

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Gestionale
Classe L-8
Ingegneria dell'Informazione

Tesi di Laurea

Statistical due date for customer satisfaction



Relatore

prof. Fulvio Corno

firma del relatore

.....

Candidato

Fabio Mazzocco

firma del candidato

.....

AA. 2018/2019

Statistical due date for customers satisfaction

Fabio Mazzocco - Student 234844

August-September 2019

1 Project proposal

1.1 Proposal title

Statistical due date for customers satisfaction

1.2 Description of the proposed problem

The due date quoting of an order is one the most difficult problems to deal with because a lot of variables contribute to the actual time that starts when the order is received and ends when the whole order is issued. A good approach is to unify all of them by studying the data from the reception of the order to its issue, using a statistical distribution (normal). If the due date is quoted by trusting the average/mean time, there is only about the 50% of probability that the order will be issued by that date, so a more conservative approach should be used in order to avoid the customer dissatisfaction or even penalties. For these reasons, a statistical approach (besides the optimization of the single variables) is needed.

1.3 Management engineering relevance of the problem

The order management is very important to minimize the delay of the deliveries and to have a decent service level in order to satisfy the customer expectations and increase the brand image. Being able to evaluate due dates means to have the ability to know if the production of important/urgent parts should be escalated or outsourcing is needed. During periods with orders pick, it is crucial to know which parts should be produced first in order to provide the customer the highest quantity of parts in the shortest time and to choose the best mix from this point of view. Carrying out simulations

of the acceptance of an order during a certain period let the company decide whether to accept the order and if it is necessary to fix a larger due date.

1.4 Data-set description for the analysis

The data-set has been provided by the company I did my internship with and it has been anonymized. It is made of one table with the following columns: customer, part-number, quantity, description, good-issue-date, and order-date. Each value is the line of an order and represents the need for a specific part, whereas an order can be made of many lines. The "order-date" does not indicate when the order was received but it is the date in which its production started. It is important to notice that the quantity is not relevant for the production time (once the production starts, it only depends on other factors and the whole quantity is produced all together). When "order-date" and "good-issue-date" are the same value, it means that the requested part was already in stock.

1.5 Preliminary description of the implemented algorithms

Due date quoting (from "Programmazione e gestione della produzione" classes) is one of the implemented algorithm based on a statistical distribution, the group mean and variance are calculated and a quantile (with the chosen probability) is the number of days to produce and issue the parts mix. Recursion is implemented too in order to find the best quantity/days rate having a certain probability of success and delivering a given amount of parts. A discrete-event simulation is carried out to check the behavior of an order during the chosen period using the historical data of a selected year, in this case, the number of parts that can be processed simultaneously is chosen by the user, the production time is a random number of days (based on the statistical distribution of the part). These last two algorithms are supported by the first one.

1.6 Preliminary description of the software functions

The user can choose the parts and the relative quantities demanded: this is the order. Through the due date quoting it is possible to choose a probability and the suggested due date is shown based on the parts statistical distributions. It is also possible to set a due date and check the relative probability using the inverse due date quoting algorithm. These two functionalities are available both for the parallel production of the parts and the serial one.

In case of production picks, it is possible to calculate the best parts mix to be produced/issued at first in order to maximize the quantity/days rate setting the probability and the minimum quantity of parts to be delivered (this can be useful if lines can be issued before the full completion of the order). In the end, it is possible to simulate the behavior of the order in a certain period regulating the number of parts that can be produced simultaneously (due date quoting is applied with a random probability to the single parts in order to determine the random number of days of each one).

2 Problem

2.1 Business context

The business context is a mechanical company where different components, parts, and products (items from now on) are fabricated by both using machines and manpower. One of the most important problems of this sector is the due date quotation because the production time is not deterministic and has a certain variability, this means that the mean time is not a good choice since there is only a certain percentage of probability that production will be completed by that date, e.g. if the statistical distribution is a normal distribution, the cumulative probability is only 50%. By using a statistical approach, it is possible to comprehend many different factors that create variability into two values: mean and variance. In this case, the production time variability does not depend on the quantity, but comes from other specific influences: material quality, environmental factors, workers presence and experience, equipment failures and quality problems that bring to reworking, suppliers reliability, downtime due to rest of the items after some operations. It is important to notice that the quantity is one of the causes of variability, but it is not relevant in relation to the others, so it can be considered negligible. A statistical approach is useful yet not sufficient, optimization of each factor, in order to minimize the variance, is needed.

2.2 Input, output, potentiality, importance of the problem

Starting from a group of parts, the relative quantities, and the probability, the software can calculate the date in which the distribution has the given cumulative probability for both the case of parallel and serial production¹. It is also important to consider the inverse problem because it often happens that the customer places the order for a certain date without consulting the seller or simply asks if the given date is possible, therefore the software can also do the inverse calculation. Both functions require a list of items and the order date, which is the day when the order is received or can be fictitious if the tool is used to decide whether to accept an order.

Another function is the so-called "Best rate" that calculates the best mix of items to maximize the quantity-days rate with two settable constraints:

¹The production can be done in many ways: for this problem the two extremities are the serial production, each item (the whole quantity) is produced after the previous one, and the parallel production, each item is produced simultaneously

the probability that is important to know the days the mix will take to be produced, and the minimum percentage of quantity to release. In case of peaks of production, this can be really important because the order can be partially issued to satisfy a part of the customer demand, releasing the best mix brings to a higher service level. Still, the higher costs of smaller deliveries represent a problem that the company can solve with different policies, if the solution is in-between (the two extremities would be: wait until the full order is ready or send the single items once they are prepared), this tool can help to calculate which parts should be produced first.

The last function is probably one of the most important, it is a simulation of the production of the given order in a chosen period based on the previous years' data, it let the company tries the data in a real situation and see which improvements need to be made to optimize the production trying different combinations. The simulation, in contrast to the other functions, uses a random outcome of the random variables, there is no more a given amount of days, whereas every time the production duration changes, repeating the same simulation means having different results. It is possible to set some parameters:

- the simulation starting and ending date (must include the order date)
- the data year from the ones present in the data-set except for the current or last one
- the number of "lines", how many items can be produced or treated simultaneously. This is the only method that let the company choose them and do not consider only the two extremities
- the maximum number of days from the order reception to the start of production, if it takes more time, the line is lost and will not be produced

As well as showing the number of order processed, the tool also calculates the actual quantity percentage, the starting and ending date of the given order, the lost ones, the ones satisfied with the stock, the total idleness of the production lines, the orders that were started the same day they were received. The simulation is random, so a single simulation is not significant to make decisions, this is the reason why "Simulation+" has been added: this section let the user simulate the period performances for a chosen number of times, the order of magnitude of the number of simulations should not exceed tens of thousands to have an answer in a adequate time, in addition, the accuracy of a simulation of this kind does not need to be the highest one,

the results are pretty much the same once ten thousands simulations are run; anyway, it is possible to run more than a hundred thousand if needed.

3 Data

The used data-set belongs to a mechanical company that both produces and stocks material. This is an incomplete collection because the whole one would have been too big considering that there are thousands of parts; for this reason, only twenty-three components were selected with more the 5500 lines, so the size can be considered significant. The order lines begin in 2012 and the last data are basically year-to-date.

There are six columns:

- "Customer", who ordered the part, also indicate where the parts must be sent (another table of the database links the customer to its warehouse but it is not present in the used dataset); it is a string (VARCHAR) starting with "CUST_" and followed by an integer
- "Part_Number", is the alphanumeric code of seven digits and a letter that uniquely identifies the ordered component/product
- "Quantity", the integer number that represents how many units are ordered
- "Description", a brief description of the category of the item
- "Order_Date", this is not a good name for the column because it does not represent when the order arrived, it represents when the production of the line of the order was started instead
- "Good_Issue_Date", is the date when the line production ended, when it was issued from the production line

Due to lack of further data, the simulation uses each line as an order (which is a common case), so "lost orders" or other statistics of this kind cannot count the real orders but their lines. For a real application, some few and easy changes to both the data-set and the software are needed.

4 Software data structures and algorithms

4.1 A statistical approach: the due date quotation

From the analysis of a sample of items, the random variables cannot be considered normally distributed, in fact, the Anderson-Darling normality test usually has a p-value which is less than 0,005.

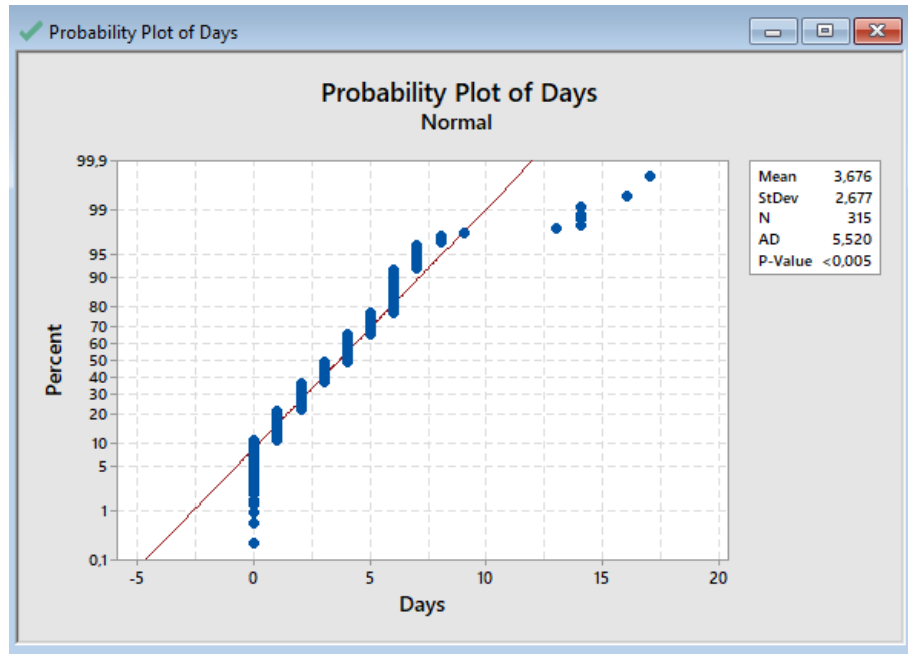


Figure 1: Example of normality test run with Minitab®

The reason could be that every value below zero (e.g. stock already present some days before the order date due to forecast orders) is converted into zero so data differ from the normal distribution. Anyway, this does not prove that the data is not following a normal distribution, in fact most of the data follow the days-percent line (see *Figure 1*). A further analysis could confirm or reject this hypothesis, but, for the purpose of this thesis, they are considered normally distributed; an improved version could include the recognition of each item distribution.

Since we can consider the items normally distributed, it is easy to calculate the mean and variance of a group of variables: the sum of two or more random variables that are normally distributed, is normally distributed as

well:

$$X \sim \mathcal{N}(\mu, \sigma) \quad \text{where} \quad \mu = \sum_i \mu_i \quad \text{and} \quad \sigma = \sqrt{\sum_i \sigma_i^2}$$

The standard deviation does not consider the covariance, the random variables are assumed independent because there is not enough information to be sure of a correlation, also, there are few items that share sub-components so, it can be considered negligible: this is a simplification.

By using the "Normal distribution"² class of Apache Common Math, it is possible to obtain the p-quantile with the "inverseCumulativeProbability(double p)" method, then it is turned into a discrete value³ that represents the maximum number of days to issue the order with a p-probability.

The same is done to find the cumulative probability of a certain date: the number of days is turned into a continuous value by adding less than 0.5 (maximum of the chances, see note³) and the cumulative probability is obtained with the "cumulativeProbability(double x)" method.

In case of parallel production: the quotation is done by considering each item alone, the largest number of days is the result; the probability is found by multiplying each probability, every item has a certain probability of being prepared by the chosen date:

$$\mathbb{P} = \prod_i \mathbb{P}_i$$

4.2 Best rate algorithm

This function uses the recursion: all the combinations are tried, and the best result is shown. Starting from one item of the order, each other combination (without repetitions) is tried and, if the mix is within the constraints, the rate is calculated by using the due date quoting algorithm (see paragraph 4.1) for the number of production days. If the rate is greater, it becomes the new best rate. The recursive method calls itself adding an item until the termination conditions become true, then it deletes an item and tries again with another one, this way it considers all the possibilities. Since all the combinations are tried, the number of items is relevant, it is possible to obtain good computational times if the number of items is less than ten ($10! = 3628800$ cycles multiplied by all the operations of each one).

²<https://commons.apache.org/proper/commons-math/apidocs/index.html>

³When the decimal portion is greater than 0.5, a day is added. Another choice could have been to add a day when the decimal portion is greater than zero.

4.3 Simulation algorithm

This is a discrete-event simulation: the orders from the selected year are added to a PriorityQueue⁴, so are the items of the given order. All the dates have the same year, i.e. the simulation start year. The Line objects are created, and the simulation starts: the less recent order is selected, the order date becomes the "current date" and a free line is chosen; in case of many free lines, the one with the greater idleness is picked; in case no line is free, the first one that is freed is picked and the "current day" changes to the first day of availability of the line. This way, all the orders are processed until the current date exceeds the simulation ending date or all the orders are treated. In the meantime, the statistics are updated in accordance with the happened event.

For the "Simulation+" section, the simulation algorithm is repeated for the chosen number of times and the results are added each time and, at the end, divided by the simulations number to calculate the mean values of the statistics.

From the statistical side, the simulation does not lean on the due date algorithm (see paragraph 4.1), instead, all the items are considered individually because the number of production days for the part is the outcome of a random variable, so the addition of the single random variables is no more needed, there are now numbers to be added and the random variables are set aside.

⁴<https://docs.oracle.com/javase/8/docs/api/java/util/PriorityQueue.html>

5 Java classes

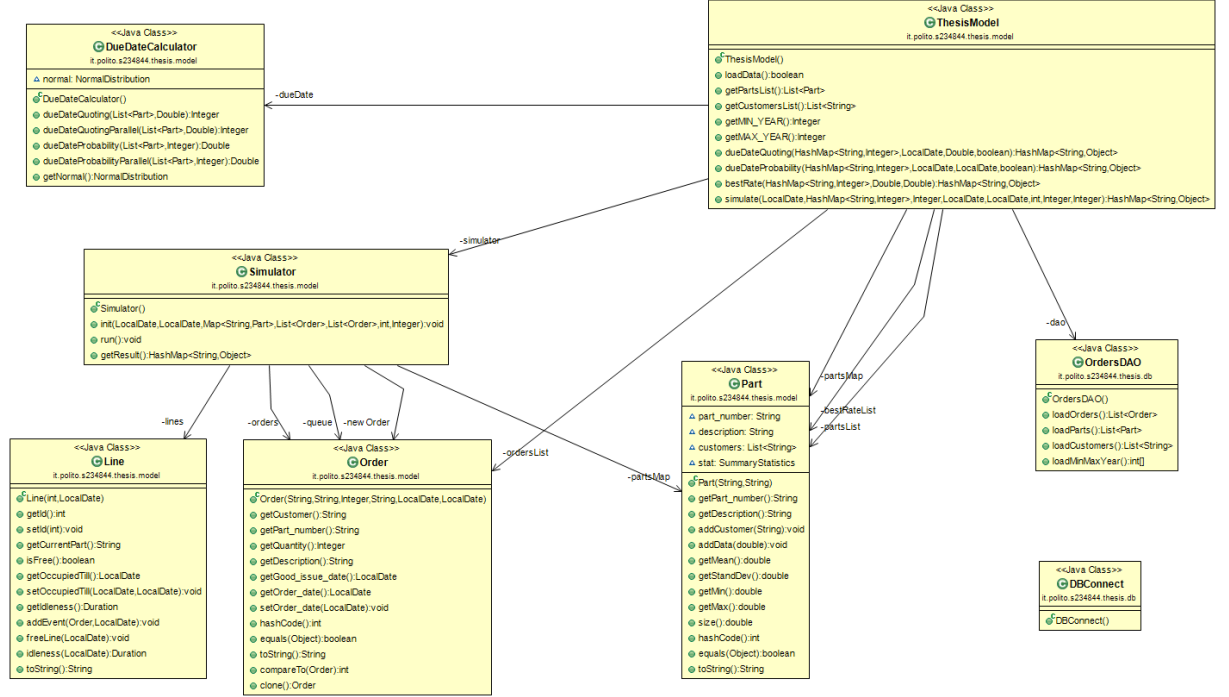


Figure 2: Most important classes diagram. Please notice that private variables and methods were not included

ThesisModel is the class that manages all the requests from the user interface: connects to *OrdersDAO* to load the data from the database (using the connection given by *DBConnect* class); manages the requests for the due date quotation and probability thanks to the *DueDataCalculator* class; starts and runs the recursion for the "Best Rate"; connects to the *Simulator* to manage both the single simulation and the "Simulation +".

The whole system works using classes that represents the parts (class *Part*), the orders (class *Order*), and the production lines (class *Line*).

In *Figure 3*, there are some missing classes, most of them are from the *it.polito.s234844.thesis* package which contains the controllers for the FXML files, that are the view and controller parts of the model-view-controller pattern.

6 Software screenshots and video

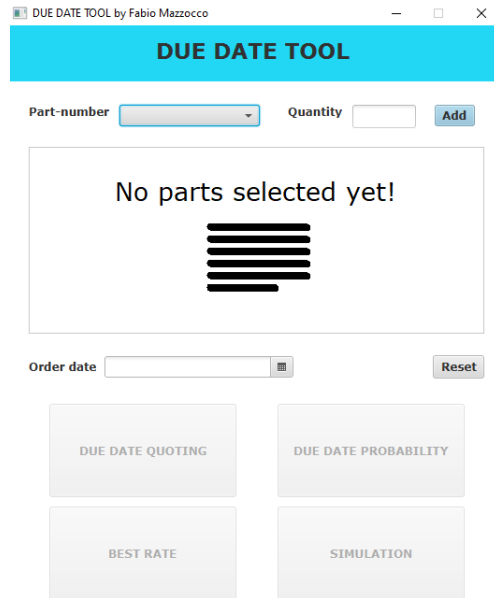


Figure 3: Screenshot of the software's homepage

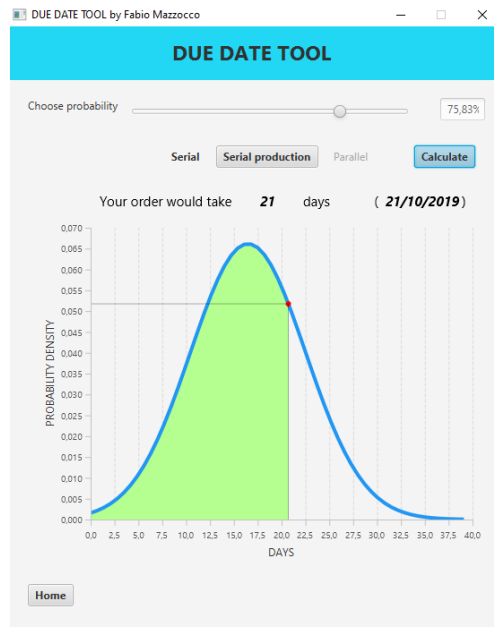


Figure 4: Screenshot of the "Due Date Quoting" function

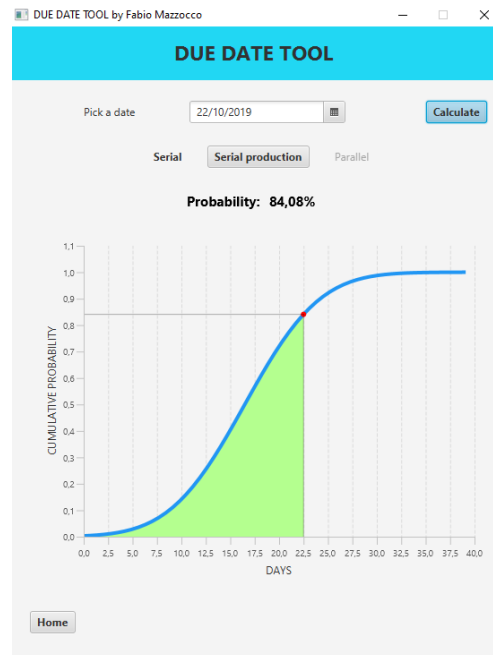


Figure 5: Screenshot of the "Due Date Probability" function

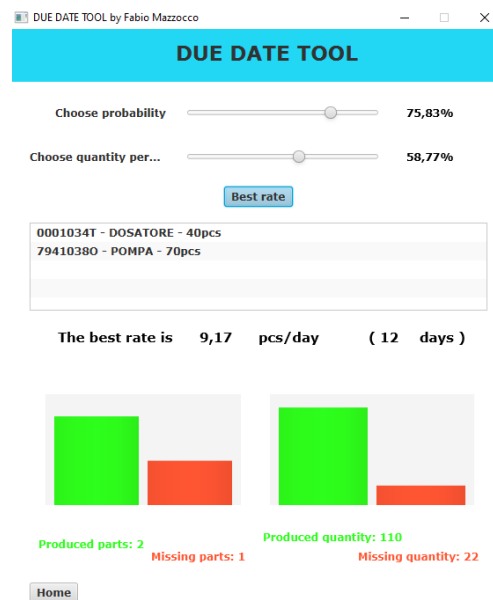


Figure 6: Screenshot of the "Best rate" function

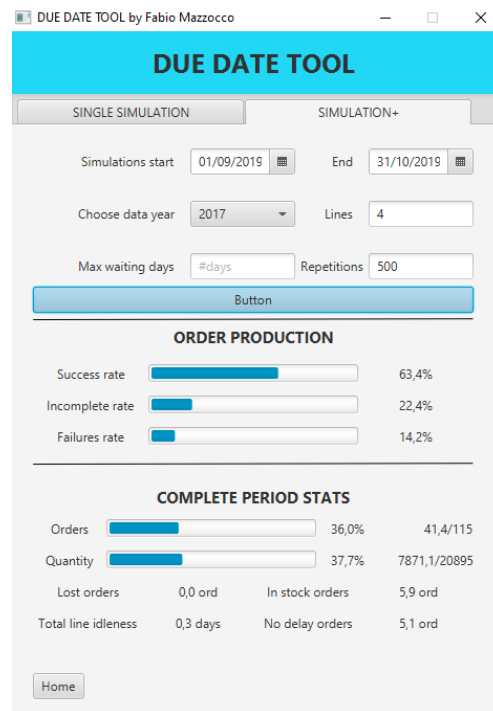


Figure 7: Screenshot of the "Simulation+" function

It is available a quick "Get started guide" on Youtube: [INSERIRE LINK](#)

7 Tests, results and final observations

Thanks to this tool, some main decisions can be taken, the due date quotation can be made consciously. Clearly, the software cannot consider the current status of the production so it does not differentiate the cases, it is crucial to follow the given due date but it is also important to supply the customer as fast as possible, so a further version of the tool could consider the environmental status to provide a more accurate due date forecast.

Items	Mean time (20 attempts)
2	7 ms
3	7.2 ms
4	7.4 ms
5	11.7 ms
6	40.2 ms
7	201 ms
8	1548 ms
9	14120 ms

Table 1: "Best rate" times with different number of items, mean value on 20 attempts

The "Best Rate" function is a pure decision-making tool, it could be important in case of production peaks but, unless all the parts are equal under all points of view, the output is rough, some more evaluations need to be made in term of revenue and importance of the item. Processing times (available in *Table 1*) underline the importance of the number of items of the order in providing an output in a reasonable time.

#Simulations	Mean time (5 attempts)
10	145 ms
100	253 ms
1000	1450 ms
10000	13200 ms
100000	134000 ms

Table 2: "Simulation+" processing times, mean value on 5 attempts (2 months long period, 4 lines, 1 attempt for each of the last 5 years)

The "Simulation+" function could really help in deciding, the decision-maker

can see a real-world case of the acceptance (or the performances) of the order in many attempts so that can see the possible outcomes and their variability, this is a the greatest contribution to the problem, it would be perfect if all the orders of that period were already known. Still, there could be more constraints: giving priority to some customer, the exclusion of some non-working days or public holidays, the loading of a mix of orders from the past years instead of taking them all from a single year. Anyway, the possibility to simulate more than an hundred thousand times in two minutes (see *Table 2*) is more than enough, in fact, after ten thousand simulations, the values tends to level off. In the *Table 2* case, a decision can be taken in about 13 seconds with the data of ten thousand different simulation output put together.



This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA).

Read the full copy of the license at
<http://creativecommons.org/licenses/by-nc-sa/4.0>