Continuous Process Improvement Implementation Framework Using Multi-Objective Genetic Algorithms and Discrete Event Simulation

Parminder Singh Kang, Rajbir Singh Bhatti

NOTICE: This is the peer reviewed version of the following article: Kang, P. S. and Bhatti, R. S. (2019), "Continuous process improvement implementation framework using multi-objective genetic algorithms and discrete event simulation", Business Process Management Journal, Vol. 25 No. 5, pp. 1020-1039, which has been published in final form at http://dx.doi.org/10.1108/BPMJ-07-2017-0188.

Continuous Process Improvement Implementation Framework Using Multi-Objective Genetic Algorithms and Discrete Event Simulation

Abstract

Purpose – Continuous process improvement is a hard problem, especially in high variety/low volume environments due to the complex interrelationships between processes. This paper addresses the process improvement issues by investigating the job sequencing and buffer size optimization problems simultaneously.

Design/methodology/approach – This paper proposes a continuous process improvement implementation framework using a modified genetic algorithm and discrete event simulation to achieve multi-objective optimization. The proposed combinatorial optimization module combines the problem of job sequencing and buffer size optimization under a generic process improvement framework, where lead time and total inventory holding cost are used as two combinatorial optimization objectives. The proposed approach uses the discrete event simulation to mimic the manufacturing environment, the constraints imposed by the real environment and the different levels of variability associated with the resources.

Findings – Compared to existing evolutionary algorithm based methods, proposed framework considers the inter-relationship between succeeding and preceding processes and the variability induced by both job sequence and buffer size problems on each other. A computational analysis shows significant improvement by applying proposed framework.

Originality/Value – Significant body of work exists in the area of continuous process improvement, discrete event simulation and genetic algorithms, a little work has been found where genetic algorithms and discrete event simulation are used together to implement continuous process improvement as an iterative approach. Also, a modified genetic algorithm addresses the job sequencing and buffer size optimization problems simultaneously by considering the inter-relationships and the effect of variability due to both on each other.

1. Introduction

Operational problems have been augmented due to increased global competition, scarcity of resources, higher customer expectations (in terms of higher quality, low cost, reduced lead times) and pressure from the government or other regulatory bodies to reduce carbon emissions and to be more efficient in the energy usage. This has kept manufacturing organizations in the quest for continuous process improvement to reduce the waste by optimizing processes at different levels. This becomes even more important in the current high variety/low volume (HV/LV) manufacturing landscape, where customer demands are extremely volatile both in terms of quantity and variety. There are numerous examples of process improvement approaches those have been applied to various manufacturing/service processes and product types, ranging from small parts/components (engines, tires, fabricated components, etc.) to the whole product (aircraft, coach/bus, automotive sector, service processes – hospitals, banking, educational sector and so on). (Alrashed and Kang 2017; Bastian et al. 2016; HM Government 2013; Lage and Godinho Filho, 2016; Yu and Lee, 2018). According to Kang et al. (2013), providing a high variety and customer focused products/services may allow organizations to stay ahead of their competitors. Traditional manufacturing approaches emphasize high production of a single commodity, which is no longer applicable since without having the sufficient variation it does not attract enough customers to increase profitability. On the other hand, HV/LV products escalate manufacturing problems at a higher rate and often problems are more complex in terms of number of variables involved and their interdependencies. Because of this, manufacturing organizations in a wide range of industries face the challenge of providing a high product variety at a very low cost. In fact, a multitude of customizable product options force the manufacturers of these products to deal with a (theoretical) product variety which exceeds several billions of models. For instance, a base model of a car can be modified according to customer requirements such as the addition of a manual or electric sunroof, air conditioning, power windows etc. (Nazarian et al. 2010). Therefore, existing methods and tools are becoming obsolete due to the increased complexity of modern manufacturing systems, where most of the existing tools are not powerful enough to solve modern manufacturing problems effectively and efficiently especially in HV/LV environments. This has amplified the need for new, efficient and effective tools and techniques to cope with these problems. Researchers have used heuristic based methods combined with simulation-based approaches to solve both simple and complex operations problems as one solution won't fit for all. Solutions need to be customized due to varying nature of production systems. For example, use of heuristic algorithms and simulation modelling to solve hybrid flow shop scheduling problem with mixed batch sizes (Yao et al., 2012). Li et al. (2015) develop a heuristic search genetic algorithm (HSGA) for job sequencing aiming at minimizing the makespan and total weighted tardiness. Costa el al. (2013 and 2014) proposed a mixed integer linear programming model combined with dual encoding and smart-decoding based genetic algorithms framework to address the makespan minimization problem for a hybrid flow shop with a parallel batching system. Aboutaleb et al. (2017) used simulation modeling and data mining approach to develop the standalone closed-loop formula for a throughput rate of normally distributed asynchronous human-dependent serial flow lines. Pfeiffer et al. (2016) used a combination of simulation-based approach and statistical learning methods to improve the to develop a multimodel prediction of manufacturing lead times. Despite using various optimization and simulation approaches, there is one commonality that simulation modelling is adopted to understand/visualize the system behavior and optimization method is customized based on the problem complexity. Kang et al (2015) compared the production scheduling problem results for a multi-machine scenario using standard scheduling methods (Forward by Due Date, Forward by Priority, Backward by Due Date, Backward by Priority, APS forward, APS Minimize WIP forward and APS Parallel Loading) with a integrated approach using modified genetic algorithms and simulation. Preactor APS400 (scheduling tool from Preactor International – A Siemens Company) scheduling package was used to model the multi-machine scenario for a wire and cable manufacturing process. Results demonstrate the inability of standard tools to capture the production environment variability and interrelationships between various attributes. In case of multi-objective optimization modified GA outperforms the standard scheduling approached part of APS400 scheduling package.

This paper represents the process improvement issue in a HV/LV manufacturing environment by addressing the job sequencing and buffer size optimization problem simultaneously. The main aim is to develop a continuous improvement implementation framework by considering the effect of the job sequence and buffer size on each other. The proposed framework uses modified genetic algorithms (GAs) for the multi-objective optimization module and discrete event simulation tool (Simul8) to evaluate the performance of solutions and to mimic the manufacturing environment respectively. The objective function is derived from two key organizational objectives; lead time (LT) and total inventory holding cost (TIHC). The concept of Pareto optimality is then used to generate a final set of solutions based on the two objectives. This paper is organized as follows. Firstly, it highlights the process improvement issues, where job sequencing and buffer size optimization problems are exemplified and GA based multi-objective optimization is introduced. Secondly, problem formulation is performed, which includes the simulation model representation, job sequencing and buffer size chromosome representation, system constraints and the proposed approach. Finally, the results discussion illustrates the effectiveness of the proposed approach.

2. Continuous Process Improvement (CPI) Issues

CPI is one of the absolute requirements for organizations to survive in modern competitive and fast-paced business environments. These conditions require tools and techniques that can provide proactive solutions quickly in highly complex and variable environments (Taha et al. 2011; Tasan et al. 2007; Varela et al. 2003; Velumani and Tang 2017). CPI problems are a well-known subclass of combinatorial optimization problems that exist in all areas, such as in the manufacturing, management and service industries. Researchers have addressed process improvement issues by focusing on the different attributes at the operational level, such as scheduling, sequencing, machine layout, grouping, batch size and buffer size (Kaylani and Atieh 2015; Li et al. 2016). Most of the associated problems are NP-hard and are combinatorial in nature, where more than one organizational objective is associated. The only practical approaches are heuristic strategies some of the most commonly used approaches are; State Space Search, Branch and Bound, Tabu Search, Simulated Annealing and GA. There are numerous entities involved in the manufacturing environment and most of these exhibits dynamic, unpredictable and complicated relationships among them. This makes the CPI process more vulnerable to failures as the effect of improving one performance measure (PM) needs to be considered on other PMs before deciding over the solution.

In fact, high levels of variability and the interrelationship between process entities increases complexity, which makes it almost impossible to solve these problems using the traditional tools and techniques. The job sequencing problem and buffer size optimization problems are regarded as NP-Complete i.e. there are no polynomial time algorithms, which can possibly solve these to the best solution. Also, there can be more than one optimal solution, which satisfies the organizational objectives/constraints based on the interdependency between both problems.

2.1. Job Sequencing

Job sequencing determines the sequence in which jobs need to be processed, where a sequence can be defined on the basis of offset priorities and constraints. These priorities and constraints define the order in which jobs are processed either due to limited resources, organizational/operational constraints and objectives. Job sequencing could play a vital role in reducing manufacturing LT and TIHC by reducing the number of changeovers due to product mix. The product mix is one of the main causes of variability due to variable processing time, setup time, quantity and the routings associated with the different products. For instance, according to El-Bouri et al. (2006), the sequence in which jobs have been processed determines the performance of organizations as one sequence may increase manufacturing lead time over another due to variable cycle and setup time associated with different part types. In fact, the job sequence optimization problem is the ordering of different parts on a machine/s, such that the optimal sequence can be obtained for some measure of effectiveness according to selected PMs, where jobs are subjected to constraints imposed on different product types (Xia et al. 2005). Bertrand and Sridharan (2001) and Burdett and Kozan (2000) regard job sequencing as one of the most difficult combinatorial optimization problems since many sequences may exist in a vast search space where objective values may exist near to each other. In addition, an optimal sequence may not provide noticeable improvements because of organizational constraints. However, the optimal job sequence may help decision-makers to determine the due date assignments more accurately by obtaining the optimal lead time, which defines the total manufacturing LT to complete a customer order. According to Veral (2001), one of the main advantages of having an optimal sequence is that knowing manufacturing LT, due dates can be set internally by scheduling software. Internally set due date reflect the constraints imposed due to the variable setup times and processing times, product mix, routings and machine failures. From the current research perspective, the focus of job sequencing remains to decrease the effect of variability due to the variable setup time induced by the product mix, which can further assist in due date assignment and scheduling.

2.2. Buffer Size Optimization

The buffer size optimization problem is addressed here to determine the optimal buffer locations and required buffer size to deal with high levels of variability, also known as buffer management. For instance, the buffer management mechanism was originally used in the Drum-Buffer-Rope (DBR) to improve material flow by reducing the effect of variability. In fact, the primary concern remains to guard the system against expected disruptions (i.e. Variability induced due to customer demand, product mix, processing times, setup times etc.) or/and unexpected (i.e. machine failure) disruptions (Umble and Umble 2006). According to (Umble and Umble 2006; Riezebos et al. 2003), buffer size optimization may assist organizations as:

(1) Decreased material flow complexity by providing the optimal buffer size at optimal locations in order to reduce the effect of variability.
(2) Provides control over LT by maintaining the appropriate buffer sizes in front of the constrained resources. This may assist in achieving maximum utilization of the constrained resource in a highly variable manufacturing environment.
(3) Improved mechanism over the Kanban system as a fixed level of inventory is maintained throughout the system, and the material is pulled by processes as required.

In HV/LV manufacturing environments buffer sizes may be used as one of the solutions to protect constrained resources against variability due to machine failure, setup, customer demand and product routing, which forms one of the objectives of the proposed approach. Also, this can be seen as a part of the process improvement methodology, as it guards the system against potential disruption by providing synchronous flow, which may have a direct impact on the manufacturing LT and TIHC. Optimal buffer sizes need to be determined in order to control the inventory holding cost, as inventory holding cost is derived from the buffer size.

3. Problem Description

One of the main aspects of this paper is to highlight the use of combinatorial optimization and simulation modeling as a tool for process improvement. This may help organizations to reduce/manage the effect of variability, as the proposed approach takes advantage of simulation modeling in order to respond to rapid changes in levels of variability. This section illustrates the problem from both simulation modeling and optimization perspectives.

3.1. Problem Representation Using Simulation Modeling

The method developed within this research has used Discrete Event Simulation (DES), Simul8, as a tool to represent the investigated working areas. Simul8 acts as an iterative tool with the combinatorial optimization module to find the optimum job sequence and buffer sizes by maintaining the given system constraints based on organizational objectives. Simulation modeling also enables the optimization module to quantify and validate the job sequence and buffer size population during the evolution process. There are numerous examples of DES being used to analyze and solve real-world problems. The advantages of using simulation modeling in the process of problem-solving being exemplified in the literature and illustrating DES advantages are beyond the scope of this paper. Readers can refer to (Banks et al. 1996; Banks 1999; Kang et al. 2013; Sandanayake et al. 2008; Taha et al. 2011; Velumani and Tang 2017) for detailed information.

In this research, the simulation model represents a flow line, which consists of "***Five WorkCentre***". The working area has different system constraints, such as routing, processing time, setup time, machine failure, buffer quantities and inventory holding cost associated with each buffer. Triangular distribution is used to represent different levels of variability in the simulation model. Triangular distribution allows simulation models to be represented close to the real manufacturing environment (Khalil et al. 2008). It is important to note that some of the variables are subjected to change as the population evolves, such as buffer quantities and job sequences, due to the fact that both buffer size and job sequence form the chromosome and will evolve as the GA progresses through the different generations. On the other hand, processing time, setup time and machine failures are, according to the limits, defined by the triangular distribution within the simulation model, while inventory holding cost remains the same with respect to each buffer location throughout the evolution process. Associated simulation and modeling element attribute can be given as in Table I and variability within the simulation model are represented based on the triangular distribution.

Note: "***M***" represents the WorkCentre and associated number gives the location of that WorkCentre in the simulation model. For instance, M1 represents WorkCentre1. Simul8 represents an area from engine manufacturing line for one of the collaborators and generic names are used to maintain information confidentiality.

Table I. Simulation Modeling Elements Attributes

3.1.1 Work Type and Associated Quantities

Since job sequence is one of the problems addressed in this research paper, the simulation model includes 10 different work types having different quantities to be produced for each work type. This data will be used to represent the chromosome for the job sequencing. Table II illustrates the work types and their associated quantities with respect to (w.r.t.) the different set of experiments.

Table II. Work Type and Associated Quantities

3.1.2 Routings

Each work type in the simulation model follows specific routes. A route defines the machines to be visited in the given order. Along with this, the simulation model maintains the data for the cycle time and setup time for each work type on a given machine. Table III illustrates the associated routings, cycle time and setup time w.r.t. each work type.

Table III (Routings and Associated Attributes)

It is important to note the working of Simul8 and how to model the environment using the simulation tool is out of the scope of this paper.

3.2. Variable Definition

The notations used to describe the problem are;

- N The total number of generations;
- G The total number of Chromosomes in a generation;
- M The total number of WorkCentre. All jobs might not go on all machines. There are five machines used to represent the selected working area;

$\mathrm{w_i}$ The i^{th} Work Type in a given chromosome.

$\mathrm{q_i}$ The quantity of i^{th} part needs to be produced. It is important to note that $\mathrm{q_i}$ is related to the $\mathrm{w_i}$ and it must hold this relation while different chromosomes are created;

$S_{k,l}$ The k^{th} sequence chromosome in l^{th} generation: $S_{k,l} \in S$; $k \leq G$ and $l \leq N$;

$\mathrm{B_{k,l}}$ The k^{th} buffer chromosome for l^{th} generation; $\forall\, \mathrm{B_{k,l}} \in \mathrm{B}$; $k \leq G$ and $l \leq N$;

$\mathrm{b_{max}}$ The upper limit for $\mathrm{b_{ij}}$;

LT_{ij} Lead Time for i^{th} chromosome in j^{th} generation, where $i \leq G$ and $j \leq N$;

$TIHC_{ij}$ Total inventory holding cost for i^{th} chromosome in j^{th} generation, where $i \leq G$ and $j \leq N$;

3.3 Chromosome Representation

A universal *U* represents the solution space for the current problem, which consists of all the set of chromosomes representing job sequences (*S*) and buffer sizes (*B*). There is no relation between both sets (*S* and *B*) in terms of the elements they contain. However, both sets exhibit an interrelationship between them based on variability induced by customer demand and product attributes.

$$(\mathrm{S} \cup \mathrm{B}) \subseteq \mathrm{U} \tag{1}$$

Therefore, set of possible job sequences, ***S***;

$$\mathrm{S} = \bigcup_{\mathrm{k,l=1}}^{\mathrm{G,N}} \mathrm{S_{k,l}} \text{ and } \mathrm{S_{k,l}} = \bigcup_{\mathrm{i=1}}^{\mathrm{P}} \mathrm{w_i q_i} \tag{2}$$

Where, *P* represents the total number of work types involved. Each work type has associated quantity according to the experimental set (Table II).

Each $S_{k,l}$ should satisfy the two constraints in order to qualify as a valid job sequence;

Constraint 1: for every $S_{k,l}$, the sum of the quantities w.r.t. each part must be equal to the total number of parts to be produced (say Q) (Reference Table II), which can be given as, Equation 2.1;

$$Q = \sum_{i=1}^{P} q_i \quad \forall\, S_{k,l} \in S \tag{2.1}$$

Say $S_{1,1}$ = {1:60, 2:50, 3:30, 4:40, 5:60, 6:50, 7:80, 8:50, 9:60, 10:20}; sum of all the quantities (q_i) should add to 500 (total quantify "Q"). Throughout the evolution process, the total number of units to be produced for the chosen scenario (Q) should remain same.

Constraint 2: When a new $S_{k,l}$is created, the quantity for each work type must be the same regardless of position in the chromosome (Reference Table II). Each work type has a one to one relation with the quantity needed. Work type and associated quantity relationship must be held in every valid chromosome regardless of their gene position within the chromosome. Some work types, however, may have the same quantity;

$$\forall\, S_{k,l} \in S : \{S_{k,l}\} = \{S_{k+i,l+i}\} \tag{2.2}$$

$$\forall\, S_{k,l} \in S : (S_{k,l}) \neq (S_{k+i,l+i}) \tag{2.3}$$

$$\text{where,} \qquad S_{k,l} = \bigcup_{i=1}^{p} w_i q_i$$

Equations 2.2 and 2.3 must hold true in terms of the relation between work type and associated quantity and position of elements within the set of a job sequence respectively. Based on equations 2.1, 2.2 and 2.3 therefore, a valid chromosome must satisfy the following condition:

$$S_{k,l} \in S \text{ iff} \sum_{i=1}^{P} q_i = Q \wedge \{S_{k,l}\} = \{S_{k+i,l+i}\} \wedge (S_{k,l}) \neq (S_{k+i,l+i}) \tag{2.4}$$

Where, Q represents the total number of parts to be produced.

Consider two job sequence chromosomes S1 and S2 for a given generation based on Experiments Set 1 (Table II). Based on Table II, equation 2.2, 2.3 and 2.4 can be illustrated as;

$$S1: S_{k,l} = \{1:60, 2:50, 3:30, 4:40, 5:60, 6:50, 7:80, 8:50, 9:60, 10:20\}$$

$$S2: S_{k+i,l+i} = \{6:50, 3:30, 8:50, 5:60, 2:50, 1:60, 7:80, 4:40, 9:60, 10:20\}$$

(1) Equation 2.2; total number of genes in both chromosomes S1 and S2 are same i.e. 10 and work type and quantity relationship is maintained for each gene. For example, S1-Gene1 (1:60) is S2-Gene6 (1:60), etc.
(2) Equation 2.3; S1 and S2 differ based on the gene positioning within the chromosome. For instance, S1-Gene1 (1:60) and S2-Gene1 (6:50).
(3) Equation 2.4 is only true when both 2.2 and 2.3 are true.

Job sequence chromosome uses a real number representation in order to maintain the relationship between the job type and the associated quantity of parts to be produced. Further, in chromosome representation, ***B*** signifies the set of possible buffer sizes (b_i).

$$B = \bigcup_{k,l=1}^{G,N} B_{k,l} \text{ and } \forall B_{k,l} \in B : B_{k,l} = \{b_1, b_2, \ldots, b_{TB}\}, \quad \text{Where } 0 < b_i \leq b_{max} \tag{3}$$

TB represents total number of buffers in the simulation model (problem)

It is important to note that buffer size for each buffer in a given chromosome for the current generation should be greater than zero and less than or equal to the b_{max}. Equation 3 can be exemplified based on Table I; there are five buffers in the system (Table I). Consider that chromosome $B_1, B_2, \ldots, B_G$ represents the buffer sizes for a given generation. B_1 can be given as, where *TB = 5 and G = 20.*

$$B = \{B_1 = \{2, 3, 5, 5, 8\}, \ldots, B_{20} = \{1, 3, 2, 5, 5\}\}$$

These buffer sizes are represented as a binary format in the optimization module.

$$B_1 = \{00010, 00011, 00101, 00101, 01000\} \text{ – Binary representation}$$

These buffer sizes are decoded back to real numbers while the evaluation process is carried out in the simulation tool, as the simulation tool can only deal with real numbers for batch sizes instead of binary representation.

There are other constraints needed to be obeyed for a proposed approach to work in an effective manner, which are validated through the simulation model. For instance;

(1) Only one job can be processed at a time on one machine. For the next job to be processed on the same machine, it needs to wait for the current operation to be finished. For instance, for the *(i+1)*th gene in the *k*th chromosome in *l*th generation to be processed at the *m*th machine, the process start time should be greater than the process finish time for the *i*th gene for the any chromosome in the *l*th generation.

(2) Operation sequence needs to be followed according to the defined sequence. In the proposed approach, operation sequence is validated through the simulation entity called jobs matrix (Reference Table III).

(3) Routing constraints should be followed i.e. some jobs can be processed on alternative machines, while the other needs to be processed on a specific machine. Each job should follow a specific route (Reference Table III).

3.4. Combinatorial Optimization Objectives and Fitness Function

In the current research two objectives are considered i.e. LT and TIHC. From Table I, cost is calculated based on the associated holding costs with respect to each buffer. Therefore, LT and TIHC w.r.t. each generation can be represented as follows;

LT_{ij} represents the lead time, which is equal to the simulation run time and also defines the criteria to terminate each simulation run.

$TIHC_{ij}$ represents the sum of all the costs associated with the queues over the period.

If the cost associated with a queue per minute is, say, c_k and there are p_k parts in the queue at given instance, therefore, for *M* buffers [one buffer space per work center i.e. M WorkCentre implies that there are M buffers]; the inventory holding cost at a given instance ($\mathrm{CPM_t}$) for all the buffers can be given as;

$$\mathrm{CPM_t} = \sum_{k=1}^{M} (\mathrm{c_k}) * (\mathrm{p_k}) \quad (4)$$

It is important to track the per unit inventory holding for each buffer w.r.t. time due to fact that inventory will vary for each chromosome due to different job sequence followed. $\mathrm{CPM_t}$ is directly obtained from the discrete event simulation model based on per unit per minute holding cost provided in Table I.

To calculate the total inventory holding cost for simulation run, inventory holding cost per instance must be added. This is achieved by adding for all instances throughout the simulation period.

Using Equation 4, $\mathrm{TIHC_{ij}}$ is calculated as;

$$\mathrm{TIHC_{ij}} = \sum_{t=0}^{\mathrm{LT_{ij}}} \mathrm{CPM_t} \text{ , where } 0 \le i \le N \text{ and } 0 \le j \le G \qquad (4.1)$$

From equation 4.1, an increase in $\mathrm{p_k}$ and $\mathrm{LT_{ij}}$ may lead to an increased $\mathrm{TIHC_{ij}}$ even if $\mathrm{c_k}$ is kept constant throughout the optimization process.

The fitness function is derived from the weighted fitness of two objectives. Random weights are generated for each chromosome. Generated weight values varies between 0.1 – 0.9. Fitness for the i^{th} chromosome in j^{th} generation is calculated as given in equation 5;

$$\mathrm{F_{ij}} = \mathrm{w_{ij}} * \mathrm{LT_{ij}} + (1 - \mathrm{w_{ij}}) * \mathrm{TIHC_{ij}} \qquad (5)$$

4. Proposed Approach

The application of GAs to real-world problems has interested many researchers (Costa et al. 2013 and 2014; Dorndorf and Pesch 1995; Guo et al. 2009; Kang et al. 2015; Khouja et al. 1998; Li et al. 2015; Niu et al. 2008; Rossi and Dinni 2007; Varela et al. 2003; Yao et al., 2012) since they seem to offer the ability to cope with the huge search spaces involved in combinatorial optimization problems. The proposed CPI approach combines the GA based combinatorial optimization and simulation modeling by addressing the job sequence and the buffer size optimization problem. As discussed earlier, the given problem is NP-complete i.e. There is no algorithm that can possibly solve the problem completely in polynomial time. There are other evolutionary approaches being used by researchers such as the ant colony optimization method (Rossi and Dinni 2007), particle swarm optimization (Niu et al. 2008), mathematical modeling combined with genetic optimization (Guo et al. 2009), Lagrangian relaxation based GA (Varela et al.2003) etc. There are numerous other modified/combined EA and AI-based approach being used, however, comparison of the proposed approach with the other evolutionary algorithms is beyond the scope of this paper.

The optimization module for proposed approach is based on modified GA, which combines the buffer size optimization problem and job sequencing while organizational objectives remain the same for both problems. For instance, according to Chen (2006), Hou and Hu (2011), Li and Wang (2007), Wang et al. (2007) and Jozefowska and Zimniak (2008) complex modified GAs are more successful and competent than the simple GAs, as modified GAs are more flexible in problem representation, genetic operators and evolution process.

4.1 GA Functionality

The optimization module utilizes crossover, mutation and inversion operators in the evolution process. The selection probability of each operator is as described in Table V. GA functionality can be summarized as;

(1) ***Chromosome;*** The GA optimization process starts with an initial population of solutions. Chromosome representation is one of the vital steps in the GA as it encodes the problem, which can influence the solution quality (Song and Hughes 1999). Each individual in the population represents a solution to the problem, called "Chromosome". The real number and binary representation are utilized to represent the job sequence and buffer size chromosome respectively (Section 3.1).

(2) ***Initialization;*** population in GA terminology represents the collection of chromosomes and set of solutions. Before starting with the optimization process a set of initial population is needed. Generating this initial set is known as the initialization process, which is created randomly in most cases (Konak et al. 2006). Therefore, a random population set (job sequence and buffer size) is created (i.e. G = 20).

(3) ***Parent Selection;*** selection process defines how to choose the individuals in a population to create offspring for the next generation. The selection process can affect the evolution process because (Song and Hughes 1999);

  a. Selection of stronger individuals reduces the diversity, which can halt the evolution process.

b. Whereas, selection of weak individuals will lead towards slow evolution.

In order to overcome these issues, current research has adopted for fitness-proportionate selection scheme by using the concept of roulette wheel.

(4) ***Crossovers;*** after selection process parents are paired for mating. This mating process is known as crossover (Konak et al. 2006). In this research, uniform crossover is used, where multiple crossover points are defined based on a random variable “r” for each individual pair. The main reason to adopt random uniform crossover is to increase the efficiency and solution effectiveness. At the start crossover probability is used as 70%. During the evolution process, crossover probability is obtained dynamically in the following manner;

a. If population is stagnant for 3 consecutive generations, then decrease the crossover probability (by 5%) until solution quality is either improved or reduced.

b. Once diversity is again introduced (solution quality changed) then use the crossover probability as 70%.

(5) ***Mutation;*** mutation is an effective and powerful process that entails random alternation of gene/genes in selected chromosomes, typically with very low probability. The main motive behind mutation is to maintain the diversity within the population for the prevention of premature convergence of an algorithm to false peak and stagnation of evolution process (Hu and Paolo 2007; Kang et al. 2015).

(6) ***Inversion;*** a simple inversion method is used (only one chromosome from the population), where the whole chromosome is inverted. For example, gene “n” becomes gene “1”, gene “n-1” becomes gene “2”, etc. The main idea behind using inversion operation is to maintain the population diversity.

(7) ***Replacement Strategy;*** once the new population has generated, old population needs to be replaced by new generations. Current research has adopted generational replacement with elite strategy. Elitism forces GA to retain some number of individuals, which are copied as such to the next generation without any changes (Tang et al. 2002).

(8) ***Evaluation;*** once the population has been copied to the new generation, it needs to be evaluated again to check the fitness of new solutions, i.e. calculate the

fitness of each chromosome in terms of given objective function, as defined in Equation 5.

4.2 Pareto Optimality

The proposed approach used the concept of Pareto optimality to generate the final set of solutions based on two objectives, which are same as the organizational performance measures i.e. LT and TIHC. In fact, the main aim was to find all the possible trade-off among given objectives, represented by the Pareto optimal set. Researchers have defined that the Pareto optimal solution is generated on the basis of domination rule and Pareto optimality, which can be described as (Jozefowska and Zimniak 2008; Sevausand Dauzere-Peres 2003);

A solution *S1* is said to dominate the solution *S2* if and only if;

(1) The solution *S1* is no worse than *S2* in all objectives and,

(2) The solution *S1* is strictly better than the solution *S2* in at least one of the objectives.

The main motive behind saving optimal solutions from each generation is to provide better decision making. The output of the simulation model and optimization module generates the Pareto front, which consists of a set of optimal job sequences and buffer sizes based on the LT and TIHC (Equation 5).

4.3 GA Implementation

The proposed combinatorial optimization approach used a GA to develop the optimization engine, which is developed in C++ and is integrated with Simul8 (Figure 1). The simulation tool here represents the manufacturing environment and the different levels of variability, such as routing, setup time, product mix, processing time and machine failures. Table IV illustrates the steps undertaken while implementing the proposed process improvement framework.

Figure 1. Proposed Approach – Logical View

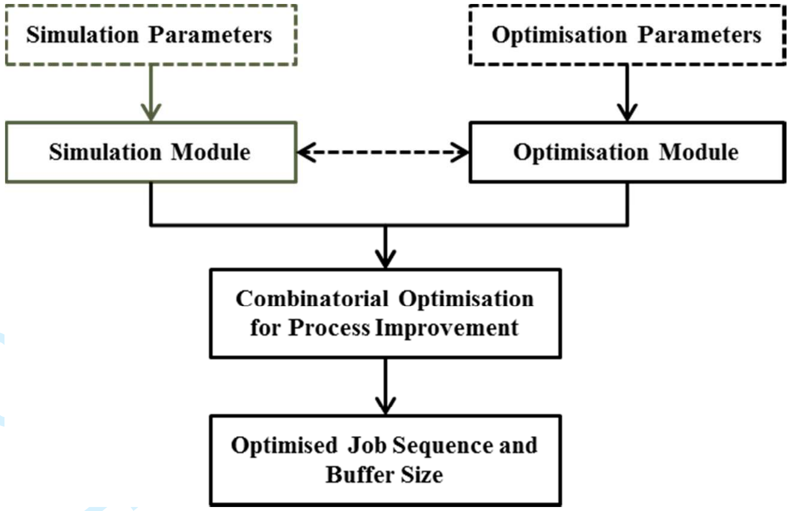


Table IV. Combinatorial Optimization Approach

One of the main advantages of the proposed approach is its applicability to the different manufacturing and service environments. The simulation module provides an opportunity to manipulate parameters or change the simulation model in order to accommodate any alterations in the current problem without changing the optimization module and vice versa. For example, either simulation parameters can be changed, such as processing times, setup time, machine failure, repair time, customer demand, and type of variability etc. or the entire simulation model representing a different scenario. Similarly, the optimization module allows the user to control the optimal parameters, such as population size, number of generations, genetic operator control parameters and selection of optimization objectives.

5. Results and Discussion

5.1 Optimization Module Parameters

Table V, illustrates the parameters used to set the limit for the combinatorial optimization algorithm, which includes the GA and Simulation Model limits.

Table V. GA and Simulation Result Collection Variable Limits

Dynamic crossover and mutation rate are adapted to make sure that the population is not stagnant. This is very important as crossover allows more controlled and justified (best fit) evolution of population from a given point in the solution space. In this research, the mutation rate is dynamically increased once the population becomes stagnant to explore new solutions in the search space in the quest for a better optimal point. To avoid the random walk best solution from each generation is saved and crossover rate is increased again back to 70% once the population is diverse again. This allows each convergence to start with the best solution i.e. peak at a given point in the search space. Job sequence and buffer size population diversity are monitored based on the change in the objective function i.e. either a better or worst solution is found. Based on the explanation provided in Section 4.1 consider the two scenarios;

(1) *Scenario 1*; start of the optimization process, this means
No. of Elites = 2 and No Inverted = 1
Crossover Rate = 70% (No Crossover = 14)
Mutation = (number of solutions) – (No. Elite + No Crossover + No Inverted) i.e. 3

(2) *Scenario 2*; at a given instance of time, says the population was stagnant for 3 consecutive generations, therefore;
No. of Elites = 2 and No Inverted = 1
Crossover Rate = 65% (No Crossover = 13)
Mutation = (number of solutions) – (No. Elite + No Crossover + No Inverted) i.e. 4

5.2 Experimental Results

Table VI exemplifies the results collected based on the different set of experiments. This includes the experimental results before optimization, using OptQuest and after combinatorial optimization (proposed approach). Along with this, the full factorial approach is used to compare the results of experiment 1.1, 1.2, 1.3 and 1.4.

Alongside the process variability presented in the simulation model the following parameters are used to create a different set of experiments;

(1) Total Quantity; three quantities are considered 500, 1000 and 2000 parts in total (Table II), this will allow the effect of variability on a system to be observed based on quantity with respect to the number of part types.
(2) Batch size; three process batch sizes are used with respect to each quantity.
(3) Machine Failure; alongside quantity and batch size experiments are inherited further based on the machine failure present or not.

Based on the given variability (as described above) 36 sets of experiments are generated by combining the total quantity, batch size and machine failure. This allowed testing the GA performance with respect to different levels of variability. Table VI illustrates the set of experiments and shows a significant improvement for both LT and TIHC for all the experiments against the existing system and OptQuest. OptQuest (off-the-shelf tool) provides advanced analysis capabilities by allowing simulation user to search for optimal solutions within your system. OptQuest version 7.0 was used. It doesn't support multi-objective optimization capabilities and combining two problems together. However, it has provided a means to compare the performance of the proposed approach against one of the existing tools.

5.3 Discussion

Using the Table VI, results can be illustrated as;

(1) Experiments with machine failure have a higher impact on TIHC than LT compared with experiments without machine failure, which highlights the need for considering machine failure as one of the constraints within the proposed optimization module. As in current experiments, machine failure is used as one of the variables to generate a different set of experiments.

(2) As product quantity increases the effect of variability decreases on the LT as the proportion between the number of work types and associated quantity decreases, which can be observed by comparing LT values before and after optimization with respect to different total quantities. However, this still signifies the opportunity for process improvement as there is a noticeable difference in total product quantity of 2000 parts.

Table VI. Combinatorial Optimization Results

*Table VI only includes the best solution w.r.t. each PM from the PO set.

(3) On the other hand, TIHC has been reduced drastically throughout all the experiments. Before optimization, job sequence and inadequate buffer size and locations allow excessive inventory to accumulate throughout the flow line, which contributes towards higher TIHC cost. By applying job sequence and buffer size optimization, the optimal sequence can be identified in order to

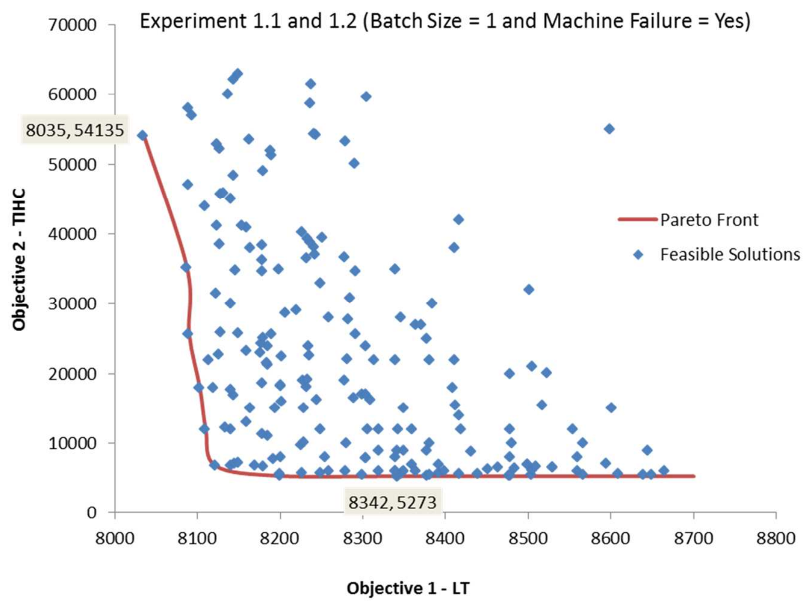
minimize the variability induced due to setups and processing times, as well as optimal buffer sizes, allowing fewer inventories to accumulate respectively.

(4) Experiments were further extended to compare the results with the existing tools (i.e. OptQuest – off-the-shelf optimization tool for Simul8) and full factorial approach. One of the key limitations of OptQuest is the inability to provide the multi-objective optimization. This can be addressed by translating the multiple objectives to one. However, the solution might be biased towards one objective, if fixed weights are used. From Table VI, GA results are closer to OptQuest in most of the cases and in some cases, it has performed better than the OptQuest.

(5) Along with this, the full factorial approach is used to test the GA solution quality. Due to the higher number of experiments full factorial approach was only used to validate the results for batch size 1 with machine failure (i.e. experiments 1.1 and 1.2) and without machine failure (i.e. experiments 1.3 and 1.4). Full factorial needed Simul8 to be linked with Excel Sheet in order to conduct all the experiments. For instance, in this study total number of solutions for batch size 1 with machine failure is 435,456,000 (all possible combinations (10!)*(5!)) for both job sequence (10!) and buffer size (by maintaining the upper limit for buffer size as 5 i.e. 5!)). Similarly, the total number of solutions for batch size 1 without machine failure is 435,456,000. The data collection process took more approximately 32 days (results collection time converted to 24X5, based on the timestamps). The brute-force method is used to get the list of full factorial experiments as it was impossible to generate all the combinations manually. Also, to speed up the experimentation process DES model is executed under four different threads of controls. As expected full factorial performed better than GA. However, GA performance is close to the full factorial in terms of finding an optimal solution. LT and TIHC for full factorial experiments are;

   a. *Experiment 1.1*; LT = 7578 and TIHC = 52,171
   b. *Experiment 1*.2; LT = 7952 and TIHC = 4,887
   c. *Experiment 1*.3; LT = 6400 and TIHC = 22,000
   d. *Experiment 1*.4; LT = 6552 and TIHC = 3,752

Figure 2. Pareto Front – Experiment 1.1 and 1.2



(6) The time taken by the full factorial is significantly higher than the GA (80 – 92 minutes per experiment), which won't be acceptable in real life scenario. On the other hand, OptQuest is much quicker than GA solution (17 – 43 minutes per experiment). This is due to the close integration between the OptQuest and Simul8 as OptQuest.

(7) Figure 2, exemplifies the Pareto front for the Experiment 1.1 and 1.2. It is clear the GA managed to search through most of the search space. As solution evolves, population tends to go towards the given objective functions. One of these solutions can be chosen by the decision maker based on the organizational priorities.

6. Discussion and Conclusion

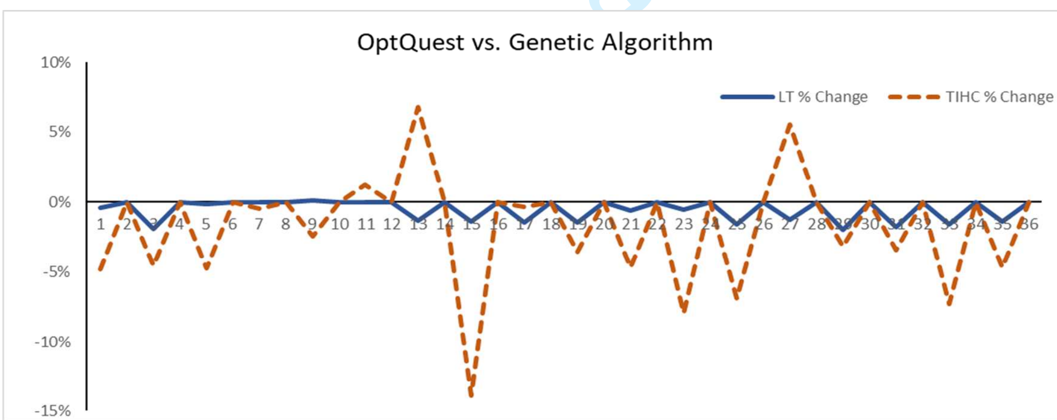
Current research is based on the Lean philosophy derived from the Toyota production system, which defines the scope of this paper by taking forward the concept of CPI. The GA based integrated approach exemplified in this paper is a part of the Lean problem-solving tool developed during the project. One of the aims of the Lean philosophy remains in targeting manufacturing system problems to reduce waste throughout the system. Therefore, the proposed approach combines the job sequence and buffer size problem in order to cope with high levels of internal and external environmental variability as a part of continuous process improvement. In fact, the process improvement can also be related to the improved decision-making process by finding the optimal job sequences and buffer sizes having the minimal LT and TIHC according to the given levels of variability. It becomes even more important when some of the problem variables are not deterministic, which is the best representation of a real-world problem. For instance, machine failure, processing time and set up time are not always constant. These are subject to change according to the conditions and the variability associated with jobs. Results obtained from the proposed approach have shown noticeable improvement based on the selected PMs and based on different sets of experiments. Having used job sequencing and buffer size together as a part of the combinatorial optimization module allows the system to behave as follows;

(1) System acts as a pull system since a job is only released into the system based on the available capacity. This allows taking control over the variable inter-arrival times.
(2) Optimal buffer sizes limit the number of jobs available any time in the system at any instance of time, as a higher number of jobs can lead towards a higher inventory holding cost and a lower number can leave the system without any job for a given instance i.e. Increased lead times. This becomes even more important having a constrained resource in the system, since having an inadequate number of jobs can lead to wasting the capacity at the constrained resource.
(3) On the other hand, constraints associated with different jobs based on the processing time, setup time and quantity affect the lead time and holding cost based on the given job sequence and buffer capacities. It is important to note that the proposed approach considers the effect of change of job sequence and buffer size on each other.
(4) Most importantly, the continuous process improvement process is implemented as an iterative process i.e. changes in the system state can easily be manipulated since the optimization module is integrated with the simulation model.
(5) In terms of comparison, the GA optimization results are closer to the OptQuest and full factorial approach. However, GA optimization provides the advantages over both. For instance, OptQuest (V7.0) doesn't support the combinatorial optimization and both problems cannot be solved simultaneously and time required for full factorial is not acceptable for the real-world problems.
(6) From Industry 4.0 aspect, simulation represents the digital factory. Simulation module provides an opportunity to visualize the system components and allows

analysis without making expensive changes to the real-world. Combining simulation module with optimization component presents the opportunity to conduct what-if analysis before agreeing to a final solution. This means a better control of operations based on the reviewed and optimal scenario based on chosen KPIs. This resembles with the PDCA (Plan-Do-Check-Act), Lean problem-solving methodology. However, it is very important to model real-world accurately by capturing the system components and constraints. Otherwise, the solution developed using above approach may not be valid in real-world.

(7) Comparison of results between modified GA and OptQuest demonstrates that results obtained from the OptQuest and modified GA are very close (Figure 3), however for most of the instances OptQuest outperforms genetic algorithms, this is due to the fact the OptQuest focuses on one objective at a time while genetic algorithms trying to solve the buffer size and job sequencing problem simultaneously. From Lean management and system's thinking perspective modified GA solution is better as it considers the interrelationship between job sequencing and buffer-size problem.

Figure 3. OptQuest Vs. Genetic Algorithms Results Comparison



Current experiments validate the proposed approach based on the different levels of variability stated in section 3. Results from Table VI demonstrate the ability of the proposed approach to deal with different levels of variability. It is important to note that results from the proposed approach are not compared against other AI or mathematical modeling methods because the main focus of this research paper is to demonstrate the capability of the proposed methodology in the context of CPI. The current implementation exemplifies the applicability of the proposed methodology directly for both manufacturing and service industry subjected to the constraints identified in section 3.1. The main applications of the proposed approach are; LT and TIHC cost reduction in highly variable environments (Section 3), internal due date assignment based on optimal LT, improved material flow based on optimal buffer size, reaction to a shifting bottleneck due to changes in process parameters and synchronous flow based on reducing changeovers and optimal buffer sizes, for both the manufacturing and service industries. To take steps forward, the proposed approach will be extended to take into consideration other objectives as required according to organizational requirements and validate it with larger systems as current experiments are based on the five workstations and ten different products. Along with this, further enhancements need to be made by comparing the computational time with other meta-heuristic methods to improve the performance of the algorithm. Most significantly, it is important to note that current optimal values are subject to change, as the level of variability changes, which takes this research further by bringing in the aspect of autonomous decision making along with the optimization process to allow the system to adjust according to changes.

References:

Aboutaleb, A., Kang, P. S., Hamzaoui, R. and Duffy A., “Standalone Closed-Form Formula for the Throughput Rate of Asynchronous Normally Distributed Serial Flow Lines,” *Journal of Manufacturing Systems*, Vol. 33, No. 1, pp. 117 – 128, 2017.

Alrashed, I. A., and Kang, P. S., “Applying Lean Principles to Health Economics Transactional Flow Process to Improve the Healthcare Delivery”, 2017 *IEEE Conference on Industrial Engineering and Engineering Management (IEEM),* 10 – 13 Dec., 2017, Singapore, pp. 897 – 883.

Banks, J., “Introduction to Simulation,” *Proceedings of the 1999 Winter Simulation Conference*, 5th – 8th Dec., 1999, Atlanta, USA, Vol. 1, pp. 7 – 13.

Banks, J., Carson, J.S., and Nelson, B.L., “Discrete-Event System Simulation,” Prentice-Hall Inc., New Jersey, USA, 2nd Edition, 1996.

Bertrand, J. W. M. and Sridharan, V., “A Study of Simple Rules for Subcontracting in Make-to-Order Manufacturing,” *European Journal of Operational Research*, Vol. 128, Issue 3, pp. 548 – 569, 2001.

Burdett, R. L. and Kozan, E., “Evolutionary Algorithms for Flow Shop Sequencing with Non-Unique Jobs,” *International Transactions in Operational Research*, Vol. 7, Issue 1, pp. 401 – 418, 2000.

Chen, Y. P., “Extending the Scalability of Linkage Learning Genetic Algorithms: Theory and Practice,” *Computer Science Research and Tech Reports*¸ University of Illinois, (2006).

Costa, A., Cappadonna, F. A., & Fichera, S., “*A Dual Encoding-Based Metaheuristic Algorithm for Solving a Constrained Hybrid Flow Shop Scheduling Problem,*” Computers & Industrial Engineering, Vol. 64, No.4, pp. 937 – 958, 2013.

Costa, A., Cappadonna, F. A., & Fichera, S., “*A Novel Genetic Algorithm for the Hybrid Flow Shop Scheduling with Parallel Batching and Eligibility Constraints*,” The International Journal of Advanced Manufacturing Technology, Vol. 75, No. 8, pp. 833 – 847, 2014.

Dorndorf, U. and Pesch, E., “Evolution Based Learning in a Job Shop Scheduling Environment,” *Computers & Operations Research*, Vol. 22, pp. 25 – 40, (1995).

El-Bouri, A., Balakrishan, S. and Popplewell, N., “Sequencing Jobs on a Single Machine: A Neural Network Approach,” *European Journal of Operational Research*, Vol. 126, pp. 474 – 490, 2000.

Guo, Z. X., Wong, W. K., Leung, Fan, J. T. and Chan S. F. “Mathematical Model and Genetic Optimisation for the Job Shop Scheduling Problem in a Mixed and Multi-Product Assembly Environment: A Case Study Based in the Apparel Industry,” *Computers & Industrial Engineering*, Vol. 50, Issue 3, pp. 202 – 219, July 2009.

HM Government (2013), Construction 2025, *Report on Industry Strategy: Government and Industry in Partnership*, pp. 1 – 78.

Hou, T. H. and Hu, W. C., “An Integrated MOGA Approach to Determine the Pareto-Optimal Kanban Number and Size for a JIT System,” *Expert Systems with Applications*, Vol. 38, Issue 5, pp.5912 – 5918, 2011.

Hu, X. and Paolo, D. E. A., “An Efficient genetic Algorithm with Uniform Crossover for the Multi-Objective Gate Assignment Problem,” *Proceedings of IEEE Congress on Evolutionary Computing,* Singapore, pp. 55 – 62, 25 – 28 Sept 2007.

Jozefowska, J. and Zimniak, A., “Optimization tool for Short-term Production Planning and Scheduling,” *International Journal of Production Economics*, Vol. 112, pp. 109-120, 2008.

Kang, P. S., Khalil, R. K. and Stockton, D., "Job Sequence Optimisation Using Combinatorial Evolutionary Approach in High Variety/Low Volume Manufacturing Environment," *International Journal of Scientific & Engineering Research*, Vol. 4, Issue. 6, pp. 2145 – 2150, June 2013.

Kang, P. S., Erhart, A., Todeschini, V. and Duffy, A. P., "Comparative Study of Two Scheduling Approaches to Resolve Scheduling Problem for a Wire and Cable Manufacturing Process," *7th Biennial Cable and Wire Technical Conference*, 3rd Nov., 2015, Dusseldorf, Germany, Vol. 1, pp. 136 – 144.

Kaylani, H. and Atieh, A. M., "Simulation Approach to Enhance Production Scheduling Procedures at a Pharmaceutical Company with Large Product Mix," *48th CIRP Conference on Manufacturing Systems – CIRP CMS,* 24 – 26 June, 2015, Naples, Italy, Vol. 41, pp. 411 – 416.

Khalil, R., Stockton, D. J. and Fresco, J. A., "Predicting the Effects of Common Levels of Variability on Flow Processing System," *International Journal of Computer Integrated Manufacturing*, Vol. 23, Issue 3, pp. 325 – 336, 2008.

Khouja, M., Michalewicz, Z. and Wilmot, M., "The Use of Genetic Algorithms to Solve the Economic Lot Size Scheduling Problem," *European Journal of Operational Research,* Vol. 110, pp. 509 – 524, (1998).

Konak, A., Coit, D. W. and Smith, A. E., "Multi-Objective Optimisation Using Genetic Algorithms: A Tutorial," *Journal of Reliability Engineering and System Safety*, Vol. 91, Issue 9, pp. 992 – 1007, 2006.

Lage, M. Jr., and Godinho Filho, M., "Production Planning and Control for Remanufacturing: Exploring Characteristics and Difficulties with Case Studies," *Production Planning & Control*, Vol. 27, No. 3, pp. 212 – 225, 2016.

Li, D., Meng, X., Liang, Q., and Zhao, J., "A Heuristic-Search Genetic Algorithm for Multi-Stage Hybrid Flow Shop Scheduling with Single Processing Machines and Batch Processing Machines," *Journal of Intelligent Manufacturing*, Vol. 26, No. 5, pp. 873 – 890, 2015.

Li, B. and Wang, L., "A Hybrid Quantum-Inspired Genetic Algorithm for Multi-Objective Flow Shop Scheduling," *IEEE Transactions on System, Man and Cybernetics: Part B*, Vol. 37, No. 3, pp. 576 – 591, 2007.

Li, J., Papadopoulos, C. T. and Zhang, L., "Continuous Improvement in Manufacturing and Service Systems," *International Journal of Production Research*, Vol. 54, No. 21, pp. 6281 – 6284, 2016.

Nazarian, E., Ko, J. and Wang, H., "Design of Multi-Product Manufacturing Lines with the Consideration of Product Change Dependent Inter-Task Times, Reduced Changeover and Machine Flexibility," *Journal of Manufacturing Systems*, Vol. 29, No. 1, pp. 35 – 46, 2010.

Niu, Q., Jiao, B. and Gu, X. "Particle Swarm Optimization Combined with Genetic Operators for Job Shop Scheduling Problem with Fuzzy Processing Times," *Applied Mathematics and Computation*, Vol. 205, Issue 1, pp. 148 – 158, Nov. 2008.

Matsuda, M., Sudo, Y. and Kimura, F., "A Multi-Agent Based Construction of Digital Eco-Factory for a Printed-Circuit Assembly Line," *48th CIRP Conference on Manufacturing Systems – Procedia CIRP,* Vol. 41, pp. 218 – 223, 2016.

Pfeiffer, A., Gyulai, D., Kadar, B. and Monostori, L., "Manufacturing Lead Time Estimation with the Combination of Simulation and Statistical Learning Method," *48th CIRP Conference on Manufacturing Systems – Procedia CIRP,* Vol. 41, pp. 75 – 80, 2016.

Riezebos, J., Korte, G. J. and Land, M. J.,"Improving a Practical DBR Buffering Approach Using Workload Control," *International Journal of Production Research*, Vol. 41, No. 4, pp. 699 – 712, 2003.

Rossi, A. and Dini, G., "Flexible Job-Shop Scheduling with Routing Flexibility and Separable Setup Times Using Ant Colony Optimisation Method," *Robotics and Computer-Integrated Manufacturing*, Vol. 23, Issue 5, pp. 503 – 516, Oct. 2007.

Sandanayake, Y. G., Oduoza, C .F. and Proverbs, D .G., "A Systematic Modelling and Simulation Approach for JIT Performance Optimisation," *Robotics and Computer-Integrated Manufacturing*, Vol. 24, Issue 6, pp. 735 – 743, 2008.

Sevaus, M. and Dauzere-Peres, S., "Genetic Algorithms to Minimize the Weighted Number of Late Jobs on a Single Machine," *European Journal of Operational Research*, Vol. 151, pp. 296 – 306, (2003).

Song, Y. and Hughes, J. G., "A Genetic Algorithm with a Machine Order-Based Representation Scheme for a Class of Job Shop Scheduling Problem," *Proceedings of American Control Conference*, Vol. 2, San Diego, CA, USA, pp. 895 – 899, 2 – 4 June 1999.

Taha, Z., Tahriri, F. and Zuhdi, A., "Job Sequencing and Layout Optimisation in Virtual Production Line," *Journal of Quality*, Vol. 18, No. 4, pp. 351 – 374, (2011).

Tang, K. S., Man K. F., Kwong, S. and He, Q, "Genetic Algorithms and Their Applications," *IEEE Signal Processing Magazine*, Vol. 13, Issue 6, pp. 22 – 37, 2002.

Tasan, S. O. and Tunali, S., "A Review of the Current Applications of Genetic Algorithms in Assembly Line Balancing, "*Journal of Intelligent Manufacturing*, Vol. 9, Issue 1, pp. 49 – 69, 2007.

Umble, M. and Umble, E. J., "Utilizing Buffer Management to Improve Performance in a Healthcare Environment," *European Journal of Operational Research*, Vol. 174, Issue 2, pp. 1060 – 1075, 2006.

Varela R., Vela C. R., Puente J. and Gomez A., "Discrete Optimization: A Knowledge-Based Evolutionary Strategy for Scheduling Problems with Bottlenecks," *European Journal of Operational Research*, Vol. 145, Issue 1, pp. 57 – 71, (2003).

Veral, E. A., "Computer Simulation of Due-Date Setting in Multi-Machine Job Shops," *European Journal of Computers & Industrial Engineering*, Vol. 41, No. 1, pp. 77 – 94, 2001.

Wang, Y., Cai, Z., Guo, G. and Zhou, Y., "Multiobjective Optimization and Hybrid Evolutionary Algorithm to Solve Constrained Optimization Problems," *IEEE Transactions on System, Man and Cybernetics: Part B*, Vol. 37, No. 3, pp. 560 – 575, 2007.

Xia, W. and Wu, Z., "An Effective Hybrid Optimisation Approach for Multi-Objective Flexible Job-Shop Scheduling Problems," *Computers & Industrial Engineering*, Vol. 48, No. 1, pp. 409 – 425, 2005.

Yao, F. S., Zhao, M., & Zhang, H., "*Two-stage hybrid flow shop scheduling with dynamic job arrivals,*" Computers & Operations Research, Vol. 39, No. 7, pp. 1701 – 1712, 2012.

Yu, J., and Lee D., "Scheduling Algorithms for Job-Shop-Type Remanufacturing Systems with Component Matching Requirements," *Computers & Industrial Engineering Journal,* Vol. 120, pp. 266 – 278, 2018.

Velumani, S. and Tang, H., "Operations Status and Bottleneck Analysis and Improvement of a Batch Processing Manufacturing Line Using Discrete Event Simulation,"*45th SME North American Manufacturing Research Conference – NAMRC 45*, Vol. 10, NY, USA, pp. 100 – 111, 2017.

List of Tables

Table I. Simulation Modeling Elements Attributes

Modeling Elements	Type	Attribute	Value
Queue for M1 Queue for M2 Queue for M3 Queue for M4 Queue for M5	Queue	Capacity (Number)	Infinite; before optimization no restriction has been imposed on queue sizes. However, queue sizes are derived during the optimization process by considering the system constraints such as batch size. However, users can have initial queue capacities if required because of the model change.
M1 M2 M3 M4 M5	WorkCentre	Cycle Time (Min)	Depends on the Product Type (Table III)
		Setup Time (Min)	Depends on the Product Type (Table III)
		Batch Sizes	1, 5, 10
Queue for M1	Queue	Holding Cost Inventory	£ 0.2 per unit per minute
Queue for M2			£ 0.5 per unit per minute
Queue for M3			£ 0.5 per unit per minute
Queue for M4			£ 0.2 per unit per minute
Queue for M5			£ 0.2 per unit per minute
M1	WorkCentre	Machine Failure	MTTF (min) = 75,85,95 MTTR (min) = 5,15,25
M2			MTTF (min) = 80,85,90 MTTR (min) = 10,15,20
M3			MTTF (min) = 70,80,90 MTTR (min) = 10,20,30
M4			MTTF (min) = 80,90,100 MTTR (min) = 0,10,20
M5			MTTF (min) = 80,85,90 MTTR (min) = 10,15,20

Table II. Work Type and Associated Quantities

Experimental Set	Work Type: Associated Quantity	Total Quantity
1	1:60, 2:50, 3:30, 4:40, 5:60, 6:50, 7:80, 8:50, 9:60, 10:20	500
2	1:100, 2:200, 3:150, 4:100, 5:100, 6:60, 7:100, 8:40, 9:100, 10:50	1000
3	1:100, 2:250, 3:50, 4:200, 5:100, 6:350, 7:300, 8:250, 9:300, 10:100	2000

Table III (Routings and Associated Attributes)

Work Type	Routing	Attributes (ME = Cycle Time: Setup Time)
1	M1 -> M2 -> M3 -> M4 -> M5 -> Exit	M1 = 5:0, M2 = 8:30, M3 = 2:10, M4 = 3:0, M5 = 5:20, Exit = 0:0
2	M2 -> M3 -> M4 -> Exit	M2 = 10:70, M3 = 5:10, M4 = 5:0, Exit = 0:0
3	M1 -> M2 -> M4 -> M5 -> Exit	M1 = 7:0, M2 = 15:30, M4 = 3:15, M5 = 3:15, Exit = 0:0
4	M1 -> M2 -> M3 -> M4 -> M5 -> Exit	M1 = 8:0, M2 = 30:30, M3 = 4:10, M4 = 5:25, M5 = 3:20, Exit = 0:0
5	M1 -> M2 -> M3 -> M5 ->Exit	M1 = 6:0, M2 = 10:45, M3 = 9:15, M5 = 4:25, Exit = 0:0
6	M1 -> M2 -> M4 -> M5 -> Exit	M1 = 5:0, M2 = 15:45, M4 = 2:0, M5 = 3:20, Exit = 0:0
7	M2 -> M3 -> M4 -> M5 -> Exit	M2 = 15:55, M3 = 3:7, M4 = 2:0, M5 = 2:15, Exit = 0:0
8	M2 -> M3 -> M4 -> M5 -> Exit	M2 = 8:35, M3 = 3:7, M4 = 2:0, M5 = 2:20, Exit = 0:0
9	M1 -> M2 -> M3 -> M4 -> M5 -> Exit	M1 = 5:0, M2 = 12:50, M3 = 3:25 M4 = 4:0, M5 = 5:25, Exit = 0:0
10	M2 -> M3 -> M5 -> Exit	M2 = 2:95, M3 = 8:0, M4 = 2:20, Exit = 0:0

Table IV. Combinatorial Optimization Approach

Start:

1. Create an initial set of the population; i.e. $S_{k,0}$ and $B_{k,0} where, 0 \leq k < G$. An initial population of job sequences $S_{k,0}$ and buffer sizes $B_{k,0}$ should obey the constraints identified in equation 2.2 and 2.3 and 3 respectively.
2. Initiate the genetic parameters i.e. number of elite solutions, crossover rate, mutation rate and inversion rate as given in Table V.
3. Initiate the Pareto Optimal set (say *PO*).
4. Start the simulation model (Simulation model has the parameters as defined in the Table I, II and III)

Loop 1: *Go through the generations until current generation < N*

Loop 2: *Go through the population until current chromosome < G*

1. Call Simulation Module and evaluate the job sequence and buffer size chromosomes from the current population. Use both of the chromosomes in parallel.
2. Save the LT and TIHC for each chromosome. TIHC is calculated based on Equation 4.1.

End Loop2:

1. Generate a set of random weights *W*, in order to calculate the weighted fitness.
2. Sort the current population, according to the weighted fitness.
3. Update Pareto based on the LT and TIHC.

If (Not Last Generation)

1. Apply genetic operators on current population, according to the set rate for genetic parameters
2. Replace Current Generation with New Population

Else

Terminate

End Loop1:

End:

Table V. GA and Simulation Result Collection Variable Limits

Variables	-	**Limits**
N	-	100
G	-	20
P	-	10
Q	-	500, 1000 and 2000
No of Elites	-	Fixed; 2 i.e. one solution per objective
Crossover and Mutation Rate	-	Calculated dynamically as solution emerges. The crossover rate however, is kept significantly higher than the mutation and inversion rate.
No of Inverted solutions	-	Fixed; 1
Simulation Run Time	-	Derived by algorithm as it is kept equal to LT
Warm up period	-	None

Table VI. Combinatorial Optimization Results

Experiment No.	Total Quantity	Batch Size	Machine Failure	Optimization criteria	Before optimization		OptQuest Results		Job Sequence and Buffer Size Optimization (Best Solution from Pareto Front)	
					LT	TIHC	LT	TIHC	LT	TIHC
1.1	500	1	Yes	LT	20,489	2,032,863	8,001	5,052	8,035	54,135
1.2				TI HC					8,342	5,273
2.1			No	LT	16,749	1,562,810	6,841	3,781	6,972	22,861
2.2				TIHC					7,013	3,881
3.1		5	Yes	LT	10,742	1,311,448	8,008	19,887	8,018	22,004
3.2				TIHC					8,357	20,800
4.1			No	LT	9,386	1,159,705	6,841	16,998	6,842	43,007
4.2				TIHC					7,029	17,082
5.1		10	Yes	LT	9,287	1,084,242	8,008	39,522	8,001	116,457
5.2				TIHC					8,125	40,521
6.1			No	LT	7,966	925,438	6,834	33,723	6,834	99,217
6.2				TIHC					6,991	33,304
7.1	1000	1	Yes	LT	29,744	5,849,512	15,903	11,833	16,115	46,474
7.2				TI HC					16,751	10,873
8.1			No	LT	28,246	4,739,098	13,570	7,012	13,761	35,328
8.2				TIHC					14,173	7,887
9.1		5	Yes	LT	20,912	4,530,910	15,903	43,462	16,136	68,694
9.2				TIHC					16,597	42,980
10.1			No	LT	18,756	4,100,013	13,540	34,022	13,739	45,132
10.2				TIHC					14,171	34,743
11.1		10	Yes	LT	18,898	4,290,147	15,903	81,652	16,006	196,339
11.2				TIHC					16,576	84,922
12.1			No	LT	16,396	3,770,432	13,564	63,524	13,639	207,377
12.2				TIHC					14,141	68,227
13.1	2000	1	Yes	LT	85,304	33,980,772	31,832	20,511	32,345	259,211
13.2				TI HC					33,640	21,597
14.1			No	LT	66,167	25,839,806	27,094	16,152	27,446	160,039
14.2				TIHC					27,998	15,045
15.1		5	Yes	LT	41,348	20,455,456	31,846	83,094	32,484	131,383
15.2				TIHC					33,090	84,049
16.1			No	LT	34,195	15,542,509	27,094	66,067	27,577	123,432
16.2				TIHC					27,885	67,161
17.1		10	Yes	LT	37,446	17,800,888	31,838	15,287	32,343	266,318
17.2				TIHC					32,725	161,613
18.1			No	LT	32,491	15,542,509	27,101	66,087	27,488	355,010
18.2				TIHC					14,141	68,227