IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684,p-ISSN: 2320-334X, Volume 15, Issue 6 Ver. II (Nov. - Dec. 2018), PP 26-38
www.iosrjournals.org

Optimizing Shared Resources in a Batch Production Process using Genetic Algorithm for Small Production Industry for Different Batch Condition

Pushpendra Singh Chahar[1], Gaurav Kumar Chhaparwal[2]
Raghav Singh Dhakar[3], Sharad Shrivastava[4]
[1,2,3,4](*Department of Mechanical Engineering, Rajasthan Institute of Engineering & Technology, Jaipur, India)*
Corresponding Author: Pushpendra Singh Chahar

Abstract: *Mathematical optimization is a process for finding and selecting the best fit values that is used to solve various engineering problems. Along with a complex area, it is important to improve productivity and efficiency for a company or even an industry. Based on a large number of literature reviews, intelligent scheduling methods are the most effective way to solve practical complex scheduling problems.*
The goal of this research is to increase the comprehensive production effectiveness for small production systems. The research is based on the actual demand of the production batch process and provides the ideal solution to the current problems. This approach considers the approach when randomly placed orders are deposited in the form of batches for production, and look for application of intelligent control under realistic conditions. As a result, it is important to develop a new model of production system which takes into account some obstacles. The problem that focuses on the production system is to reduce the time spent by the mathematical model. The optimization problem is solved using intelligent algorithms, which include Genetic Algorithm (GA) while providing some effective improvement strategies based on the lack of application.

Date of Submission: 26-11-2018 Date of acceptance: 07-12-2018

I. Introduction

Their use in industry is increasing, including genetic algorithms (GA), simulate annealing (SA), ant colony optimization (ACO), taboo search (TS) etc. Stark weather, et al. (1992) has implemented a GA for resolving multipurpose JSP in Brewery, these objectives include the minimum average time of the list, the minimum wait time of the customer's orders etc. It has also been implemented for optimization based on GA's dynamic model. In the latter part, GA is used to obtain a series of curves at different temperatures during certain fermentation time to find the optimum temperature curve. This ensures that the ultimate quantity of alcohol in the fermentation process reaches the maximum and also ensures that the concentration of the product is lowest, which protects the quality of beer. It produced mathematical models for product scheduling and showed results that the customized plan has strangeness and practicality. Compared to intelligent control, intelligent optimization technology is still relatively rare in industry, and there are many practical issues that need to be adapted and resolved [2].

The purpose of this research is to achieve significant improvement in efficiency and performance in the breweries production system to meet the different market demands. Based on the development of modeling approach, algorithm analysis, and optimization techniques, which reduce production time and maximize profits. The research objectives of this study can be divided into specific parts to achieve the goal:

1. In order to better understand existing problems in this domain and to investigate the existing literature available on optimization methods for the scheduling problems of the manufacturing production system, to look for some possible ways to solve problems.
2. To analyze logic and development of GA in different domains.
3. Prepare problems using mathematical models and learn the barriers and conditions.
4. To simulate the scenario of Brewery production system to improve performance and efficiency based on simulation results obtained by Simulink models. The following tasks should be achieved:

a. To obtain the result of the sequence of orders
b. To obtain the result of the accumulated batch based on the decision making.
c. To obtain the result of the different routes of operations
d. To obtain the result of the total production

5. Developing and implementing the heuristic algorithm to optimize the real-life production system to reduce total production time. The following tasks are considered in this section.

a. To apply the GA to optimize the brewery production system
b. To modify and improve the GA
c. To validate the hybrid algorithm as contrasted with other algorithms for optimizing a production system.

II. Relevant Heuristic Algorithms

Genetic algorithms (GA)

Genetic algorithms were proposed by John Holland (1975). It is inspired by the biological evolution of random search algorithms based on natural selection and natural genetic mechanisms, to solve both bound and uncontrolled optimization problems, the process of GA is to revise the population of individual solutions repeatedly. In each stage, the current population will be randomly selected to be parents who will produce children for the next generation. In generations continuously, the population develops towards an optimal solution. Similarly, (Michelle 1998; Shaw et al 2000) has defined that GA is a biological simulation in the natural environment of the adaptive genetic and evolutionary process, which is a kind of adaptive capacity and a global search prospect. The probable solution of each problem will be considered as a population (chromosomal) of the population, which will make the cluster of each chromosome in the form of encoding, for each person to provide the assessment according to the predetermined purpose, and also provide a fitness value. The algorithm will be based on the fitness value of its search process. In each stage, the three main types of rules are used to make the next generation from the current population by the selection of three genetic operators, crossover and mutation.

Encoding

Encoding is the primary problem that needs to be solved by GA, the Holland coding method is binary code, but this simple coding method is difficult to describe directly to the nature of problem in many GAS applications, especially in industrial engineering.

Initial population and the evaluation of fitness

First of all, it requires determining the number of people in the population, that is the size of the population (pop size), and then randomly generate the initial population using the fitness function to evaluate the performance of each of the early species. The initial solution that is to calculate the fitness of each initial solution, if the fitness is high, then the individual performance is better, and then it is close to optimal objectives, so the definition of fitness function plays an important role in GA.

In addition, many GA solutions require a significant amount of time to calculate some practical problems, usually its size is large, and for many individuals, especially in calculating and evaluating, more important genetic and evolutionary operations need to be implemented. Large number of fitness can be the reason for low efficiency of this evolutionary calculation process, and fail to meet the requirements of speed calculation. It is believed that GA is likely to have parallel processing, so many parallel GAs have been proposed in the past decades (Tomassini 1995; Concrete 2004). These algorithms have also obtained better customization quality than classical GA.

Selection

On the basis of the size of the selection fitness is to select the better person for the next generation, so it guarantees the population of development. The selection operation is better to choose and reduce the survival of the best of the population is the operation. Individuals have high potential for high fitness for the selection of large groups in the next generation; for lower fitness individuals, there are fewer possibilities to choose for next generation. The task of selecting the operation is to select some individuals from parents according to the following methods:

- **Roulette selection**
- **Rank selection**
- **Tournament selection**
- **Elitism selection**

Recombination

Recombination is a new method to generate a new individual (chromosome), then to recombine after the selection, the most common method of the recombination that includes crossover and mutation.

- **Crossover**

Crossover is the operation where two individuals as the parent are chosen by the selection methods to generate, and replace the two new individuals. Crossover operation is the most important feature of GAs, which is different from other evolutionary algorithms. It plays a key role in the convergences of GAs. Chromosome crossover operation is executed in accordance with a certain probability, called P_c, so that it has $P_c \times$ *popsize* individual for crossover operation. More specifically, each individual will generate a random number *r* in between 0 and 1, if $r \leq P_c$, the individual will be selected for crossover. Then it will randomly match pairs of the chromosome to generate a random number *pos* (where *pos*=1...*m*-1,*m*.*m* is the number of genes in the chromosome), *pos*is a crossover point which is for crossover and replace of an individual gene. Crossover can be described as following ('|' is the crossover point) in Figure 1:

There are also other methods to make the crossover, such as two crossover points, uniform, etc. specific crossover made for a specific problem can improve the performance of the GA. In the GA, it is necessary to pair the individuals in the population before the crossover operation, and the common matching strategy is random matching. Crossover operator is normally designed to include the contents of two aspects: how to determine the position of the cross point? How to carry out the exchange of genes? Here some kinds of crossover operator have been classified that are applicable to binary coding or real number coding as follows (Barros, de Carvalho& Freitas 2015):

Before:

Chromosome 1: **11011 | 00100110110**

Chromosome 2: 11011 | 11000011110

After:

Offspring 1: **11011** | 11000011110

Offspring 2: 11011 | **00100110110**

Figure no.1. Example of crossover operation at the single crossover points

1) Single point crossover; also known as the simple crossover, which is to select one crossover point randomly in the individual encoding cluster, and then to exchange a pair of individual parts of the gene at that point.
2) Two-point crossover; the specific procedure of implementation is to select two crossover points in the pairing between two individuals of the encoded string, and then to exchange the part of the genes at two crossover points.
3) Uniform crossover; this refers to every gene in two pairs of individuals having the same probability to exchange, so as to form two new individuals.
4) Arithmetic crossover; it refers to the linear combination of two individuals in order to generate new individuals.

- Mutation

The mutation operation is defined when some of the gene values of the individual encoding cluster are randomly rearranged from the crossover operations, so as to mutate, and then to obtain a new individual. Mutation is intended to break one or more individual and to jump out of a local optimum to discover a better minimum or maximum space. It maintains genetic diversity and avoids premature convergence on a local minimum or maximum. A mutation operation can also be described by binary encoding as follows:

Before:

Original offspring 1: 1101100100110110

Original offspring 2: 1101111000011110

After:

Mutated offspring 1: 1101110100110110

Mutated offspring 2: 1111111000011010

Figure no.2. Example of a mutation operation

In the example in Figure .2, we have selected one and two random values corresponding to the bit length of the chromosome. In this case, 6 have been selected in the original offspring 1; 3 and 14 have been selected in the original offspring 2. Then simply take the bits from the chromosome and swap them. Chromosome mutation operation is determined by the specified P_m. The design of mutation operation includes two aspects: how to determine the mutation position? How to process the replacement of gene value? Some types of mutation operators have been classified that are applicable to binary coding or real number coding as follows (Barros, de Carvalho& Freitas 2015): 1) Flip bit; 2) Boundary; 3) Uniform mutation; 4) Non-uniform; 5) Gaussian.

All in all, crossover and mutation have both co-operation and competition (Eiben& Smith 2003). Crossover is explorative that to discover promising areas in the search space, such as gaining information on the problem. It makes a big jump to an area somewhere in between two (parent) areas; mutation is exploitative that to optimize within a promising area, such as using information. It creates random small diversions, thereby staying near the area of the parent.

Convergence

For the selection, crossover and mutation operation, in order to produce new species, the fitness of the new species is requested to be evaluated. The above steps are repeated until the algorithm reaches a pre-determined condition, or the fitness of the population will no longer increase.

III. Result

Modelling and optimize the use of shared resources in a system, to identify resource deficiencies and improve capacity planning. The is based on a batch production process, where production orders are processed only according to the availability of batch reactors. This represent both the production orders of the manufacturing process, and the batch reactors that are required to process them. Then it will find the optimal resource capacities of the system by applying the Genetic Algorithm solver of MATLAB.

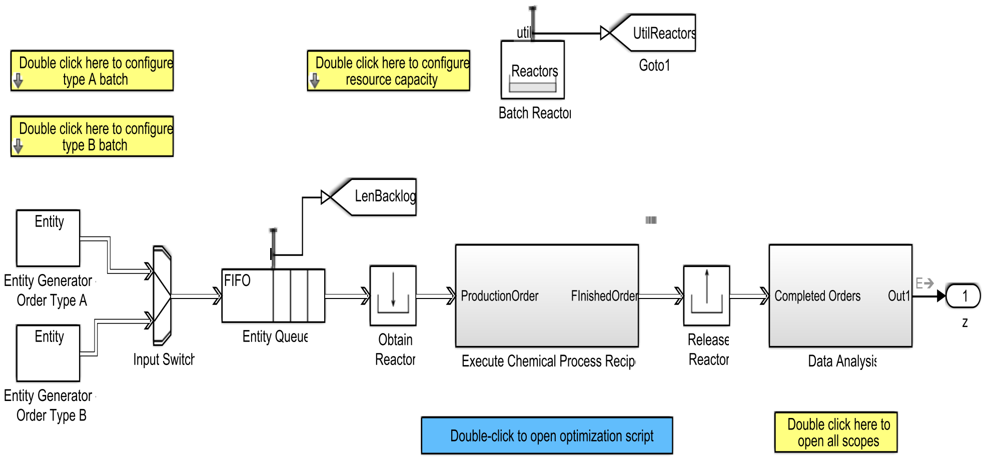


Figure no.3. MATLAB model for Batch Production Process

Structure of Model

At the top level of the model, the Entity Generators simulate the generation and backlog of production orders by generating entities that represent production orders. When a new entity is generated, the Obtain Reactor block requests a batch reactor to process the order. After the Execute Chemical Process Recipe subsystem completes the order according to a specified chemical process recipe, the block labeled Release Reactor releases the batch reactor back into the pool of resources, where it is now available to process a new order. The Data Analysis subsystem analyzes data related to completion of production orders.

Shared Resources in the Production Process

The Execute Chemical Process Recipe subsystem simulates the chemical process to produce sol (a type of colloid). A six-step recipe models the main operations in sol production. Execution of these steps requires different resources. A batch reactor provides built-in capabilities to execute steps like adding color, adding particles and stir. Thus the resources required by these steps do not need to be modeled separately. On the other hand, the steps to add water, heat up and drain require extra resources. These resources are shared by all the batch reactors and are limited by the capacity of the production system.

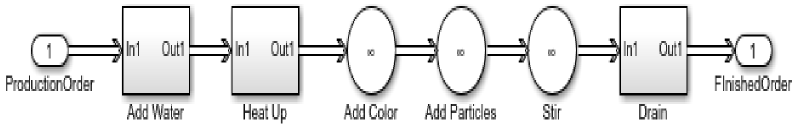


Figure no.4. Block Diagram for process control of water, heat and drain.

For example, when water usage reaches the full capacity, water pressure is too low for another batch reactor to access. In this case, production in that reactor pauses until the water supply becomes available again. In the Execute Chemical Process Recipe subsystem, the example models such a resource sharing process with a Queue block labeled Wait for Water Supply and an Entity Server block labeled Add Water in the Add Water subsystem. The Capacity parameter of the Entity Server block models the capacity of the water supply. During simulation, the number of entities in the Queue block indicates the number of batch reactors waiting for water. The number of entities in the Server block represents the number of batch reactors accessing water.

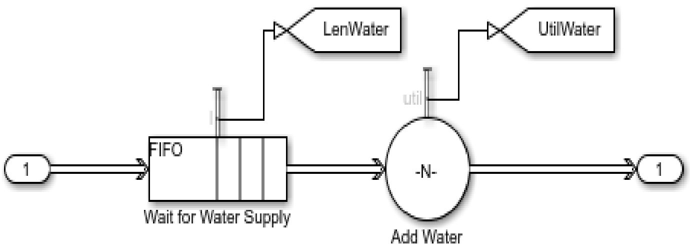


Figure no.5. Water adding and supply system for process

The modeled batch production process is capable of producing two types of batches: type A and type B. Although the main steps required to produce either batch are the same, the chemical process recipes are different. For example, the recipe to produce type B requires more water, so the step to add water takes more time to complete.

Table no.1. Variation in parameter variation of three different conditions of industry.

TYPE A BATCH

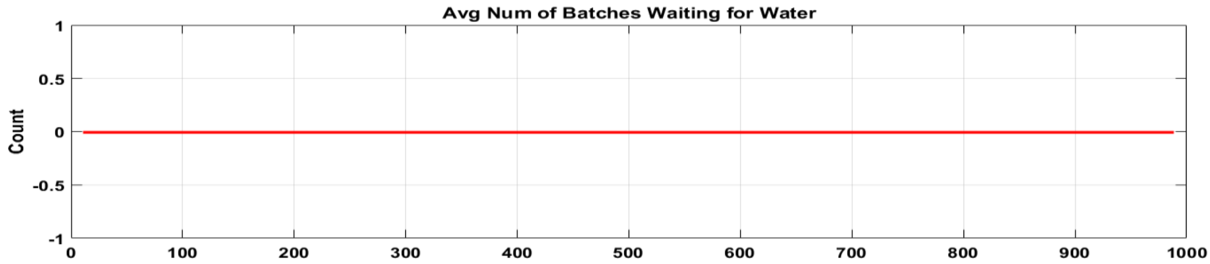
S.NO	PARAMETERS	VALUE-1	VALUE-2	VALUE-3
1	Order generation period of type A batch:	10	20	40
2	Time needed to add water:	10	20	40
3	Time needed to heat up:	20	40	80
4	Time needed to add color:	3	6	12
5	Time needed to add particles:	5	10	20
6	Time needed for agitation:	10	20	40
7	Time needed to drain:	10	20	40

TYPE B BATCH

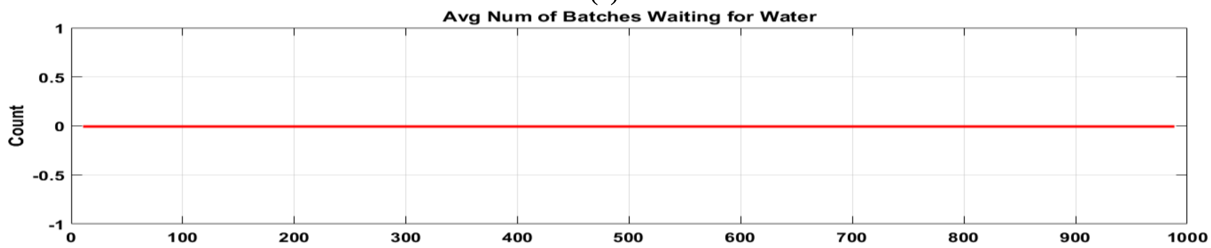
S.NO	PARAMETERS	VALUE-1	VALUE-2	VALUE-3
1	Order generation period of type B batch:	15	30	60
2	Time needed to add water:	15	30	60
3	Time needed to heat up:	30	60	120
4	Time needed to add color:	6	12	24
5	Time needed to add particles	10	20	40
6	Time needed for agitation:	20	40	80
7	Time needed to drain:	15	30	60

RESOURCE CAPACITY

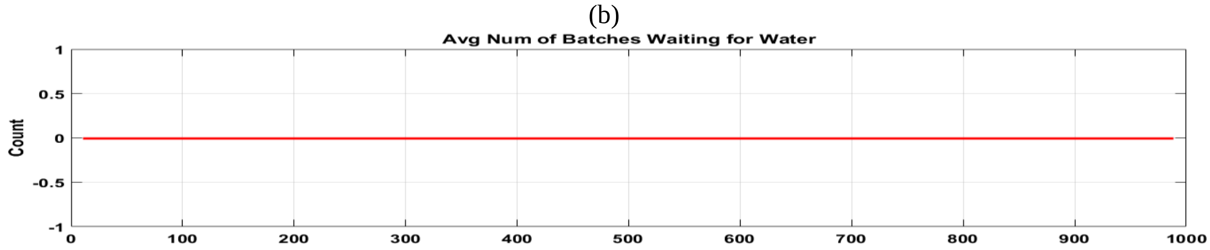
S.NO	PARAMETERS	VALUE-1	VALUE-2	VALUE-3
1	Number of batch reactors	2	4	8
2	Capacity of water supply (number of reactors that can access simultaneously)	2	4	8
3	Capacity of heat supply (number of reactors that can access simultaneously)	2	4	8
4	Capacity of draining facility (number of reactors that can access simultaneously)	2	4	8



(a)



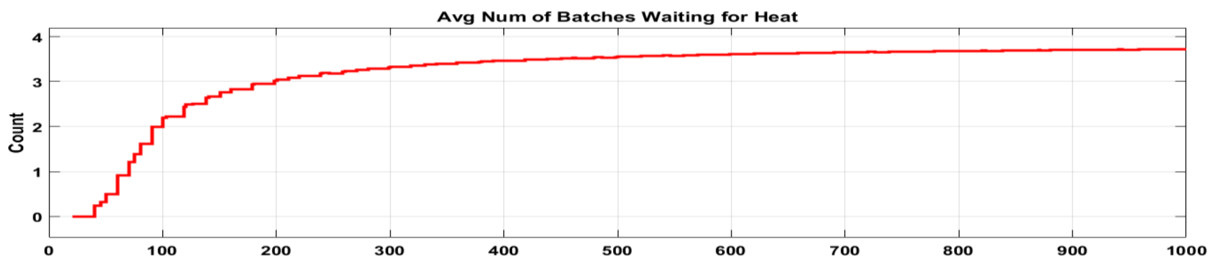
(b)



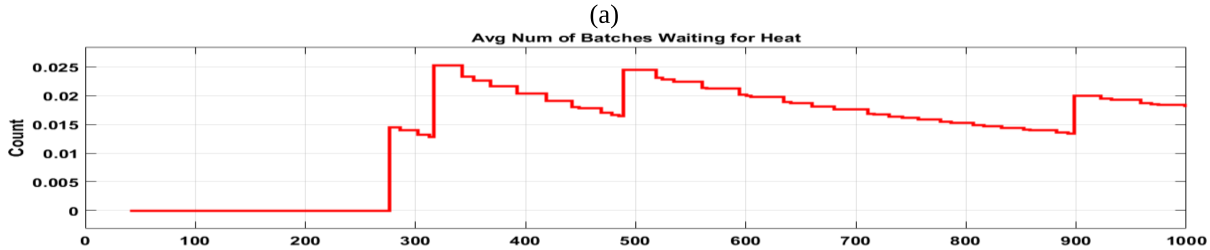
(c)

Figure no. 6. (a-c) Waveform for average number of batches waiting for water in different conditions

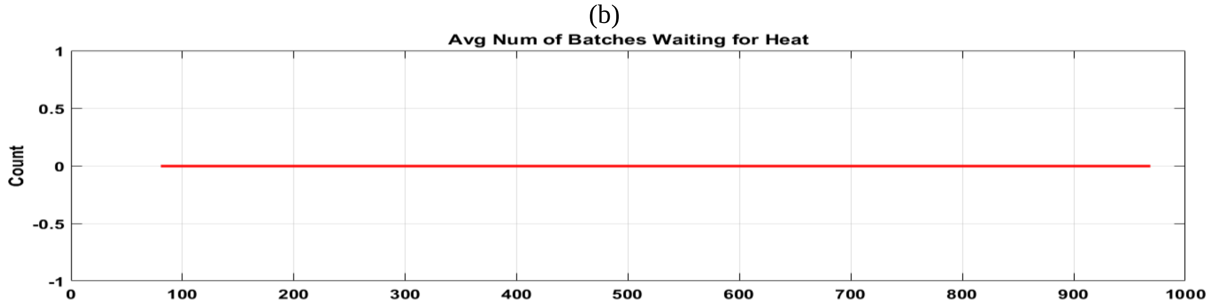
As shown in figure 6. the average number of batches waiting for water is Zero after optimization, so it is clear that the process is at most optimized condition in all three stages of parameters.



(a)



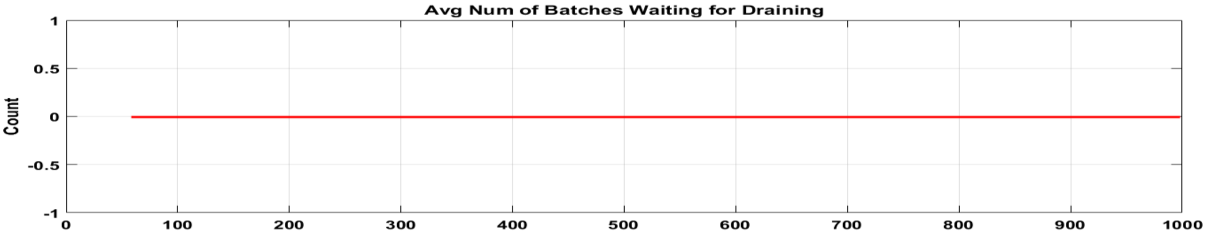
(b)



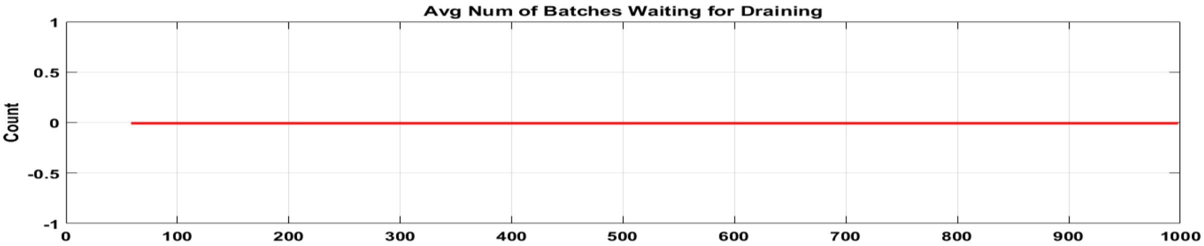
(c)

Figure no.7. (a-c) Waveform for average number of batches waiting for heat in different conditions

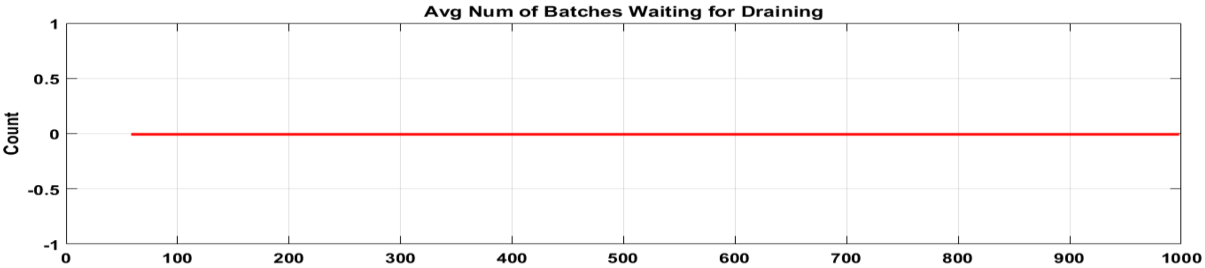
As shown in figure 7. the average number of batches waiting for heat varies in condition a and b, where it is Zero in c after optimization, so it is clear that the process is at most optimized condition in third stages of parameters.



(a)



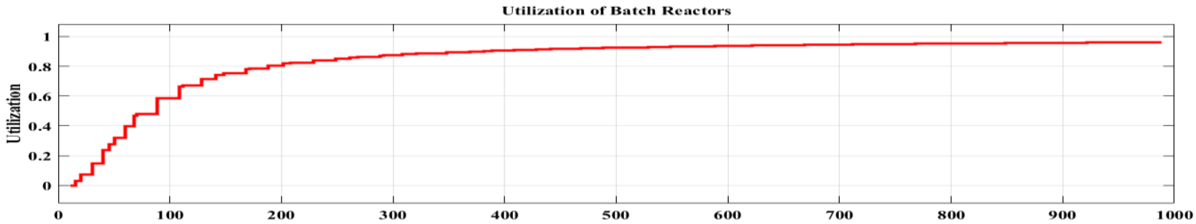
(b)



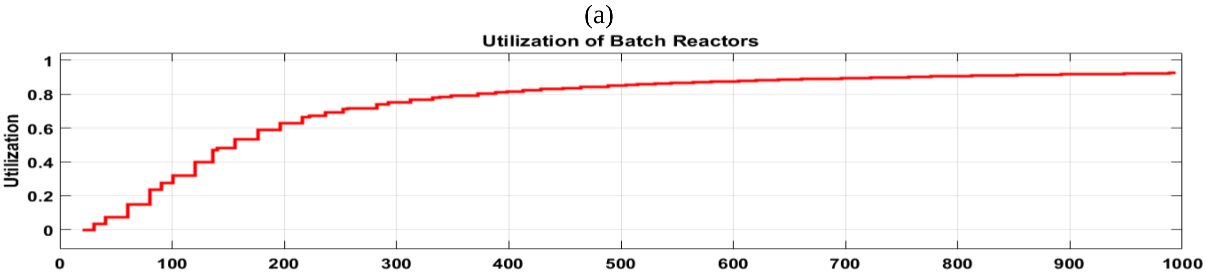
(c)

Figure no.8. (a-c) Waveform for average number of batches waiting for Draining in different conditions

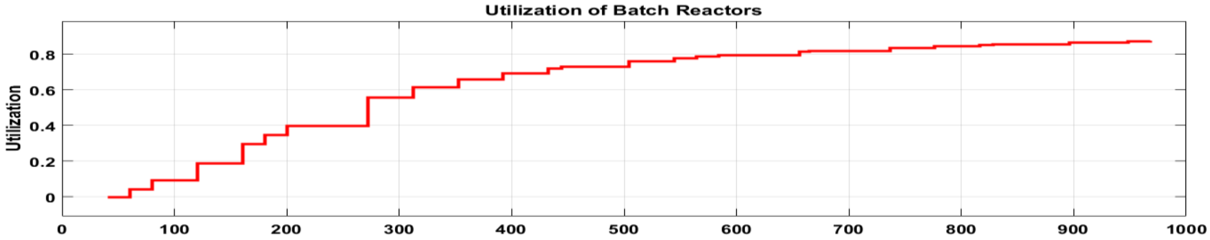
As shown in figure 8. the average number of batches waiting for Draining is Zero after optimization, so it is clear that the process is at most optimized condition in all three stages of parameters.



(a)



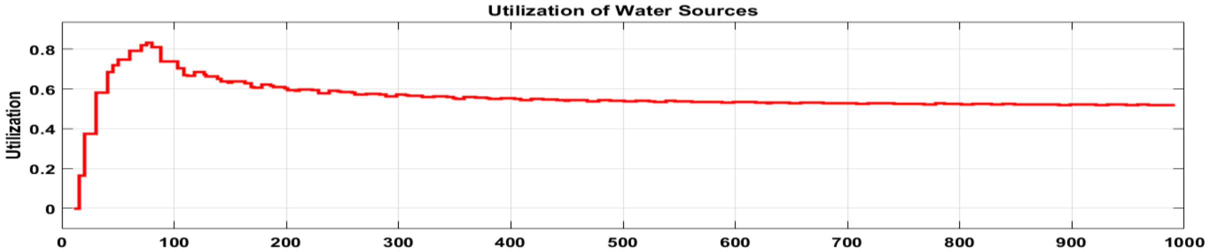
(b)



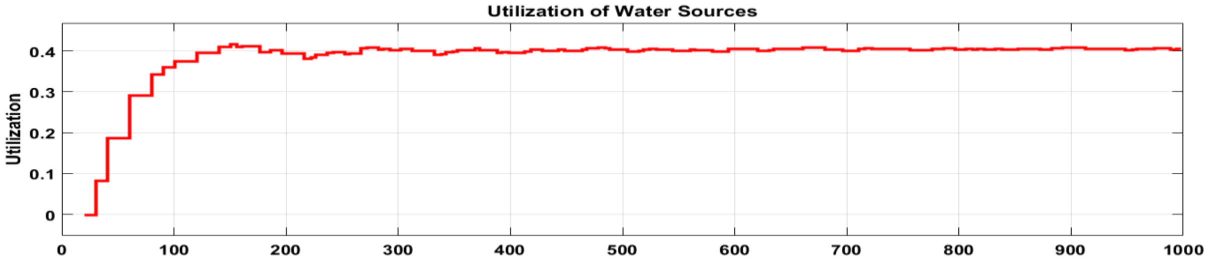
(c)

Figure no. 9. (a-c) Waveform for Utilization of Batch Reactors in different conditions

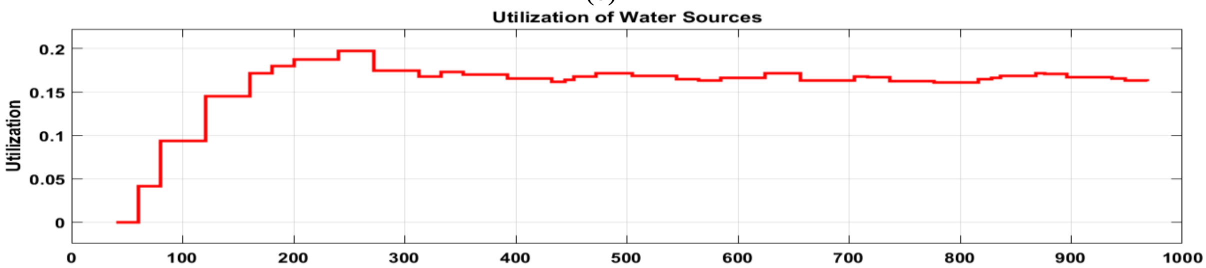
As shown in figure 9. the Utilization of Batch Reactors variable between 0 to 0.8 for all the conditions, where for condition a and b its stable to time but in c its variable for time.



(a)



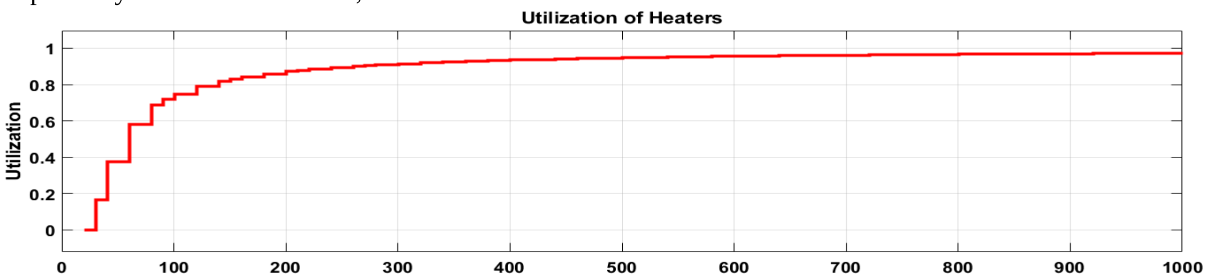
(b)



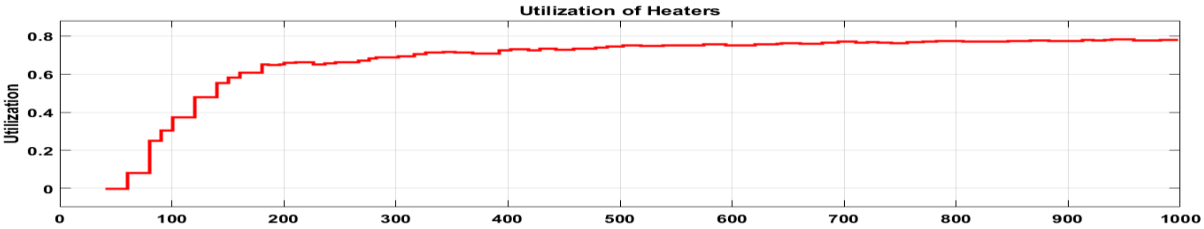
(c)

Figure no. 10. (a-c) Waveform for Utilization of Water Sources in different conditions

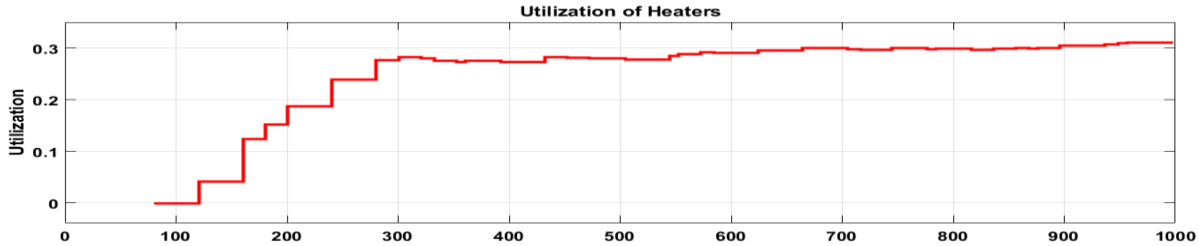
As shown in figure 10. the Utilization of Water Sources variable between 0 to 0.8, 0 to 0.4 and 0 to 0.2 respectively for all the conditions, where for condition a and b its stable to time but in c its variable for time.



(a)



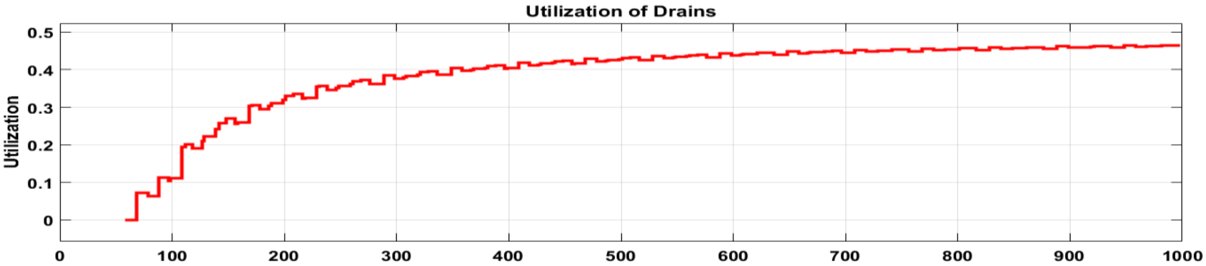
(b)



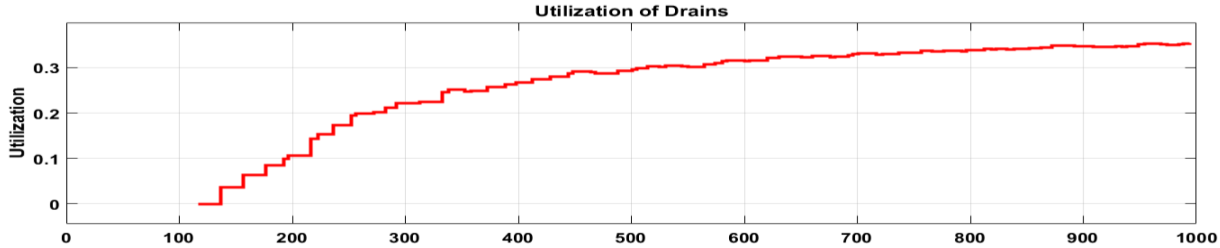
(c)

Figure no 11. (a-c) Waveform for Utilization of Heaters in different conditions

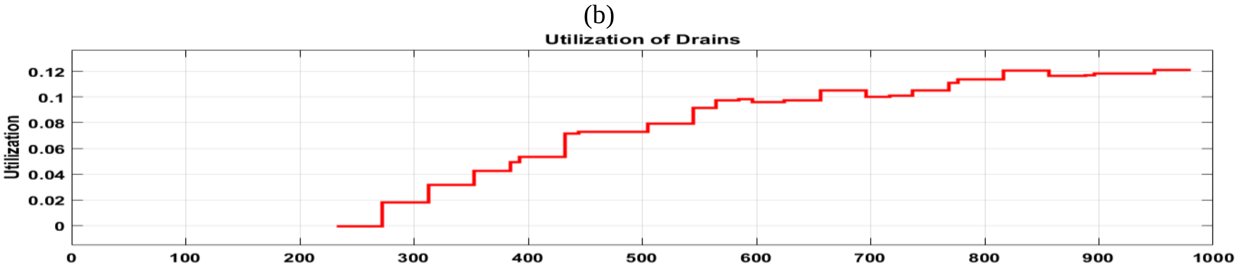
As shown in figure 11. the Utilization of Heaters variable between 0 to 1, 0 to 0.8 and 0 to 0.3 respectively for all the conditions, where for condition a and b its stable to time but in c its variable for time.



(a)



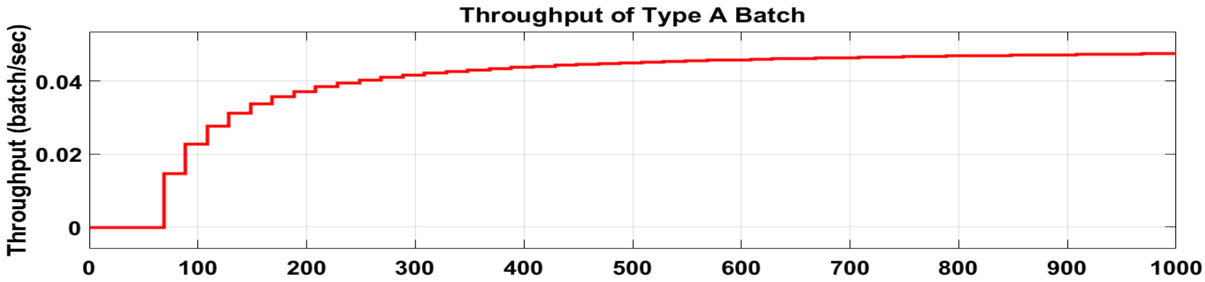
(b)



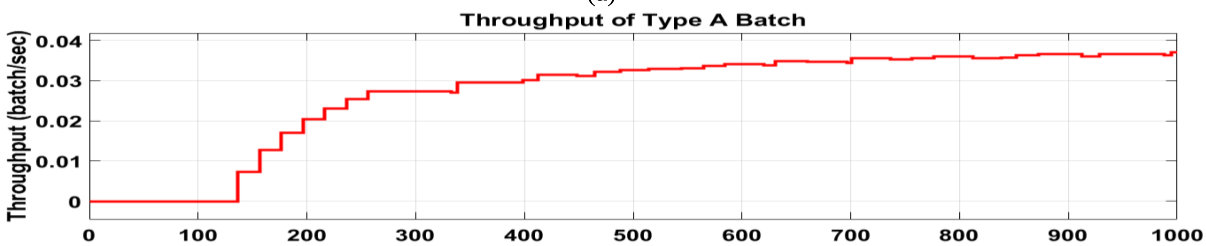
(c)

Figure no. 12. (a-c) Waveform for Utilization of Drains in different conditions

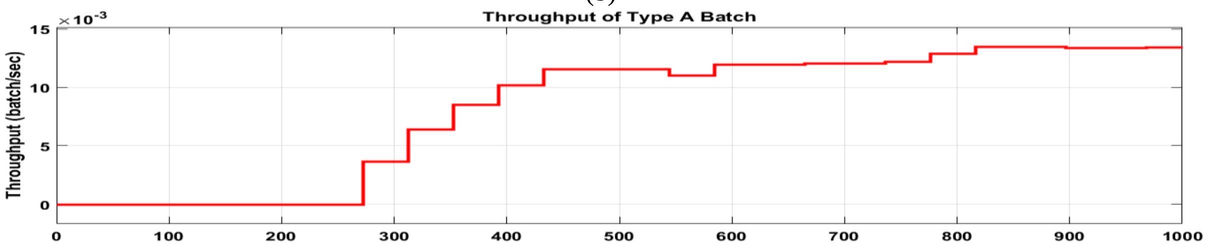
As shown in figure 12. the Utilization of Drains variable between 0 to 0.5, 0 to 0.3 and 0 to 0.12 respectively for all the conditions, where for condition a and b its stable to time but in c its variable for time.



(a)



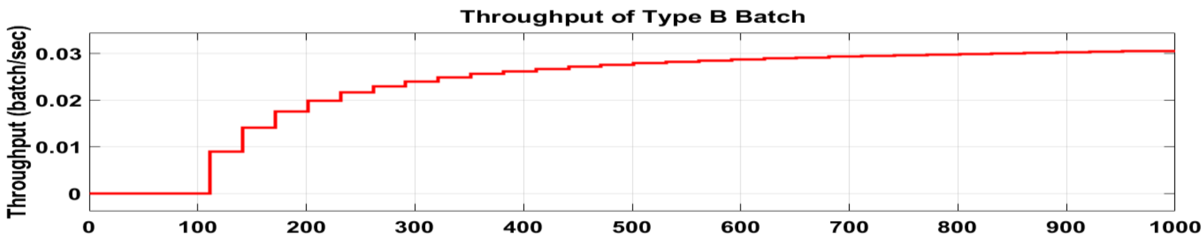
(b)



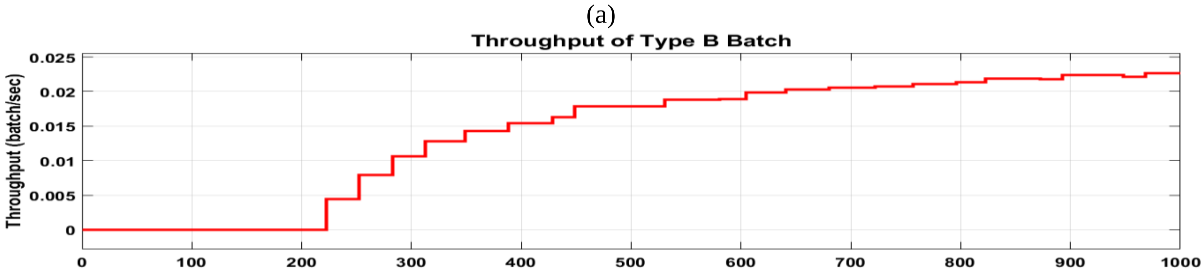
(c)

Figure no.13. (a-c) Waveform for Throughput of Type A Batch in different conditions

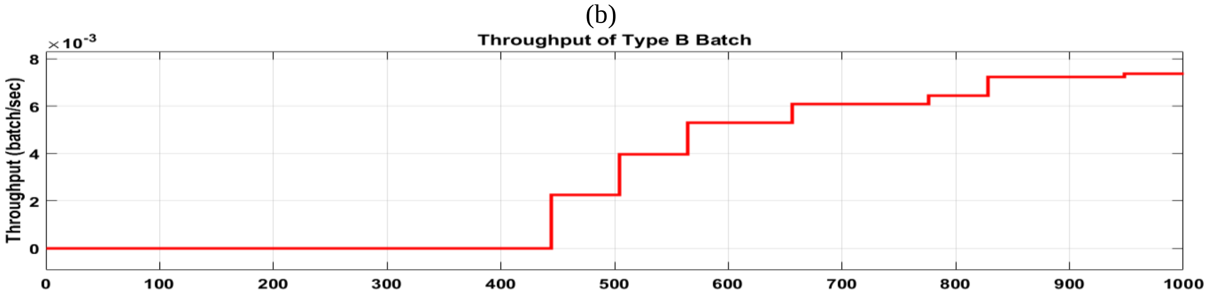
As shown in figure 13. the Throughput of Type A Batch variable between 0 to 0.5, 0 to 0.4 and 0 to 0.015 respectively for all the conditions, where for condition a and b its stable to time but in c its variable for time.



(a)



(b)



(c)

Figure no. 14. (a-c) Waveform for Utilization of Batch Reactors in different conditions

As shown in figure 14. the Throughput of Type B Batch variable between 0 to 0.3, 0 to 0.025and 0 to 0.008 respectively for all the conditions, where for condition a and b its stable to time but in c its variable for time.

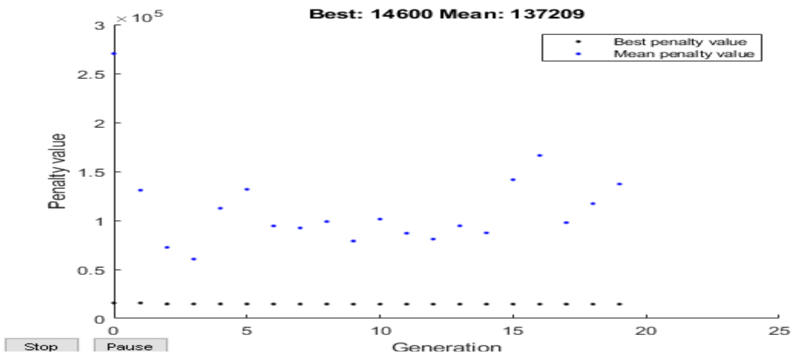
Optimizing Resource Capacities

Now apply a Genetic Algorithm solver from the MATLAB Global Optimization Toolbox to this Sim Events model to find optimal resource capacities for this system. The genetic algorithm solves optimization problems by repeatedly modifying a population of individual points. Due to its random nature, the genetic algorithm improves your chances of finding a global solution. It does not require the functions to be differentiable or continuous.

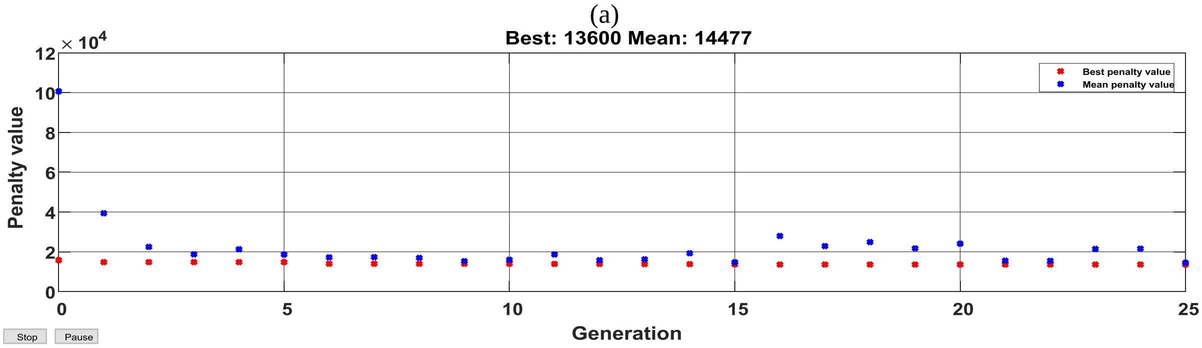
The decision variables in this optimization are:

- Number of batch reactors
- Number of water tanks
- Number of heaters
- Number of drains

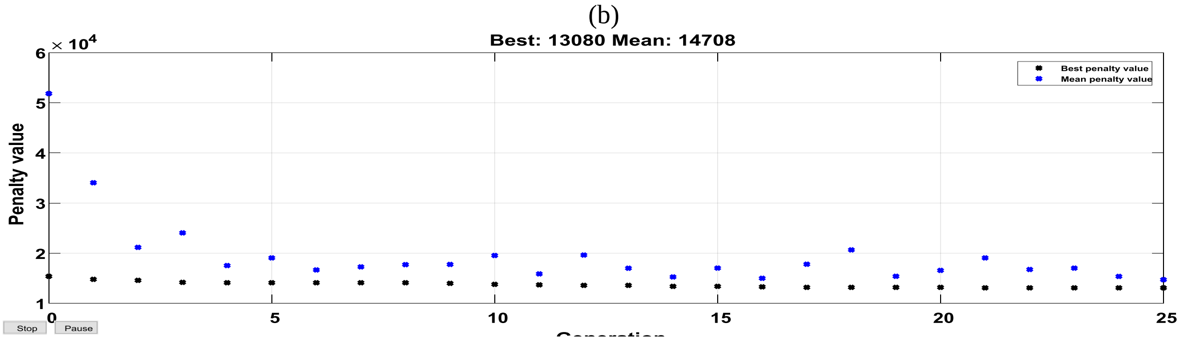
The genetic algorithm sets these variables as it runs multiple simulations of the model via the variable Resource Capacity. The starting values of resource capacities are shown below:



(a)



(b)



(c)

Figure no. 15. (a-c) Waveform for penalty values using Genetic Algorithm in different conditions

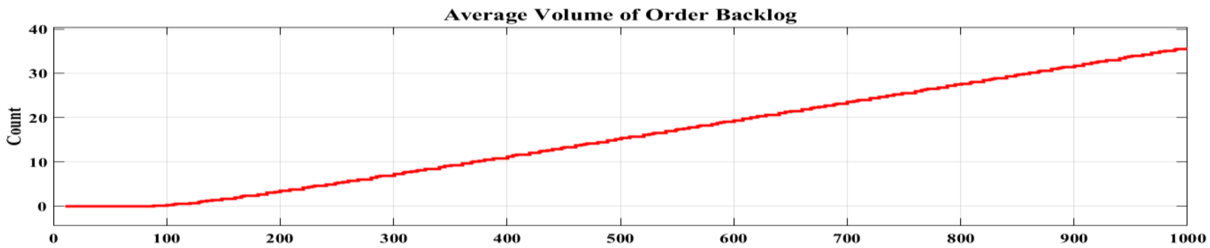
As shown in figure 15. Waveform for penalty values using Genetic Algorithm in different conditions. The variation in mean and best values of penalty values is shown in below table.

Table no.2 Variation in penalty values in different conditions

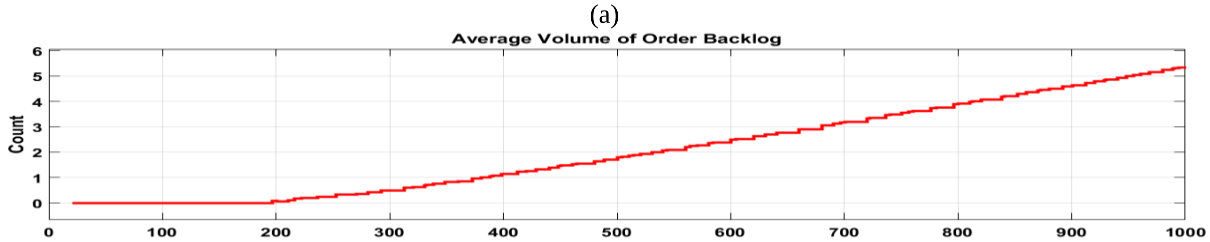
CONDITION	BEST	MEAN
1	14600	137209
2	13600	14477
3	13080	14708

Table no. 3 . Resource Capacity optimization through GA

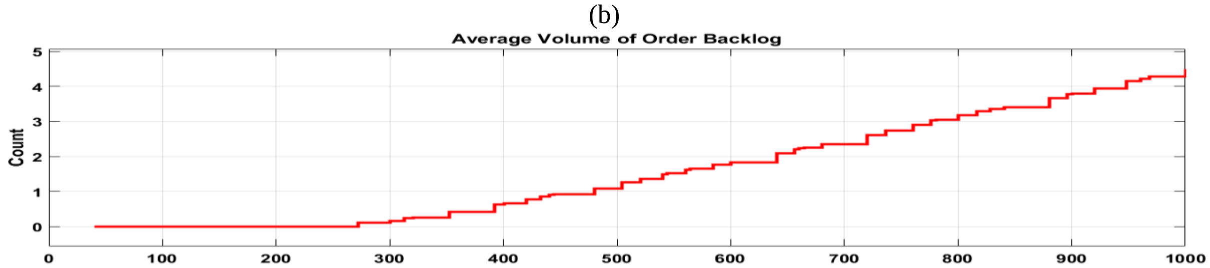
S.NO	no. of workers	Resource Capacity before optimization				Resource Capacity after optimization				Elapsed time (seconds)
1	12	2	2	2	2	13	2	4	2	47.712335
2	2	4	4	4	4	13	1	1	1	252.864797
3	2	8	8	8	8	12	1	1	1	179.006573



(a)



(b)



(c)

Figure no.16. (a-c) Waveform for Average number of orders in backlog in different conditions

As shown in figure 16. The most illustrative result here is the first one, Average number of orders in backlog, which represents the wait time for orders as the system struggles to keep up with inflow.

IV. Conclusion

This chapter has introduced the traditional and intelligent optimization methods and has given GA algorithms. The basic concept, principles rationales have been discussed. In the development of the application of intelligent optimization algorithms, there are existing contradictions between the optimization results and computational time due to the computational speed and time constraints.

Hence, it is difficult to guarantee the computational results for the global optimum and the optimization effect is not very ideal. It is effective to avoid the local optimal solution, to speed up the convergence and to obtain the better global search ability, etc. Other results of the system include Average number of batches waiting for water, Average number of batches waiting for heat, Average number of batches waiting for draining, Utilization of batch reactors, Utilization of water supply, Utilization of heat supply, Utilization of draining facility, Throughput of type A batch and Throughput of type B batch..

GA is implemented to improve the total production time as for 12 workers it consumes 47.71 sec, for 2 workers its 252.86 sec and for another system it consumes 179 sec.

References

[1.] Z. Zou, Z. Wang, L. Meng, M. Yu, D. Zhao and N. Guo, "Modelling and advanced control of a binary batch distillation pilot plant," *2017 Chinese Automation Congress (CAC)*, Jinan, 2017, pp. 2836-2841.

[2.] J. Q. Wang, F. Y. Yan, P. H. Cui and T. Xia, "Performance evaluation of bernoulli serial lines with batching machines," *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, Xi'an, 2017, pp. 295-295.

[3.] E. C. Balta, K. Jain, Y. Lin, D. Tilbury, K. Barton and Z. M. Mao, "Production as a service: A centralized framework for small batch manufacturing," *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, Xi'an, 2017, pp. 382-389.

[4.] A.Depalo, M. Barolo, F. Bezzo and R. Muradore, "Phase identification for product quality prediction in batch processes: Application to industrial resin production," *2015 European Control Conference (ECC)*, Linz, 2015, pp. 3496-3501.

[5.] K. Gonzalez *et al.*, "Modeling and parameter identification of the batch lactic acid production process from wheat flour," *2013 17th International Conference on System Theory, Control and Computing (ICSTCC)*, Sinaia, 2013, pp. 238-243.

[6.] Carlos A. Méndez, Jaime Cerdá, Ignacio E. Grossmann, IiroHarjunkoski, Marco Fahl, State-of-the-art review of optimization methods for short-term scheduling of batch processes, Computers & Chemical Engineering, Volume 30, Issues 6–7, 2006,Pages 913-946,ISSN 0098-1354.

[7.] B. Sharda and S. Bury, "Evaluating the impact of batch degradation and maintenance policies on the production capacity of a batch production process," *Proceedings of the Winter Simulation Conference 2014*, Savanah, GA, 2014, pp. 2261-2268.

[8.] Z. Shi, P. Liu, H. Gao and L. Shi, "Production planning for a class of batch processing problem," *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, Gothenburg, 2015, pp. 1188-1193.

[9.] S. Geethalakshmi, S. Narendran, S. Ramalingam and N. Pappa, "Optimization of Fed-Batch Process for Recombinant Protein Production in Escherichia coli Using Genetic Algorithm," *2011 International Conference on Process Automation, Control and Computing*, Coimbatore, 2011, pp. 1-5.

[10.] R. De, S. Bhartiya and Y. Shastri, "Dynamic optimization of a batch transesterification process for biodiesel production," *2016 Indian Control Conference (ICC)*, Hyderabad, 2016, pp. 117-122.

[11.] Y. Ma, T. Li and Y. B. Gianchandani, "A Low-Temperature Batch Mode Packaging Process for Submillimeter Microsystems in Harsh Environment Applications," in *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 6, no. 3, pp. 366-372, March 2016.

[12.] Xu Jie, "A manufacturing process management platform for single-piece and small-batch production," *MSIE 2011*, Harbin, 2011, pp. 1107-1111.

[13.] K. E. Vrontakis, A. N. Kampianakis and G. J. Tsinarakis, "A Petri net based methodology for modelling, analysis, demand forecast and optimal planning of batch production systems," *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, 2016, pp. 001136-001141.

[14.] J. Yao, H. Wu and Y. Zhou, "A method for inhibition of product reliability degradation based on manufacturing process quality variation control," *2016 Prognostics and System Health Management Conference (PHM-Chengdu)*, Chengdu, 2016, pp. 1-6.

Pushpendra Singh Chahar. "Optimizing Shared Resources in a Batch Production Process using Genetic Algorithm for Small Production Industry." IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE) , vol. 15, no. 6, 2018, pp. 26-38.