objectives of the research. In this section, there are a few related questions to the research goal which is risen to give the audiences a comprehensive overview of the thesis content.

In chapter 2, some typical scheduling problems in literature will be review. These problems will build the back ground understanding out scheduling problems. In addition, the literature related to brewery scheduling will be discussed. The discussion of brewery scheduling literature includes brewery production processes, the difficulty or the problem needed to be solved by scheduling. Several models and algorithm from literature researchers also be discussed to give an overview of the past contribution for solving the scheduling problem in brewery industry. Finally, GA will be described both in general idea and application for network models. The application of GA in network model problem will be discussed in detail since the brewery scheduling problem is modeled as a network problem. In Chapter 3, the brewery production processes will be described in detail. The production processes understanding is very helpful in building the mathematical models. Since the goal of this thesis research is creating a scheduling function for a real case, the model need to be closed to the real case as possible. In chapter 4, the proposal genetic approach will be described in five components: 1. a genetic representation of potential solution to the problem, 2. a way to create population, 3. an evaluation function rating solutions in term of their fitness, 4. genetic operators that alter the genetic composition of the offspring (mutate, crossover, etc.), 5. parameter values that GA uses (population size, probabilities of applying genetic operators, etc.). In chapter 5, the thesis will mention about the research result and conclusion.

# Literature Review

## Scheduling Problems in Industrial Management

Scheduling is one of the most important functions in Industrial Management. There are many different models have been proposed to solve different scheduling problems in different industries. However, it is noticed that there are three major models in the field of scheduling which considered one plant, they are: the job-shop scheduling model, the flexible job-shop scheduling model, the integrated operation sequence & resource selection model. In Industrial Management, Scheduling function should create a good feasible solution whereas consider all the constraints(Mitsuo, 2008 #20;Cheng, 1995 #32):

* Material availability
* Machine and labor capacity
* Customer service level requirements (due dates)
* Inventory safety stock levels
* Cost
* Distribution requirements
* Sequencing for set-up efficiency
* etc.

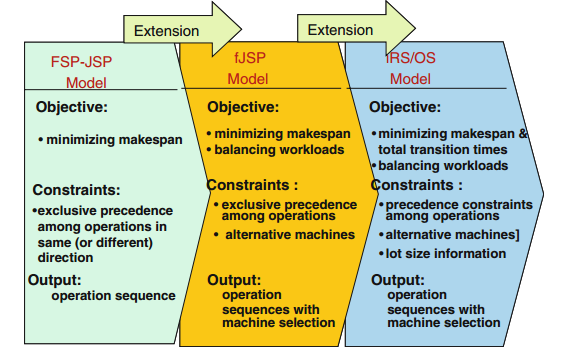


Figure : Typical scheduling problems (Zhang et al. 2006)

### The Job-Shop Scheduling Problem

Job-shop scheduling problem can be describe as flowing (Cheng et al. 1996): there are *m* different machines and *n* different jobs to be schedule. Each job is composed of a set of operations and the operation on machines is prespecified. Each operation is characterized by the required machine and the fixed processing time. The assumption for the job shop scheduling problem includes:

* A job does not visit the same machines twice.
* There are no precedence constraints among operations of different jobs.
* Operation can not be interrupted.
* Each machine can process only one job at a time.
* Neither release times nor due dates are specified.

The objective of the job-scheduling model is to determine the operation sequences on the machines in order to minimize the makespan (the completion time of all jobs).

Since the job-shop scheduling problem is classified as NP-hard, many researchers focus on heuristic approach to solve job-shop scheduling problem. There are many heuristic algorithms have been proposal range from simple dispatching rules to complex hybrid search algorithms. Haupt (1989) have done a comprehensive survey on priority rule-base scheduling. Adams et al. (1988) proposed the heuristic algorithm called the shift bottleneck procedure to solve the job-shop scheduling problem. This heuristic algorithm is considered one of the most effective heuristic algorithms to solve the shop-job scheduling problem. In the paper, the authors present the problem by disjunctive graph *G = (N, A, E)*, with a set of nodes *N*, a set of ordinary (conjunctive) arc *A*, and a set of disjunctive arc *E*. The disjunctive graph is illustrated by the figure below:

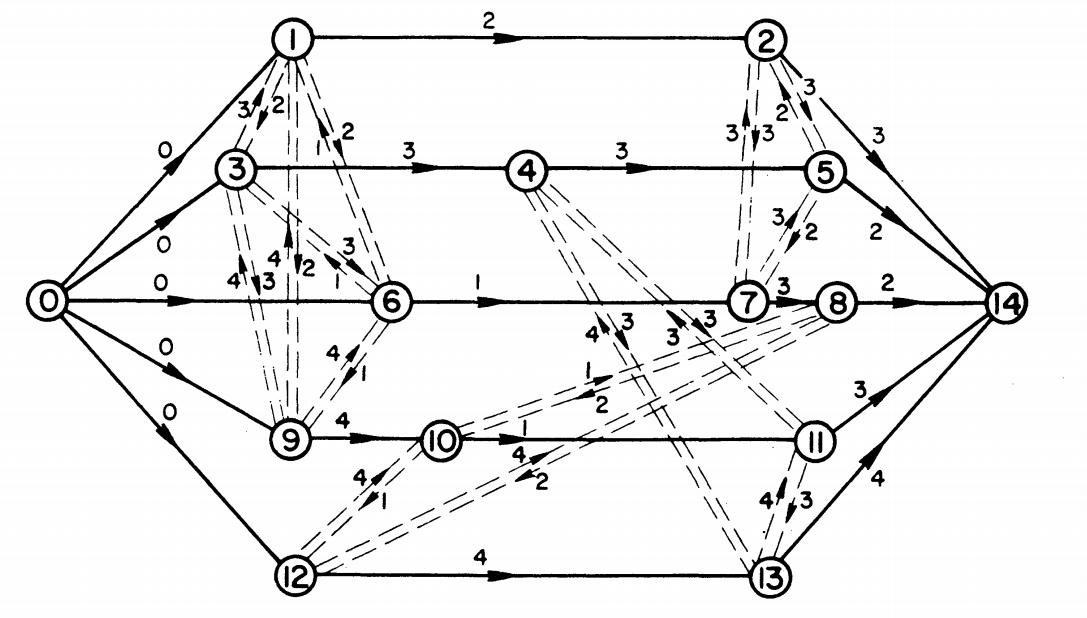


Figure : Graph of job-shop scheduling problem (Adams et al. 1988)

The figure represents the job-shop scheduling problem with 15 operations (5 jobs) on 4 machines. The nodes of *G* present operations, the directed arcs present the precedence operations, the pair of disjunctive arcs present possible sequence on the machine. The number on directed arc is the processing time of the operation. A brief description of the shifting bottleneck algorithm (Adams et al. 1988) is as flowing:

* Let be the set of machines already sequenced ( = ∅ at the start).
* Step 1: Identify a bottleneck machine *m* among the machines *k* ∈ *M\* and sequence it optimally. Set ← ∪ {*m*} and go to 2.
* Step 2: Reoptimize the sequence of each critical machine *k* ∈ in turn, while keeping the other sequences fixed; i.e., set : = - {k} and solve P(k, ). Then if = *M*, stop; otherwise go step 1.

The main contribution of the paper is not the idea of solving the sequencing problem one by one, but the way of using the definition of bottleneck machine to decide which machine to be scheduled. This approach based on the idea of giving the priority to the bottleneck machine. In recent year, the GA has become more popular. Many researchers have tried to use GA to solve scheduling problem, such as Cheng et al. (1995), Xia et al. (2016), Kundakci and Kulak (2016). In recent years, many researchers focus on the dynamic aspect of scheduling and using hybrid GA to solve the dynamic scheduling problem. Kundakci and Kulak (2016) proposed a hybrid GA to solve the dynamic job-shop scheduling problem. Xia et al. (2016) proposed an integrated process planning and scheduling model and solved it by using the hybrid GA with variable neighborhood search.

### The Flexible Job-Shop Scheduling Problem.

The flexible job-shop scheduling problem is the extended problem of the job-shop scheduling problem. That is a machine can process more than one type of operation. Meaning that, for given any operation, there must exist at least one machine capable of performing it (Mitsuo et al. 2008). There are two kind of flexible on the machines (Kacem et al. 2002):

* Total flexibility: all operations can be processed on all machines.
* Partial flexibility: some operations can only be processed on a sub set of machines.

Flexible job-shop scheduling problem can be described as flow: there are *n* jobs to be schedule on m machine. Each job *i* contained ordered operations. A machine can only process one operation as a time, and the machine is busy until the operation is complete. The flexible job-shop problem is to assign operations on machines and to schedule operations assigned on each machine, subject to the constraints that:

* The operation sequence of each job is prescribed.
* Each machine can process only one operation as a time.

Since, flexible job-shop scheduling problem is the generalized problem of job-shop scheduling problem, which is even harder to solve. In recent year, the heuristic approach to solve the problem in reasonable time is popular (Gen et al. 2009, Pezzella et al. 2008).

### The Integrated Operation Sequence and Resource Selection Problem:

The integrated operation sequence and resource selection problem is extended from the traditional scheduling problem (the job-shop scheduling problem and the flexible job-shop scheduling problem), which is closer to the real manufacturing environment. For example, job is combined by a set of operations. Some operations have precedence constraint, other do not (Mitsuo et al. 2008):

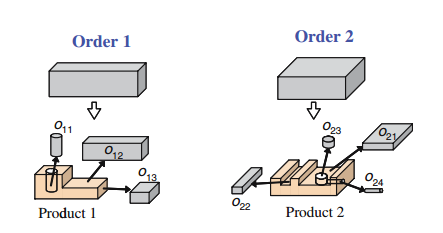


Figure : Operation components of jobs (Zhang et al. 2006)

Table : The jobs, operations, precedence constraints

|  |  |  |
| --- | --- | --- |
| Job | Operation | Precedence constraint |
| 1 | O11, O12, O13 | (O12, O13) |
| 2 | O21, O22, O23, O24 | (O21, O23), (O23, O24), (O24, O22) |

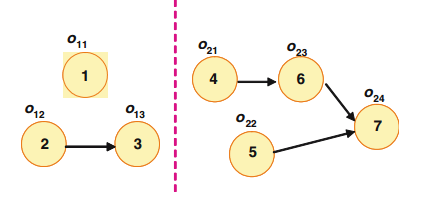


Figure : The precedence constraint of operation in each job (Baldo et al. 2014)

The integrated operation sequence and resource selection problem can be describe as flow (Zhang et al. 2006): given a set of K orders with lot size find an operations sequence for each job, a schedule in which jobs pass between machines and a schedule in which operations on the same jobs are processed such that it satisfies the precedence constraints and it is optimal with respect to minimize makespan and balance the workloads.

Since the integrated operation sequence and resource selection problem is resembling with the real world manufacturing problem, it has been received great attention from researchers. However, the problem is a NP-hard problem and it is even more complex than the job-shop scheduling problem and the flexible job-shop scheduling problem. Therefore, to solve the problem, many heuristic algorithms have been proposed and the GA is one of the main branches of the movement. Zhang et al. (2006) proposed two vector-based coding approach for encoding the operation sequence and resource selection in chromosome. They also used multistage operation-based GA combine with left-shift hillclimber to speed up the computation time. Amin-Naseri and Afshari (2012) proposed the similar approach with (Zhang et al. 2006), which is used GA combined with local search technique to speed up the processing time.

### The Scheduling Problem in Soft Drink and Brewery Industry.

The production process of soft drink and brewery industry is characterized by two major production stages: liquid preparation (stage I), bottling (stage II), and the sequence dependence between the two stage, the raw material have to go through the two processing stage consecutively in order to become a complete product (Ferreira et al. 2009), (Baldo et al. 2014).

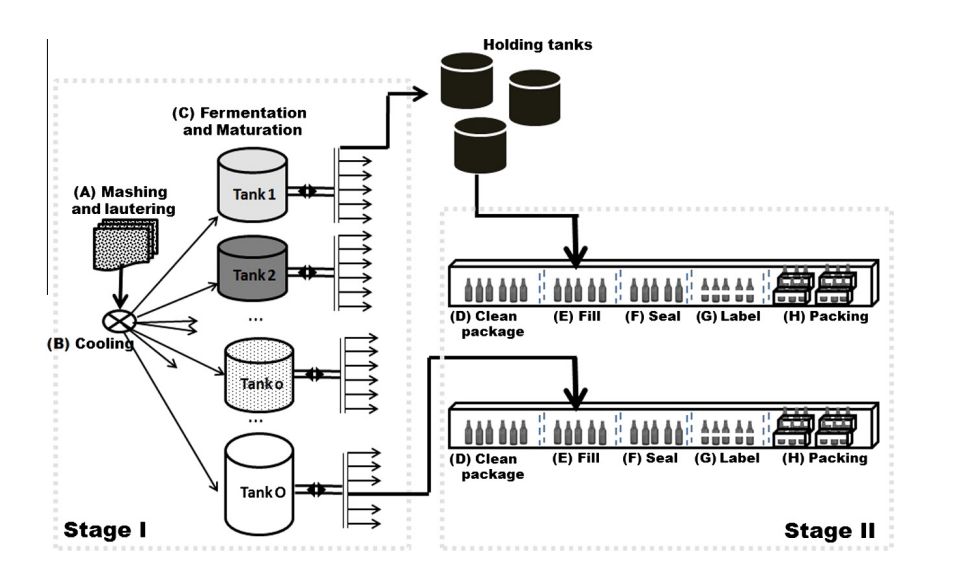


Figure : The production process in brewery industry (Baldo et al. 2014)

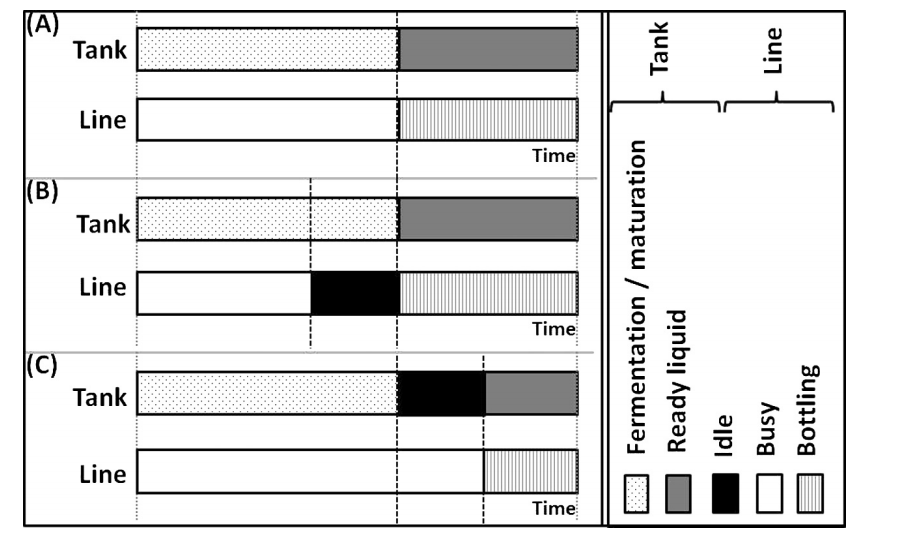


Figure : Production schedule situation in brewery industry (Baldo et al. 2014)

One of the important questions in soft drink and brewery industry is how to use the source (tanks and bottling lines) effectively. The figure 5 depicts situations in production process of a typical brewery production line. Situation (A) is the ideal situation, in which the tank and line are synchronized, the ready liquid in the tank go immediately to the filling line. Situation (B), the filling line has to wait for ready liquid. Situation (C), the liquid in the tank is ready but the filling line is busy for other job, then it has to wait. The problem rising from production is how to synchronize production process between the two stages, reduce the bottle neck and increase utilization of the asset.

The scheduling problem in soft drink and brewery industry usually is formulated as a variance of the general lot sizing and scheduling problem (Fleischmann and Meyr 1997). The model schedule a set of products= *1, 2, …, J* over a finite planning horizon containing a set of macro periods *t = 1, 2, …, T*. Each macro-period *t* is divided into a fixed number of non-overlapping micro-periods with variable length, where denotes the set of micro-periods *s* belonging to the macro-period *t* and all micro-periods are ordered in the sequence *s = 1,..., S*. The two layer of time structure is based on the idea that the external dynamics of the system (demands and holding costs, etc.) should be modeled by a fixed discrete time grid. And, the internal dynamics of the system (the changes of system stage) should be independent on external dynamics, and controlled by decisions. Both internal and external dynamics are linked by the assignment of micro-periods to macro-periods. The length of micro period is also decision variable but expressed in the model by the quantity produced within. The number of micro-periods || within macro-period t is fixed in advance to allow MIP-modeling. It sets an upper bound to the number of possible changeovers within macro period t. A lot consists of a sequence of micro-periods assigned to the same item and may continue over different macro-periods.

The general lot sizing and scheduling problem then was adapted by Ferreira et al. (2009) to formulate the two-stages model for soft drink production. Ferreira et al. (2012) formulate the two stage production process with single-stage formulation. Baldo et al. (2014) modeled the brewery production process with two-stage MIP model and solve it with relax-and-fix, relax-and-optimize heuristic. Toledo et al. (2014) used heuristic GA to solve the lot sizing and scheduling problem in soft drink industry. In their approach, GA deals with sequencing decisions for production lots, after that the remaining linear model gives answer for lot sizing decision. The figure 7 depicts the GA and math programming approach for two-stage production process in soft drink industry.

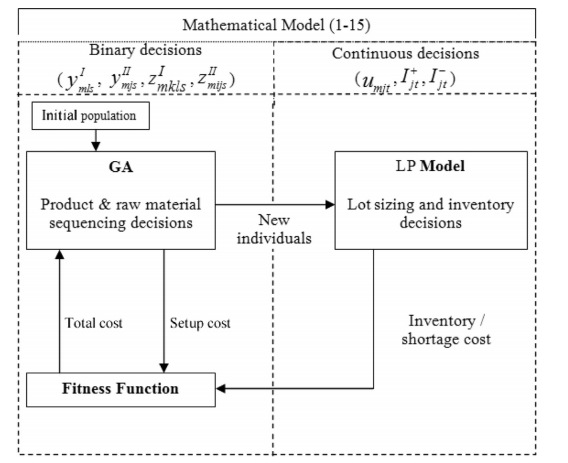


Figure : GA/math programming approach (Toledo et al. 2014)

## GA Approach

### GA in General

Computer scientist has been dreaming about one type of algorithm that can solve different type of optimization problem. The GA has been born from that desire and it try to mimic the natural evolutionary process. Indeed, the GA has been applied to many fields such as Finance, Supply Chain Management, and Information Technology. Orito et al. (2009) used GA solve Index Fund Optimization Problem, which is a very useful tool for hedge trading strategy in Financial Market. Leu and Namatame (2008) use GA to solve optimize network design problem.

As mention above, GA can be used as optimization solver for a wide range of problem in many fields. However the principle structure of the algorithm is the same across applications as mention by Mitsuo et al. (2008). In general, a GA has five basic components:

1. A genetic representation of potential solutions to the problem.
2. A way to create a population (an initial set of potential solutions).
3. An evaluation function rating solutions in term of their fitness.
4. Genetic operators that alter the genetic composition of offspring (crossover, mutation, selection, etc.).
5. Parameter values that GA uses (population size, probabilities of applying genetic operators, etc.).

Mitsuo et al. (2008) pointed out the important advantage of GA compare to the conventional optimization search method, that is GA is populated base search approach.

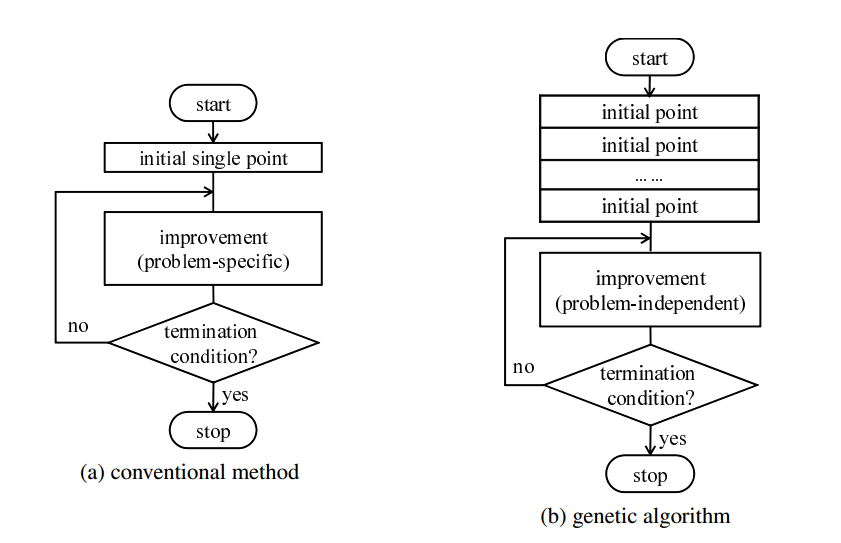


Figure : Point to point search versus populated based search (Mitsuo et al. 2008)

The conventional approach is point to point search, can be trap in the local optimum, whereas GA is population to population search method with a degree of randomness. Therefore, the probability of being trap in local optimum of GA is lower than the conventional optimization methods. GA can get good solution in a short period of time since it has both good combination of exploration (search randomly) and exploitation (search in certain direction). At first, the random initialization of the population helps GA explore the search space. As the fitness improving through generations, the crossover, mutation functions help solution move to a certain direction to get the better solution quicker.

### GA and Network Modeling

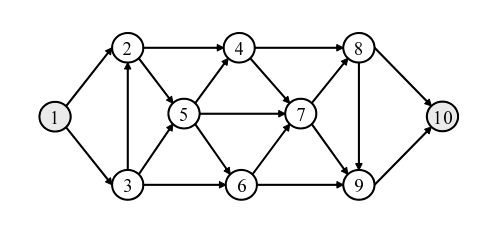


Figure : A graph G (Mitsuo et al. 2008)

The directed graph (or graph) G can be presented in three different ways: edge lists, adjacency lists, adjacency matrix. In the first way, the graph can be characterized by the number of nodes and a list of arcs. In the figure 9, the graph G can be presented by 10 nodes and a list of arcs *A* = {(1,2), (1,3), (2, 4), (2, 5), (3, 2), (3, 5), (3, 6), (4, 7), (4, 8), (5, 4), (5, 6), (5, 7), (6, 7), (6, 9), (7, 8), (7, 9), (8, 9), (8, 10), (9, 10)}. In the second way, a graph can be described by the number of nodes and n list , …, , …, , where contains all nodes j for which G contains an arc (i, j). For example, the graph G can be presented by 10 nodes and the adjacency lists: = {2,3}, = {4,5}, = {2,5,6}, = {7,8}, = {4,6,7}, = {7,9}, = {8,9}, = {9,10}, = {10} and = ∅. In the third way, a graph of n nodes can be presented by a (n\*n) adjacency matrix *A = (),* where = 1 if and only if (*i, j)* is an arc of *G*, and = 0 otherwise.

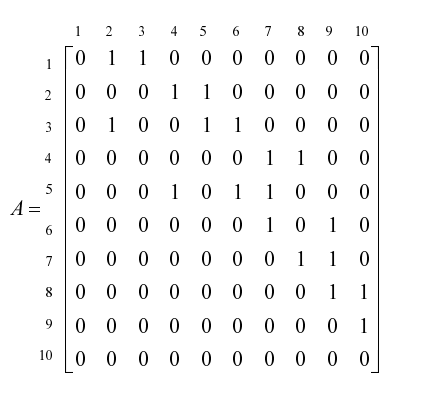


Figure : Adjacency matrix presentation of the graph G (Mitsuo et al. 2008)

Genetic presentation is one of the five major components of a GA. Indeed, a good GA has to be both fast and accuracy. The accuracy ability is to present a feasible solution in the chromosome as well as make sure the feasibility of the chromosome after going through all the operations (mutation, crossover, etc.). Feasible solutions can be presented in the chromosome in several ways. Chang Wook and Ramakrishna (2002) use variable-length encoding technique to solve the shortest path routing problem. Inagaki et al. (1999) use fixed-length encoding technique to solve the multiple routing problem.

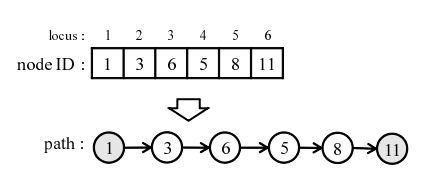


Figure : Variable-length encoding (Mitsuo et al. 2008)

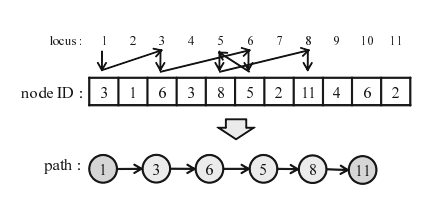


Figure : Fixed-length encoding (Mitsuo et al. 2008)

In the variable-length encoding technique, locus (the position of gene in chromosome) sequence presents the sequence of nodes that the path goes through, and the value of gene is the id of the node (figure 11). In fixed-length encoding technique, to present an arc from node *i* to node *j*, put node *j* to the *i-th* locus. This process is reiterated from source node 1 through sink node n. If the route does not path through a node *x*, select one node randomly from the set of nodes which are not connect to node *x* and put it in the *x-th* locus. The disadvantage of variable-length and fixed-length encoding technique (Mitsuo et al. 2008) is that the infeasible chromosome can be created through genetic operation.

In recent year, priority-based encoding technique is become popular. Mitsuo et al. (2008) listed the advantage of priority-based encoding technique as flowing: (1) any permutation of the encoding corresponds to a path (feasibility), (2) most existing genetic operators can be easily applied to the encoding, (3) any path has a corresponding encoding (legality), (4) any point in solution space is accessible for genetic search.

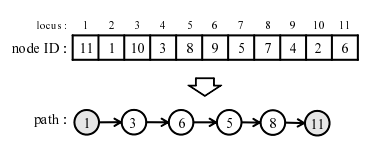


Figure : Priority-based encoding technique (Mitsuo et al. 2008)

In the priority-based encoding technique, the locus (position of gene in chromosome) represents the id of node. The alleles (the value of gene) represent the priority level of that node. To illustrate the priority-based encoding technique, let take the example from figure 13. First, we try to find the next node from the source 1. From the adjacency list S1, we know node 3 and node 2 are eligible. However node 3 has higher priority than node 2, therefore node 3 is in the path. The iteration continues with adjacency list S3 until we get to the path. The priority-based encoding technique was used by (Zhang et al. 2006), (Gen et al. 2009).

In recent year, GA has evolved in to hybrid heuristic search method, which is the combination between genetic search and local search. The important question is how to design a good combination between the two search method to get a reliable and fast hybrid GA.

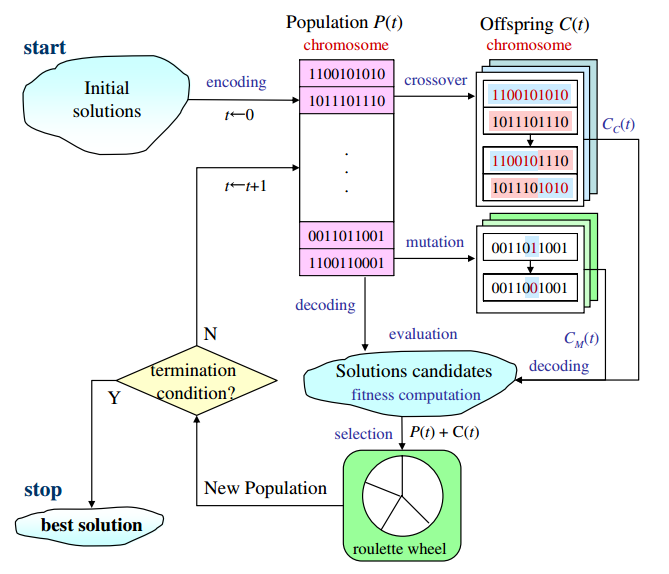


Figure : The general structure of hybrid GA (Lin and Gen 2009)

The general structure of hybrid GA is depicted by figure 14. The traditional local optimization search (hill-climbing, etc.) is applied to help offspring get better genes and then be put back to the population. By adding a local search algorithm to each iteration of GA, we have better algorithm. Since, the algorithm has two good characteristics from both GA and local search algorithm. The randomness from GA will help the hybrid GA avoid being trapped in local optimization. The directed search from local search algorithm will help the hybrid GA move toward the local optimum faster than the GA alone.

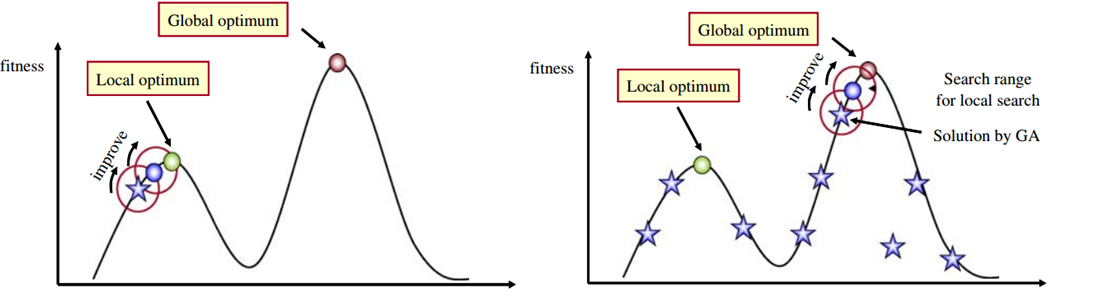


Figure : Applying local search to GA (Lin and Gen 2009)

Lin and Gen (2009) proposed a very interesting approach called auto-tuning strategy. GA was invented based on the evolutionary idea in which the algorithm adapts itself to the type of problems. It is desired to have a GA that can adapt to different type of problems and auto-tuning strategy is one of the answers. The auto-tuning strategy (Lin and Gen 2009) help the GA balancing between exploration (random search) and exploitation (directed search) by control the probability of genetic operators. Generally speaking, if the GA is biased toward exploitation then the probability of random search is low and the probability of local search is high. On the other hand, if the GA is biased toward exploitation then the probability of random search is high and the probability of local search is low. We can control the balance between exploitation and exploration by controlling the probability of GA operators like cross-over (neural), immigration (exploration bias), mutation (exploitation bias).

# Brewery Production and Problem Description

## Beer Production Process

The main ingredients for making beers include: barley, hops (responsible for the bitterness of beer), water and yeast (responsible for transforming sugar to alcohol and C02). The manufacturing is varying between different manufacturers, however the core production process is the same for all brewery industry. There are four major steps for beers making. The first step is malting. The grains (mainly barley) are harvested and processed through a process of heating, drying out and cracking. The main goal of malting is to isolate the enzymes needed for brewing. The second step is mashing in which grains are steeped in hot (but not boiling) water for an hour. Then, water is drained out from the mash. This activates enzymes in the grains that cause it to break down and release sugars. The result of mashing is sticky water which contains sugar extracted from the mash, called *wort*. The thirst step is boiling, in which wort is boiled for about an hour while ops and other spices are added several times. The fourth step is fermentation, after the boiling step, the wort is cooled, strained and filter. It is then put in a fermenting vessel and yeast is added to it. At this point the brewing is complete and the fermentation begins. The beer is stored for a couple of weeks at room temperature (in the case of ales) or many weeks at cold temperatures (in the case of lagers) while the yeast transform the sugar in the wort. The final step is bottling. The ready liquid will pour to the bottle in a filling line.



Figure : Beer production process (www.shutterstock.com)

## Problem Description and Modeling

In brewery production processes, there are two key types of machines. They are tanks and filling lines, these types of machines are also the bottleneck of production processes. Tanks are used to hold the liquid and the holding time can take from one or two weeks to one or two months depending on the type of brewery. Filling lines are used to fill the bottles with liquid. Filling lines are the most important machines in brewery manufacturing plant since they are the most expensive machines of all type of machine used in the plant. Due to the long fermentation in tanks and the expensive filling lines, the task of synchronizing operation between the two types of machines is very important in brewery production.

The brewery production process is modeling with two types of machine, they are tanks and filling lines. The mathematical model used in this thesis is proposed by Ferreira et al. (2012), and the main contribution of this thesis is proposing a GA to solve the production scheduling problem in brewery industry. Note that, the single stage-formulation model (Ferreira et al. 2012) is formulated for soft drink industry. However, the fermentation process and filling process of brewery production is exactly similar with the soft drink production process. Therefore, it is understandable to adapt the soft drink model (Ferreira et al. 2012) in the case of brewery production case. The single stage formulation for two-stage lot sizing and scheduling problem has some assumption:

* Two different liquid types cannot be prepared simultaneously in the same tank
* A tank has to be cleaned before new lot of liquid even if the new lot of liquid is the similar type with the last one.
* There are sequences dependence of liquid types in term of preparation time and cost.
* There is minimum requirement for lot of liquids which make sure the homogenous of the liquid in tanks.
* The tank is available to prepare other lot of liquid once the corresponding line completing the bottling process.
* The filling line can only start to work only when the liquid in the corresponding tank is ready.
* A line can only receive liquid from only one tank as a time, no matter there are how many ready liquid in other tanks.
* A tank can supply liquid to multiple filling lines simultaneously as long as they filling the same type of liquid.

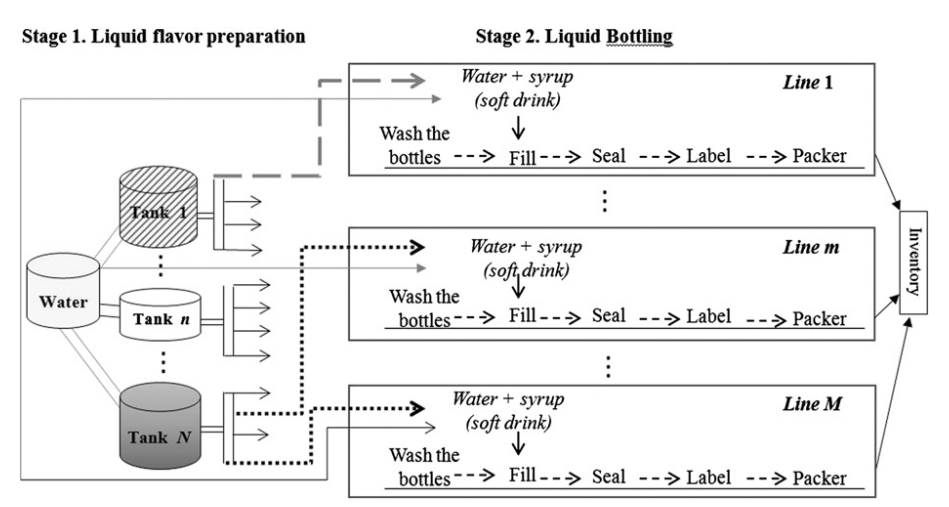


Figure : Two-stage production process

To dealing with the scheduling problem in brewery industry, this thesis proposes an approach which is the combination of math programming and GA for flexible job shop scheduling problem.

First, to deal with job assignment for tanks a math model is proposed such that it minimizes the fermentation time. The math model is described as flowing:

**Set:**

p ∈ P with P is a set of products.

t ∈ T with T is a set of tank types.

**Decision variable and parameter:**

: The number of time use tank type t for producing product p.

: Fermentation time of product p.

: Demand for product p.

: Tank type t capacity.

**Objective function:**

Min (1)

**Constraint:**

≥ ∀ p ∈ P (2)

= … = (3)

≥ 0, ∈ Z (4)

The objective function (1) is to minimize the fermentation time of a given quantity demand. Constraint (2) is to make sure that the demand is always fulfill. Constraint (3) is to make sure that tank types are use equally. Equation (4) is the valid condition of decision variables. Note this in this model, tank types are considered but not specific tanks. Because, the second algorithm will help to choose specific machines in a flexible job shop scheduling problem.

The second problem is to synchronizing between tanks and filling machines. To solve this problem a GA is proposed to dealing with flexible job shop scheduling problem. The flexible job shop scheduling problem is formulated as flowing:

Set:

i ∈ I with I is a set of jobs.

j ∈ J with J is a set of machines.

k ∈ K with K is a set of operations.

Parameter:

n: total number of jobs

m: total number of machines.

: total number of operations in job i.

: the i-th job.

: the k-th operation of job i,

: the j-th machine.

: processing time of operation on machine j.

U: a set of machines with the size m.

: a set of available machines for the operation .

: workloads (total processing time) of machine j.

Decision variables:

= 1 if machine j is selected for operation , 0 otherwise.

= completion time of the operation .

Objective function:

Min = (5)

Min = {} (6)

Min = (7)

Constraints:

- - ≥ 0 k = 2, …, K; ∀ i, j (8)

= 1 ∀ k, i (9)

∈ 0, 1 ∀ j, k, I (10)

≥ 0 (11 )

# Solution Approach and Sabeco Case Study

## Tanks types assignment with math programming

To solve the scheduling problem in brewery industry, this thesis proposes approach of two stage computation. First step, given the demand of different products the program will be assign jobs to tank types. The tank types is consider instead of specific tanks because flexible machine chosen for the later GA algorithm for flexible job shop scheduling problem. Second step is trying to synchronize the production process between fermentation state and filling bottle stage. This is where GA is deployed to help the production process more smoothing.

Sabeco is the largest brewery producer in Vietnam. In 2013 Sabeco produced 1,330 billion litter of beer. The company has around 24 plants in Vietnam with the capacity of 1.8 billion litters (According to Sabeco website: <http://sabeco.com.vn>). For a large brewery producer like Sabeco, large tanks and high capacity filling line are used in production process. Tanks with the capacity of 1000 BBL (barrel, and 1 BBL = 1.17 hectoliters (hl), and 1 hectoliter = 100 liters (l), 1 litter = 1000 milliliters (ml), ml is the volume measure for beer bottle or can in Vietnam) and 500 BBL, bottle filling line and can filling line with the capacity around 30.000 bottle or can/hour (According to Sabeco website: <http://sabeco.com.vn>).

Table : Sabeco product set.

|  |  |  |  |
| --- | --- | --- | --- |
| Product Code | Product Name | Beer Type | Volume/bottle or can |
| B1 | 333 premium (bottle) | premium lager | 330ml |
| B2 | 333 (can) | lager | 330ml |
| B3 | saigon lager (bottle) | lager | 450ml |
| B4 | saigon lager (can) | lager | 330ml |
| B5 | saigon export (bottle) | lager | 355ml |
| B6 | saigon special (bottle) | lager | 330ml |
| B7 | saigon special (can) | lager | 330ml |



Figure : Sabeco product set

Table : Data input table 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Product Code | Product Name | Volume/bottle | Demand | Beer Type | Fermentation Time |
| B1 | 333 premium (bottle) | 330ml | production demand | premium lager | 28 |
| B2 | 333 (can) | 330ml | production demand | lager | 21 |
| B3 | saigon lager (bottle) | 450ml | production demand | lager | 21 |
| B4 | saigon lager (can) | 330ml | production demand | lager | 21 |
| B5 | saigon export (bottle) | 355ml | production demand | lager | 28 |
| B6 | saigon special (bottle) | 330ml | production demand | lager | 28 |
| B7 | saigon special (can) | 330ml | production demand | lager | 28 |

Table : Machines capacity table

|  |  |  |
| --- | --- | --- |
| Machine | Capacity | unit |
| TA1 | 500 | BBL |
| TA2 | 500 | BBL |
| TA3 | 500 | BBL |
| TA4 | 500 | BBL |
| TA5 | 500 | BBL |
| TB1 | 1000 | BBL |
| TB2 | 1000 | BBL |
| TB3 | 1000 | BBL |
| TB4 | 1000 | BBL |
| TB5 | 1000 | BBL |
| TB6 | 1000 | BBL |
| TB7 | 1000 | BBL |
| TB8 | 1000 | BBL |
| FC1 | 30000 | can/hour |
| FC2 | 30000 | can/hour |
| FB1 | 35000 | bottle/hour |
| FB2 | 35000 | bottle/hour |

The program in this thesis is design so that the users can easily get their schedule result given the demand quantity that they want to produce. Therefore the users only need to change the demand for each product (demand is measure in ml) and put the start button then the program will automatically compute good schedule result.

Products, machines are coding so that it is easier to make reference and compute the result in computer system. The coding for products and machines are explained as flowing:

B: product (beer) range from 1 to 7.

TA: tank type A which has the capacity of 500 BBL.

TB: tank type B which has the capacity of 1000 BBL.

FC: can filling line which has the capacity of 30.000 cans/hour.

FB: bottle filling line which has the capacity of 35.000 bottles/hour

Note: all the information about the product set and machine type is taken from Sabeco website <http://sabeco.com.vn>, exact machine is based on the assumption of author. The program is design with flexibility in mind. The user can change, contract or expand and the program should work find.

## Genetic Algorithm Approach for Synchronize The Production Process

Basically, Brewery production process includes two major operations. First, liquid is fermented in large tanks. After go through the fermentation time, the mature liquid is filling in bottle or can in bottle filling line or can filling line. The first stage of computation will automatically assign job to tank types in order to minimize the fermentation time. Base on the job assignment result, job and operation definition need to be defined in order to prepare the input for the synchronize production stage. The preparation step is conduct as following:

If tank type A (TA) is assign to use n time to produce SaiGon Larger (can) or product (B4). Then the program will define the job as TAB41, …, TAB4n. Since each job has only two operation (fermentation and filling). Each job TAB4n will be divided into two smaller operation TAB4n1 for fermentation and TAB4n2 for filling the liquid to bottle and can. Since the coding of job and operation have the information about tank types and products, we can use these information to get the fermentation time and compute the filling time of each operation.

Pseudocode for data preparation:

Input: the tank types assignment for production.

Output: processing time table. The processing time table is present as a dictionary with key is operations and machines and value is the time to finish the operation given the machine.

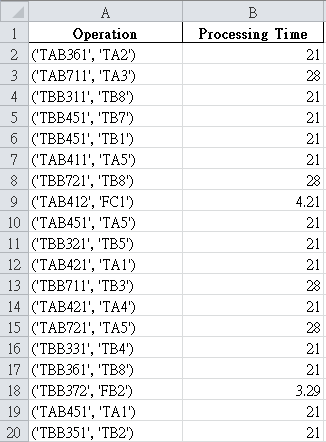
Step 1: given the tank types assignment create job list coding.

Step 2: given the job list coding create the operation list coding.

Step 3: given the operation list coding get the machine set that can process each operation (present like a dictionary called operation machine dictionary).

Step 4: given the operation list coding, operation machine dictionary, machine capacity get fermentation time or compute filling time for the operation.

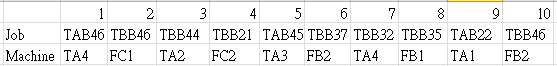
Table : A snapshot of processing time table



### Genetic Presentation

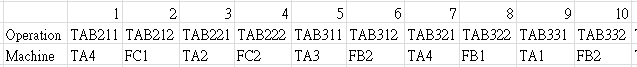
Each individual (or solution) is present by two vectors. Vector 1 presents operation sequences decision. Vector 2 presents machine selection decision.

Table : A snapshot of genes presentation



The idea for this genes presentation was proposed by Gao et al. (2007). Each operations sequence will be present by a list of job. For example the first job coding ‘JAB46’ will present the operation coding ‘JAB461’, the first job coding ‘TBB46’ will present the operation coding ‘TBB461’, the second job coding ‘JAB46’ will present the operation coding ‘JAB462’ and so on. The sequence decision is read from left to right. Machines selection decision is present by a list of machines and each machine is mapping with a given list of operations. The table below will present a snapshot of machines selection decision.

Table : Machines selection presentation

 The coding of the operation give us the information about the operation itself, therefore we can use it reference to a set of machine that can process the operation. For example ‘TAB211’ is fermentation operation of product B2 on tank ty A, and ‘TAB212’ is the filling operation of product B2.

### Decode the genes to actual plan

From the genes, the actual plaining is obtained through a decoding mechanism. The decoding mechanism is described as a pseudocode as flowing:

Input: genes presentation, operations list, processing time table.

Output: the actual plan like a dictionary with key is (‘operation’, ‘machine’) and value is starting time and completion time.

Step 1: from the genes presentation get a set of machines and a set of jobs.

Step 2: from a set of machines and a set of job create a dictionary of machines time and jobs time to keep track of machine time and jobs time. Machines time is the latest time

Step 3: For each operation on the sequence decision:

Step 3.1 get the operation time of that operation on the chosen machine from the processing time table.

Step 3.2: update the machines time of the chosen machine by operation time, update the jobs time of the current operation by the operation time.

Step 3.3: get the completion time of the current operation by choosing the maximum of machines time of the chosen machine and the jobs time of the current job.

Step 3.4: compute the starting time of the current operation by the completion time of the current operation – processing time of the current operation.

Step 3.5: update both machines time of the chosen machine and the jobs time of the current job by the completion time of the current operation.

Step 3.6: get the starting time and completion time of the plan (‘current operation’, ‘chosen machine’) by the current starting time and completion time respectively.

The plan is built on the assumption that the next operation will enter the machine immediately when the chosen machine is free.

### Compute fitness of individual genes

A schedule plan can be decoded from genes of a individual. From the plan of a specific genes encoding, it is easy to get the completion time of all the jobs, the maximum workload of a machine, and the total workload of all machines. The fitness of individual genes is design with 3 attributes:

* The last time to finish all the jobs.
* The maximum workload of a machine.
* The total workload of all machines.

The fitness of an individual with the above three attributes is design with hierarchical structure to compare the fitness between individuals. The first priority is the last time to finish all the jobs. The second priority is the maximum workload of a machine. The third priority is the total workload of all machines. The smaller value of each fitness attribute the better the individual is.

The hierarchical attributes structure is design not only to help the GA find the shorted time to complete all the jobs but also help it to archive balance workload between machines. In reality, it is desirable to finish all the jobs as soon as possible, however the balance workload between machines should be considered.

To summary, each individual is design with genes and fitness. Genes carry the information about sequence decision and machine selection decision. From genes of an individual a plan can be derived. Fitness of an individual is design with the hierarchical structure. The structure is helpful to recognize better genes which present shorter time to finish all the jobs and balance workload between machines.

### Genetic operations

In GA literatures, there are two basic genetic operations. They are mutation and crossover. These genetic operations are inspired by the evolutionary process in the nature world. Mutation is the process of changing the material of genes in an individual. Crossover is the process of exchanging the genes material between two individuals to create new individual. Through the evolution process and the pressure of natural selection, the better adapted individuals have more change to survive.

The mutation mechanism affects both sequence decision genes and machines selection genes of an individual. Mutation mechanism on sequence decision genes is processed by randomly selection two genes on the sequence decision genes then swap them. Since each gene is a job coding, and the decoding mechanism will decode to actual operation plan by reading through the ordered sequence of job on the sequence decision genes. Therefore, this mutation process will change the genes material but will not create any unvalid genes. On the other hand, mutation on machines selection genes is processed by randomly select an operation in the operations set, and choose the machine that can process that operation. By doing so, the mutation process on machine selection genes always create valid genes.

Contract with mutation, crossover is the process of exchange genes material between two individuals. Crossover operation on sequence decision genes is described by pseudocode flowing:

Step 1: get the sequence decision genes on the two parent individuals.

Step 2: cut the two sequence decision genes at the middle point.

Step 3: combine the head of the first genes with the tail of the second genes, and the head of the second genes with the tail of the first genes to create two new individuals.

Step 4: find missing or exceeding genes on the two new genes and fix them (since the operations set is known it is easy to find the missing and exceeding operation and then fix them).

Step 5: return the two new valid genes.

Crossover operation on machines selection genes is simpler than on sequence decision genes. Since the machine selection genes is reference to a fixed set of operations, therefore by cutting the two parent genes at the middle point and combine them in two new genes, the process will always create valid genes.

### Populated Genetic Algorithm

Each individual is design with genes, fitness value. By using hierarchical fitness structure, it is easy to compare the fitness value between individuals. Furthermore, individuals are put into an evolutionary environment. The evolutionary environment is design with selection mechanism (we randomly choose individuals to apply mutation or crossover and keep the better individuals only), the population size, the probability of mutation and crossover. The evolutionary process in populated GA is described as flowing pseudocode:

Input: population size, generating population mechanism, mutate and crossover mechanism, the probability of mutation and crossover, number of generation.

Output: the population of individuals for each generation.

Step 1: randomly create a population with given population size by using generating population mechanism.

Step 2: randomly select individuals to apply mutation and crossover mechanism with the given probability.

Step 3: compare between the new individuals and their parents, choose the better individuals.

Step 4: put the better individuals back the population and algorithm go on for the given number of generation.

### 

## Proposed Time Line

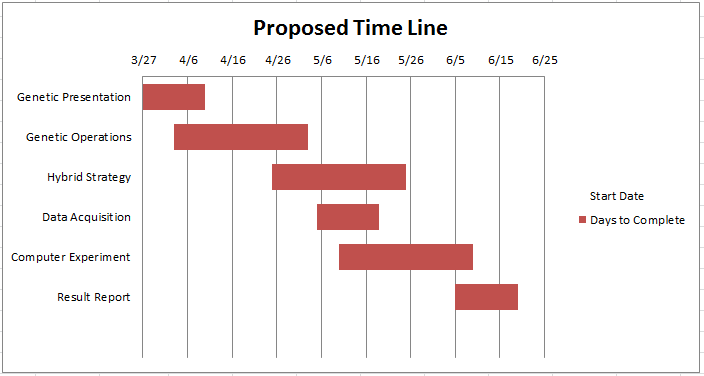


Figure : Proposed time line

Figure 18 describes the time line to complete this thesis. Starting by development of genetic presentation, it will take about 2 weeks. Then the other hybrid GA components will be developed in more than one and a half month. Data Acquisition will start at May fifth and take around 2 weeks. After having the GA and the data, computer experiment will be conducted in about 1 month. The result from this step will help to improve the hybrid GA. The final step is

References

Adams Joseph, Balas Egon, Zawack Daniel. (1988). The Shifting Bottleneck Procedure for Job Shop Scheduling. Management Science. Vol 34(3), pp. 391-401.

Amin-Naseri M. R., Afshari Ahmad J. (2012). A hybrid genetic algorithm for integrated process planning and scheduling problem with precedence constraints. The International Journal of Advanced Manufacturing Technology. Vol 59(1), pp. 273-287.

Baldo Tamara A., Santos Maristela O., Almada-Lobo Bernardo, Morabito Reinaldo. (2014). An optimization approach for the lot sizing and scheduling problem in the brewery industry. Computers & Industrial Engineering. Vol 72, pp. 58-71.

Chang Wook Ahn, Ramakrishna R. S. (2002). A genetic algorithm for shortest path routing problem and the sizing of populations. IEEE Transactions on Evolutionary Computation. Vol 6(6), pp. 566-579.

Cheng Runwei, Gen Mitsuo, Sasaki Masato. (1995). Film-copy deliverer problem using genetic algorithms. Computers & Industrial Engineering. Vol 29(1), pp. 549-553.

Cheng Runwei, Gen Mitsuo, Tsujimura Yasuhiro. (1996). A tutorial survey of job-shop scheduling problems using genetic algorithms—I. representation. Computers & Industrial Engineering. Vol 30(4), pp. 983-997.

Ferreira D., Clark A. R., Almada-Lobo B., Morabito R. (2012). Single-stage formulations for synchronised two-stage lot sizing and scheduling in soft drink production. International Journal of Production Economics. Vol 136(2), pp. 255-265.

Ferreira D., Morabito R., Rangel S. (2009). Solution approaches for the soft drink integrated production lot sizing and scheduling problem. European Journal of Operational Research. Vol 196(2), pp. 697-706.

Fleischmann Bernhard, Meyr Herbert. (1997). The general lotsizing and scheduling problem. Operations-Research-Spektrum. Vol 19(1), pp. 11-21.

Gao Jie, Gen Mitsuo, Sun Linyan, Zhao Xiaohui. (2007). A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. Computers & Industrial Engineering. Vol 53(1), pp. 149-162.

Gen Mitsuo, Gao Jie, Lin Lin. Multistage-Based Genetic Algorithm for Flexible Job-Shop Scheduling Problem. In: Gen M, Green D, Katai O, McKay B, Namatame A, Sarker RA, et al., (Eds.) Intelligent and Evolutionary Systems. Berlin, Heidelberg: Springer Berlin Heidelberg; (2009), pp. 183-196.

Haupt R. (1989). A survey of priority rule-based scheduling. Operations-Research-Spektrum. Vol 11(1), pp. 3-16.

Inagaki J., Haseyama M., Kitajima H. (1999). A genetic algorithm for determining multiple routes and its applications. Circuits and Systems, 1999 ISCAS '99 Proceedings of the 1999 IEEE International Symposium on. Vol 6, pp. 137-140.

Kacem I., Hammadi S., Borne P. (2002). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews. Vol 32(1), pp. 1-13.

Kundakci Nilsen, Kulak Osman. (2016). Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem. Computers & Industrial Engineering. Vol 96, pp. 31-51.

Leu George, Namatame Akira. Evolving Failure Resilience in Scale-free Networks. Intelligent and Evolutionary Systems (2008), pp. 49-59.

Lin Lin, Gen Mitsuo. (2009). Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation. Soft Computing. Vol 13(2), pp. 157-168.

Mitsuo Gen, Runwei Cheng, Lin Lin. Network Models and Optimization: Multiobjective Genetic Algorithm Approach. 1st ed. London: Springer-Verlag London; (2008).

Orito Y., Jun Takano, Takeda M., Yamamoto H. Index Fund Optimization Using Genetic Algorithm and Scatter Diagram Based on Coefficients of Determination. Intelligent and Evolutionary Systems (2009), pp. 1-11.

Pezzella F., Morganti G., Ciaschetti G. (2008). A genetic algorithm for the Flexible Job-shop Scheduling Problem. Computers & Operations Research. Vol 35(10), pp. 3202-3212.

Toledo C. F. M., de Oliveira L., Pereira R. D., Franca P. M., Morabito R. (2014). A genetic algorithm/mathematical programming approach to solve a two-level soft drink production problem. Computers & Operations Research. Vol 48, pp. 40-52.

Xia Hao, Li Xinyu, Gao Liang. (2016). A hybrid genetic algorithm with variable neighborhood search for dynamic integrated process planning and scheduling. Computers & Industrial Engineering. Vol 102, pp. 99-112.

Zhang Haipeng, Gen Mitsuo, Seo Yoonho. (2006). An effective coding approach for multiobjective integrated resource selection and operation sequences problem. Journal of Intelligent Manufacturing. Vol 17(4), pp. 385-397.