

A Multi-Objective Optimization Approach Using Genetic Algorithms for Quick Response to Effects of Variability in Flow Manufacturing

¹Riham Khalil

Deputy Director, Centre for Manufacturing organization
De Montfort University
Leicester, United Kingdom

²David Stockton

Director, Centre for Manufacturing
De Montfort University
Leicester, United Kingdom

³Parminder Singh Kang

Research Assistant, Centre for Manufacturing
De Montfort University
Leicester, United Kingdom

⁴Lawrence Manyonge Mukhongo

Research Assistant, Centre for Manufacturing
De Montfort University
Leicester, United Kingdom

Abstract— This paper exemplifies a framework for development of multi-objective genetic algorithm based job sequencing method by taking account of multiple resource constraints. Along this, Theory of Constraints based Drum-Buffer-Rope methodology has been combined with genetic algorithm to exploit the system constraints. This paper introduces the Drum-Buffer-Rope to exploit the system constraints, which may affect the lead times, throughput and higher inventory holding costs. Multi-Objective genetic algorithm is introduced for job sequence optimization to minimize the lead times and total inventory holding cost, which includes problem encoding, chromosome representation, selection, genetic operators and fitness measurements, where Queuing times and Throughput are used as fitness measures. Along this, paper provides a brief comparison of proposed approach with other optimisation approaches. The algorithm generates a sequence to maximize the throughput and minimize the queuing time on bottleneck/Capacity Constraint Resource (CCR). Finally, Results are analysed to show the improvement by using current research framework.

Keywords- Synchronous Manufacturing; Drum-Buffer-Rope; Flow Lines; Multi-Objective Optimisation; Job Sequence.

I. INTRODUCTION

All standard paper components have been specified for three reasons: (1) ease of use when formatting individual papers, (2) automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products, and (3) conformity of style throughout the proceedings. Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided throughout this document and are identified in italic type, within parentheses, following the example.

A. Drum-Buffer-Rope: A TOC based philosophy

The pull production system is evolved as a revolutionary system, which enabled organizations to meet the uncertain customer demands at lower production cost and higher profits. Organizations can produce high variability/low volume

products more efficiently and at lower cost by controlling critical process parameters, such as work-in-progress (WIP), changeover %, buffer sizes etc. For example, according to [1], accurate WIP management can have the huge impact on organizational performance as it can minimize the inventory levels, which can have direct impact on the throughput levels and queuing times.

Achieving and maintaining system efficiency is not a simple task, it needs close monitoring of critical processes, when it involves high product variability and each product has different resource/processing requirements such as setup time, processing time and routings. Otherwise, it can create organization wide devastating effects. For example, inventory level within system cannot be controlled individually; high inventories can appear in front of the bottleneck machine, just like a push system. Along this, bottleneck resource not only accumulates work in front of constrained resource but also it is one of the root cause for other problems such as extended lead times, missing due dates, higher inventory holding cost etc., which contributes towards increased operational cost, decreased profit and customer dissatisfaction. Therefore, bottleneck/Capacity Constraint Resource (CCR) is one of the decisive factors for controlling the production system. This is well emphasized by [1], [2] and [3], “If the bottleneck resource wasted one hour, it will be equivalent to one hour wasted for whole system”. Bottleneck can be seen as a resource with limited capacity to satisfy the demand. Also, there can be more than one bottleneck in a system, but only one bottleneck may be the real constraint. In fact, complexity and randomness involved (such as product dependent setup times, variable processing times, machine failure etc.) in actual system makes it harder to control the bottleneck/CCR.

Researchers have proposed numerous tools to support pull production system and process of continuous improvement (CI). Drum-Buffer-Rope (DBR) is one of the vital tools that are used to maintain the system performance by exploiting the capacity constraint resources (CCR) and bottlenecks. DBR is

based on the theory of constraints (TOC) production methodology originated by Goldartt in 1980. Researchers have defined TOC implementation in five steps ([4], [5] and [6]);

- 1) *System constraint(s) identification.*
- 2) *Exploitation of identified constraint(s).*
- 3) *Subordinate everything else according to above decision.*
- 4) *Elevations of system's constraint(s).*
- 5) *Go to step 1 if any system's constraint is broken.*

DBR approach represents a set of rules to implement the first three steps of TOC. For instance, the constraints can be market demand, plant capacity or material shortages etc. DBR can be used here to improve the organization performance when it is limited by capacity constraint resources (CCR) or bottleneck i.e. identification and exploitation of CCR/bottleneck processes to maximize their utilization ([5] and [7]). CCR/bottleneck utilization limits the organization throughput, it needs to maximize in order to maintain on time deliveries, minimum WIP level and production cost, maximum profit and quality. As lack of material at CCR or underutilization of CCR can have devastating effects on throughput. In summary, the main aim of DBR is to schedule the material flow through the production line to produce according to customer demand by keeping lead time, WIP and production cost as lower as possible [7]. From current research's perspective dynamic market demand can be seen as one of the causes of variability (in terms of processing times, setup times and routings followed by different work types) in the system, as in high variety/low volume manufacturing variable product demand can cause more often setups without an optimal schedule. This can cause larger queues, decreased throughput and bottleneck shifts.

The main focus of DBR theory is to concentrate on bottleneck constraints to achieve the maximum throughput with minimum lead time, operation expenses and inventory. According to [8], DBR system consists of three main elements (TABLE 1);

- 1) *Exploitation of CCR (Drum);* Bottleneck defines throughput of production system i.e. capacity of production system must be set what a bottleneck can handle.
- 2) *Protection of CCR from starving (Time Buffer);* Bottleneck should always have work i.e. buffer of jobs should be maintained to accommodate the upstream process interruption. As time wasted on the bottleneck resource is unrecoverable and can affect throughput of entire system.
- 3) *A material release schedule (Rope);* Bottleneck processing capacity predicts the arrival of jobs. Jobs should be released only after receiving signal from bottleneck.

This provides one of the major benefits, production accordance to customer demand with a minimal manufacturing lead-time, inventory and production cost. Also, DBR provides ability to maintain flow at high variety and low

volume. In Summary, DBR endeavour to achieve the three tasks [8];

- 1) Very reliable due date performance.
- 2) Constraint exploitation.
- 3) Achieving shortest possible response time within imposed limitations by CCR.

B. Multi-Objective Optimization:

Evolutionary computing is a research area within computer science that used for solving combinatorial optimization and complex problems, which they perform base on principles of generic population-based heuristic techniques [9]. Researchers have used various evolutionary optimisation techniques in manufacturing process optimisation; such as [19] has used practical swarm optimisation for flow shop scheduling to minimize the makespan with the limited buffer space. [20] has exemplified the buffer size optimisation using the genetic algorithms in an asynchronous assembly system. The main aim remains to determine the optimal buffer size in order to prevent blocking and waiting for succeeding and proceeding WorkCentre, but proposed method here only considers the single objective i.e. improvement reducing make span might degrade other performance measures. On the other hand, [21] has used variable neighbourhood search approach for flexible job shop scheduling with sequence dependent setup times to minimise the makespan and mean tardiness, where scheduling problem is solved by dividing it into two sub problems i.e. machine selection and sequence assignment. However, proposed algorithm in this research has tested without imposing any constraints on the flow line. Similarly, in the research literature there are other optimisation approaches been used such as ant colony mechanism, chaotic harmony search algorithms, mixed integer goal programming, Makovian analysis, immune algorithms etc. However, most of the approaches are single objective and are not integrated with the simulation model.

Current research has been used genetic algorithms (GA) for optimization process to get the optimal job sequence such that queuing time can be reduced and throughput can be increased. GA's have been applied in wide range of applications. Some of the examples are; Optimization (job shop scheduling), Machine Learning (weather forecasting and prediction of protein structure), Automatic Programming (computer programs evolve for specific task or for other computational structure), Economic Models (development of bidding strategies and emergence of economic markets), Immune System Modelling, Ecological Modelling, Population Genetics Models, Interactions between Evolution and Learning and Social System Models ([9], [10] and [11]). Also, genetic algorithms are always remained as the one of the dominant approach in the optimisation process because the;

- 1) Adoptability and versatility that almost any problem can be described in GA code.
- 2) The uncomplicated nature of underlying GA code, as GA mimics the process of natural evolution.
- 3) Ability to deal with new problems, change in problem definition or change in objective function.

- 4) Multi-objective optimization (MOO) can be achieved effectively than the traditional techniques.

Fundamentally, genetic algorithms (GA) are the computer programs that mimic the process of biological evolution to solve complex problems and to model evolutionary systems. GA the phenomenon of natural adaptation and this mechanism can be used in evolutionary programming. According to theoretical framework of GA is simply to move from one population of chromosomes to other in order to find an optional solution, where the selection of chromosomes is based upon the genetic operators, known as; crossover, mutation and inversion ([12] and [13]). There are various examples where GA's have proved their effectiveness and efficiency to solve the complex computational problems. For example; algorithm to find a protein structure from large number of amino acids and algorithms to find fluctuation in financial markets. Some of the main advantages of GA's can be listed as ([11], [13] and [14]);

- 1) GA provides effective use of parallelism i.e. different possibility can be explored simultaneously by using chromosomes.

- 2) GA as a tool of adaptive programming, where system can maintain its performance level with respect to changing environment.

- 3) GA provides solution for complex computational problems. For example, creating an artificial intelligent (AI) system from simple rules using bottom up approach, where GA can drive further rules from the simple rules.

GA's are different from traditional optimization tools and based on digital imitation of biological evolution, using basic genetic operators *Selection*, *Crossover*, *Mutation* and *Elitism*. The population comprises a group of chromosomes from which candidates can be selected for the solution of a problem. Initially, a population is generated randomly. The fitness values of the all chromosomes are evaluated by calculating the objective function. A particular group of chromosomes (parents) is selected from the population to generate the offspring by the defined genetic operations and the fitness of the offspring is evaluated in a similar fashion to their parents. Current population is then replaced by newly generated offspring, based on a certain replacement strategy. Such a GA cycle is repeated until a desired termination criterion is reached (for example, a predefined number of generations are produced or objective function has been met). If all goes well throughout this process of simulated evolution, the best chromosome in the final population can become a highly evolved solution to the problem ([11], [13], [14] and [15]).

Current research has focused on multi-objective optimization. The main aim here is to find all the possible trade-offs among the multiple objective functions i.e. finding all the Pareto optimal solutions. Pareto optimal solution can be defines on the basis of domination rule. Researches have exemplified the concept of Pareto optimality based on two domination rules. These can be described as in [16];

A solution "S1" is said to be 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 domination concept has been used in current research to determine the better solution by combining the multiple objectives using the weighted sum approach. But non-dominance of objective functions has been maintained by generating variable weights for each chromosome [16] and [17]. Current GA implementation can be exemplified as;

- 1) *Initialization*; Generate an initial random population "P" having "m" chromosomes (strings), where "m" represents the population size, which can be given as;

$$P_i = \{pi_1, pi_2, \dots, pm-1, pm\}, \text{ where } i=1 \text{ and } i < n$$

Where "n" is the number of generations.

- 2) *Evaluation*; Evaluate the fitness of each chromosome "Pi" against the fitness function "F", where "F" is derived by using weighted sum approach using multiple objectives and weights are generated randomly for each chromosome. Update the tentative set of Pareto optimal solutions and replace the current generation with new population.
- 3) *Parent Selection*; It emulates the survival-of-the-fittest mechanism in nature. It is expected that a fitter chromosome receives a higher number of offspring and thus has a higher chance of surviving in the subsequent generation. There are many ways to achieve effective selection; including ranking, tournament, and proportionate schemes. Select the pair of chromosomes/chromosome from the current population to take it further to generate new offspring.
- 4) *Crossover*; A recombination operator is applied to combine subparts of selected pair to produce new offspring. Based on problem complexity, different crossover strategies are proposed, such as single point, multipoint or m-point crossover. However, according to researchers crossover operator to be the determining factor that distinguishes the GA from all other optimization algorithms [10].
- 5) *Mutation*; It introduces the variation into chromosome to prevent segmentation or premature convergence. Mutation is carried out according to the predefined, which randomly alters the value at specific string position/positions [10].
- 6) *Elitism*; Elitist strategy where one chromosome or a few of the best chromosomes are copied into the succeeding generation. The elitist strategy may increase the speed of domination of a population by a super chromosome, but on balance it appears to improve the performance [10].
- 7) *Termination*; Finally, GA can be terminated when stopping condition is satisfied, which can either the population limit "n" has been reached or fitness

function has been satisfied. Otherwise go to step 2 for next iteration.

- 8) The final set of Pareto optimal solutions represents dominated solutions from the each generation and it is up to the decision maker to select a solution according to the selected objectives.

II. SIMULATION OPTIMIZATION USING DBR

Current research has opted three phases approach to identify the CCR/Bottleneck resource in selected simulation model and quantify the DBR and GA based multi-objective optimization model. This can be given as;

A. Simulation Modelling

The simulation model has been established using discrete event simulation software Simul8 (Figure 1). There are total 240 jobs to complete having 5 work types (TABLE 1). The attributes of simulation models are;

- 1) Total simulation time was kept equal to the result collection period i.e. 20000 min and simulation warm-up period is kept as 0.
- 2) Triangular distribution has been used for work entry point to match system closely to real manufacturing environment. Also, inter arrival time has not been changed for different batch sizes.
- 3) Travelling time between workstations and machine failures are kept as zero. Job loading is kept as first-come first-serve (FCFS) dispatching policy at all stations. This enabled the system to work with the sequence generated using genetic algorithm.
- 4) Each work type follows a different route, which is defined in job matrix (product routing). Similarly, processing time and setup time with respect to each job and work station is established using job matrix (TABLE 1).
- 5) Initial buffer sizes are kept as default as set in simulation model, which will allow genetic algorithm to decide the optimal buffer size. This will be implemented at the later stage of research.

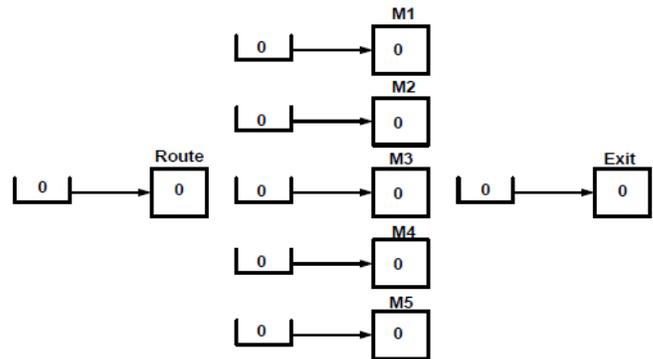
B. CCR/Bottleneck Identification

Bottleneck/CCR has been identified by analysis of data from initial runs. It is important to note the in current experiments the inter-arrival time has been kept constant for all batch sizes. A resource is said to be a bottleneck if [4];

- 1) Had largest pre-processing queues.
- 2) Servicing high capacity requirement jobs.
- 3) Had jobs longest waiting before being processed.
- 4) Possessed longest cycle time.

These four factors represent a simplistic approach; however in real world, high variety/low volume manufacturing bottleneck can be combination of more than one factor. For example, a bottleneck may not necessarily be the slowest or least capacity operation, but it may be result of combination of

more than one factors discussed above or other reasons such as high inter-arrival times, product mix, routings and setups. TABLE 2, shows the bottleneck as machine “M2” based on the maximum queue size with in system at any time “t”, which can related to the physical constraints. The Queue for “M2” is always largest than the all other machines. Along this, this argument can be supported by looking at the capacity requirements from the job matrix. “M2” is having relatively high capacity requirements then other machines. Similarly, one can argue “M2” is bottleneck by considering the logical constraints, such as current policies and procedures involved to process the job. In current scenario, job sequencing can be considered as a logical constraint, for example bad sequencing of jobs effects changeover, which can increase the queuing. Initial results have been collected by using a default sequence generated by program.



[The “route” WorkCentre is there to use the job matrix. The work entry point sets up the “work type” label that is used by job matrix]

Figure 1: Simulation Model

TABLE 1. JOB MATRIX

WORK TYPE	JOB	LOCATION	TIMING	CHANGE OVER
1	1	M1	5	10
1	2	M2	6	30
1	3	M3	2	10
1	4	M4	3	20
1	5	M5	5	10
1	6	EXIT	0	0
2	1	M1	5	20
2	2	M3	3	10
2	4	EXIT	0	0
3	1	M1	5	20
3	2	M2	5	30
3	3	M3	4	10
3	4	EXIT	0	0
4	1	M2	7	30
4	2	M3	2	15
4	3	M4	3	20
4	4	EXIT	0	0
5	1	M2	8	30
5	2	M4	3	10
5	3	M5	4	10
5	4	EXIT	0	0

C. Model Optimization

Once the bottleneck/CCR has been identified, the next phase is to improve the system and to make it work near the ideal state (Figure 2). An improved genetic algorithm has been

proposed to generate an optimal sequence by using two objective functions i.e. maximizing the throughput and minimizing the queuing length. After running experiments a significant improvement has been shown.

The five primary components of the genetic algorithm used here are;

- 1) A chromosomal representation of solutions to the problem i.e. keeping track of job sequence with respect to the work type. The most important point to note here is that chromosome should not lose its integrity in terms of number of jobs encoded when genetic operations are performed.
- 2) Genetic operators that change the composition of the chromosomes.
- 3) A method to initialize a population.
- 4) An evaluation function that represents how well the individual solutions function in the environment, called their "fitness".
- 5) The parameters that are required in order to implement the above components, including population size, number of generations that will be allowed, and stopping criteria.

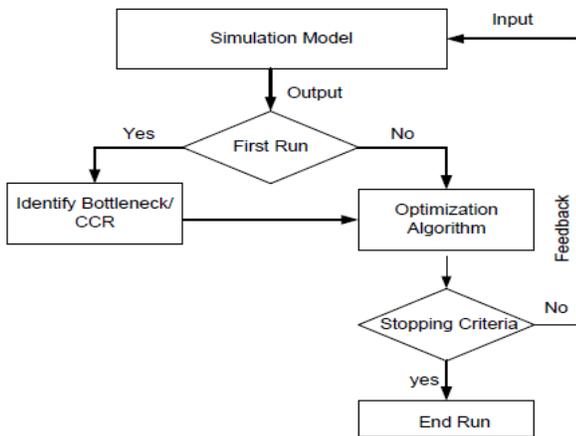


Figure 2. Optimization Model

TABLE 2. BOTTLENECK IDENTIFICATION

EXP. No.	BATCH SIZE	QUEUE FOR					THROUGHPUT
		M1	M2	M3	M4	M5	
1	20	115	130	46	62	17	240
2	15	96	145	35	66	16	240
3	10	79	135	59	9	35	240
4	5	44	119	42	14	14	240
5	2	9	117	13	4	4	169
6	1	2	96	4	2	2	113

TABLE 3. OPTIMIZED RESULTS

EXP. No.	BATCH SIZE	QUEUE FOR					THROUGHPUT
		M1	M2	M3	M4	M5	
1	2	15	69	23	10	9	240
2	2	15	65	30	8	6	234
3	1	4	36	10	7	6	234

III. PRELIMINARY RESULTS, DISCUSSION AND CONCLUSION

From section 2-C, algorithm has shown improvement over the initial results without any optimization. Preliminary results were calculated based on the following parameters TABLE 3;

- 1) Batch size was kept 1 and 2 during the optimization process.
- 2) Genetic parameters; population size = 30, Number of generations = 100, Simulation time = 20000 min, No of elite solutions = 2 and crossover, mutation rates are calculated dynamically as solution emerges.

TABLE 3, shows the preliminary results collected after the optimization process. The results analysis has been shown in the Figure 3. Queue for M2, which was identified bottleneck initially was reduced from 117 to 69 and 65 having throughput 240 and 234 respectively. Similarly, for batch size 1 queue size for M2 has been reduced from 96 to 36 and throughput has been improved from 113 units to 168 units.

In current research a framework has been proposed using drum-buffer-rope and genetic algorithms to achieve identify bottlenecks/CCR, to decide on the optimal buffer sizes and to increase the throughput. It shows an improvement in throughput and noticeable decline in queue sizes. Along this, work is more equally distributed within system (TABLE 3), which will improve system utilization and efficiency. In current implementation, inter-arrival time was kept constant to collect preliminary results. However, algorithm will be improved by adding the facility to deal with the inter-arrival time variations.

Further, improvement will be made on the performance of algorithm and to bring in adaptive inter-arrival times to match exactly with DBR system, as well as the most important factor to determine the optimal buffer sizes and batch sizes to enhance the performance further. As, selecting the appropriate buffer sizes for the flow manufacturing system is a complex task that must account for the random fluctuation in the production rates by individual WorkCentre. If buffer sizes are too large, it can lead to the excessive processing delays and more in-process inventories. On the other hand, if buffer sizes are too small, then small processing delays are small but the small buffer sizes may block upstream WorkCentre from releasing the work.

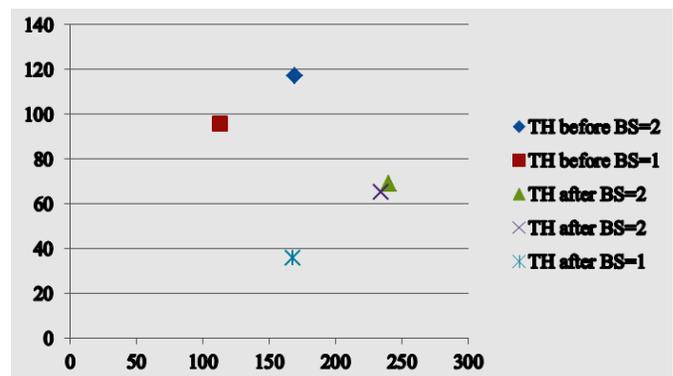


Figure 3. Results Analysis

ACKNOWLEDGMENT

The authors would like to thank TSB – K1532G and De Montfort University for funding the research paper.

REFERENCES

- [1] Y. Wang, C. Jun and L. Kong, "Hybrid Kanban/Conwip Control System Simulation and Optimisation Based on Theory of Constraints", IEEE International Conference on Intelligent Computing and Intelligent Systems, November 2009, pp. 666-670.
- [2] C. Chen and C. Chen, "A Bottleneck-based Heuristic for Minimizing Makespan in a Flexible Flow Line With Unrelated Parallel Machines", Journal of Computer & Operational Research, 2009, Vol. 36, pp. 3073-3081.
- [3] J.D. Blocher, D. Chhajed and M. Leung, "Customer Order Scheduling in a General Job Shop Environment", Journal of Decision Sciences, 1994, Vol. 29, Iss. 4, pp. 951-980.
- [4] M. Umble, E. Umble and S. Murakami, "Implementing Theory of Constraints in a Traditional Japanese Manufacturing Environment: The Case Study of Hitachi Tool Engineering", International Journal of Production Research, 2006, Vol. 44, Iss. 10, pp. 1863-1880.
- [5] C. Y. Huang, C. P. Chen, R. K. Li and C. H. Tsai, "Applying Theory of Constraint on Logistic Management in Large Scale Construction Sites: A Case Study of Steel Bar in TFT-LCD Factory Build-up", The Asian Journal of Quality, 2008, Vol. 9, Iss. 1, pp. 68 – 93.
- [6] K. V. Watson, J.H. Blackstone and S. C. Gardiner, "The Evolution of a management Philosophy: The Theory of Constraints", Journal of Operations Management, 2007, Vol. 25, Iss. 2, pp. 387-402.
- [7] H. Wu, C. Chen, C. Tsai and C. Yang, "Simulation and Scheduling Implementation Study of TFT-LCD Cell Plants Using Drum-Buffer-Rope System", Journal of Expert System with Applications, 2010, Vol. 37, pp. 8127-8133.
- [8] J. M. Nicholas, Competitive Manufacturing Management, McGraw-Hill, 1998.
- [9] A. Konak, W. C. David and A. E. Smith, "Multi-objective Optimization using Genetic Algorithms: A Tutorial", Journal of Reliability Engineering and System Safety, 2006, Vol. 91, pp. 992 – 1007.
- [10] K. S. Tang, K. F. Man, S. Kwong and O. He, "Genetic Algorithms and Their Applications", IEEE Signal Processing Magazine, 1996, Vol. 13, Iss. 6, pp. 1053-5888.
- [11] A. Norozi, M. K. A. Ariffin and N. Ismail, "Application of Intelligence Based Genetic Algorithm for Job Sequencing Problem on Parallel Mixed-Model Assembly Line", American Journal of Engineering and Applied Sciences, 2010, Vol. 3, Iss. 1, pp. 15-24.
- [12] T. Murata, H. Ishibuchi and H. Tanaka, "Genetic Algorithm for Flowshop Scheduling Problems", Journal of Computer Industrial Engineering, 1996, Vol. 30, No. 4, pp. 1061-1071.
- [13] T. Pasupathy, C. Rajendran and R. K. Suresh, "A Multi-Objective Genetic Algorithm for Scheduling in Flow Shops to Minimize the Makespan and Total Flow Time for Jobs", Industrial Journal of Advanced Manufacturing Technology, 2006, Vol. 27, pp. 804-815.
- [14] P. Pongcharoen, C. Hicks and P.M. Braiden, "The Development of Genetic Algorithms for the Finite Capacity Scheduling of Complex Products with Multiple Levels of Product Structure", European Journal of Operational Research, 2004, Vol. 152, pp. 215-225.
- [15] D.E. Goldberg, "Genetic Algorithm in Search, Optimization, and Machine Learning" Addison Wesley Publishing Company, 1989.
- [16] T. H. Hou (Tony) and W. C. Hu, "An Integrated MOGA Approach to Determine the Pareto-Optimal Kanban Number and Size for a JIT System", Journal of Expert Systems with Applications, 2010, In Press.
- [17] J. Jozefowska and A. Zimniak, "Optimization tool for Short-term Production Planning and Scheduling", International Journal of Production Economics, 2008, Vol. 112, pp. 109-120.
- [18] T. H. Hou (Tony) and W. C. Hu, "An Integrated MOGA Approach to Determine the Pareto-Optimal Kanban Number and Size for a JIT System", Journal of Expert Systems with Applications, 2010.
- [19] An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers, Computers & Operations Research, Volume 35, Issue 9, September 2008, Pages 2791-2806, Bo Liu, Ling Wang, Yi-Hui Jin
- [20] Buffer size optimization in asynchronous assembly systems using genetic algorithms, CoTime Series Gene Expression Prediction using
- [21] Neural Networks with Hidden Layers Computers & Industrial Engineering, Volume 28, Issue 2, April 1995, Pages 309-322, A.A. Bulgak, P.D. Diwan, B. Inozu
- [22] Bi-criteria flexible job-shop scheduling with sequence-dependent setup times—Variable neighborhood search approach, Journal of Manufacturing Systems, Volume 30, Issue 1, January 2011, Pages 8-15, A. Bagheri, M. Zandieh

AUTHORS PROFILE

Riham **Khalil** is a Authors Reader in Manufacturing Science and Deputy Director of Centre for Manufacturing in the Department of Engineering, Faculty of Technology, De Montfort University, United Kingdom. Email: rkhalil@dmu.ac.uk

David **Stockton** is the Head of Commercial Development and Director of Centre for Manufacturing, Professor of Manufacturing Systems Engineering in the Department of Engineering, Faculty of Technology, De Montfort University, United Kingdom. Email: stockton@dmu.ac.uk

Parminder Sing **Kang** is a Research Assistant at the Centre for Manufacturing, Department of Engineering, De Montfort University, United Kingdom. Email: pkang@dmu.ac.uk

Lawrence Mukhongo **Manyonge** is a Research Assistant at the Centre for Manufacturing, Department of Engineering, De Montfort University, United Kingdom. Email: p06249913@myemail.dmu.ac.uk