

***Multiobjective Meta-Heuristic Product
Scheduling for Multi-Machine Manufacturing
Systems***

Afsahr, Puya and Wang, Hong

2011

MIMS EPrint: **2012.13**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: <http://eprints.maths.manchester.ac.uk/>

And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

Multiobjective Meta-Heuristic Product Scheduling for Multi-Machine Manufacturing Systems

Puya Afshar, Member, IEEE, Hong Wang, Senior Member, IEEE.

Abstract—Flow line are one of the most commonly encountered layouts in manufacturing industries, where several product types (grades) are manufactured using a sequence of sub-systems or machinery with different tasks. With increasing prices of energy and specific customer demands employing effective product scheduling strategies has become essential for manufacturing industries to maintain their business viability. In this paper, a new product scheduling method is proposed for multi-machine, multi-product flow lines. The objective here is to control the production start time for each grade so that the product delivery time errors are minimised. It is also desired to minimise the overall makespan variability caused by non-Gaussian uncertainties formulated by the entropy of the delivery time errors. Therefore, the proposed product scheduling strategy is a nonlinear multi-objective optimisation problem with non-Gaussian uncertainties. To solve this problem, the nonlinear dynamic flow line model is converted to a linear dynamic equivalent using a $(Max, +)$ algebraic approach. Then, a Proportional-Integral (PI) scheduling controller is used to control the production start time for each grade. The scheduling controller coefficients are tuned by a Multi-Objective Differential Evolution (MODE) algorithm. Simulation results show the effectiveness of the proposed technique and a comparison is made between MODE, Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO).

I. INTRODUCTION

Many production lines in manufacturing industries are formed of material flow lines. In these flow lines the raw materials are processed by a sequence of several machines to form the finished product. Often, several product types (i.e. grades) are manufactured based on customer-specific needs and the capacity of machinery. Examples of such production lines can be found in automotive, steel making, and pulp and paper industries. With rising costs of energy and process maintenance, manufacturing industries are increasingly challenged to deploy effective product scheduling strategies so that their makespans (i.e. the total production time associated to a certain product grade) and product delivery times are kept at desired levels. This is particularly important for manufacturing industries to maintain their business viability.

The product scheduling can be defined as planning of the production or the sequence of operations according to which jobs pass through machines and is optimal with respect to certain optimisation criteria [1]. In this sense, the scheduling can be a class of Discrete Event Systems (DES) [2]. A variety of different solutions have been proposed to model the system using DES. Among them, perhaps Petri Nets

have been one the most popular techniques [3]. Also, several attempts have been made to model the manufacturing system behaviour using Markov Decision Processes (MDP) [4]. As for MDP-based approaches, often Reinforcement Learning (RL) methods [5] are chosen for solving the resulting optimisation problem. Furthermore, Branch and Bound algorithms as in [6] and Linear Programming (LP) algorithms as in [7] have been applied to makespan minimisation problems. In addition, variants of meta-heuristic methods have been developed to multi-machine, multi-product manufacturing systems ([8] and references therein).

In all of the above mentioned techniques, the discrete event models used to explain the system behaviour are nonlinear. Therefore, the analysis and optimisation of such models are often very difficult and optimisation algorithms are computationally expensive. In addition, the scheduling controllers found using above techniques do not have a standard structure which makes it difficult to implement in practice. Using the concept of $(Max, +)$ algebra, nonlinear discrete event dynamic systems can be expressed as a linear state system [9]. This way, the nonlinear optimisation problem can be reduced to a closed loop control design. Based on this, a number of product scheduling techniques using Model Predictive Control (MPC) have been proposed [10]. However, MPC-based methods still have a number of control parameters which have to be set by the operators. On the other hand, less attention has been drawn towards the economic aspects of scheduling due to uncertainties caused by raw material variations and operational failures. These (often) unknown and non-symmetrically distributed variations in turn cause random process variations which cause random increase to processing time of each machine and consequently increase to the energy consumption. It has been shown that the entropy minimisation idea [11] can be applied to minimise variations associated with non-Gaussian uncertainties. The minimum entropy control is in fact a generalisation of minimum variance control for non-Gaussian uncertainties [12].

In order to address the gaps mentioned above, this paper presents a new product scheduling approach for multi-product, multi-machine flow shop manufacturing systems considering non-Gaussian processing time variations for each machine. For this purpose, firstly, a $(Max, +)$ model of the flow shop is developed. Then a fixed-structure PI scheduling controller with tunable parameters is designed to control the times when the production of a certain grade is started. The scheduler parameters are then tuned so that the delivery delay (i.e. Integral of Product Time Squared Error, IPTSE) for

The work is funded by the UK Leverhulme Trust (F/00 120/BC). Puya Afshar and Hong Wang are with the Control Systems Centre, School of Electrical and Electronic Engineering, The University of Manchester, M13 9PL, Manchester, The United Kingdom p.afshar@ieee.org

each grade as well process variations (i.e. Entropy of Product Time Error, EPTE) caused by non-Gaussian processing time variations are minimised. Apart from the fact that the mentioned scheduling problem is inherently multiobjective, it has been previously shown that minimum entropy objectives can be non-differentiable, multi-modal and highly nonlinear [13]. Therefore, in order to tune the scheduling controller, a multiobjective differential evolution (MODE) search method is applied. Fig.1 outlines the proposed scheme for the product scheduling problem. As shown in the figure, product

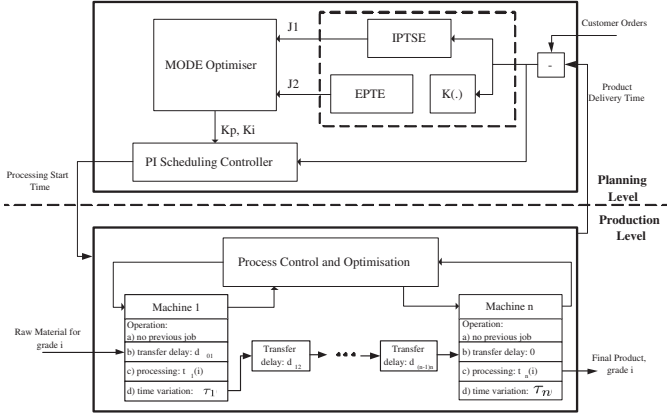


Fig. 1. General scheme of the proposed product scheduling solution

scheduling task is considered a high-level activity compared to factory-level process optimisation and control tasks. It is shown in this paper that this can be formulated as linear ($Max, +$) algebra which facilitates replacing the nonlinear scheduling controllers by fixed-structure controllers such as PI. In the next section, the multi-product, multi-machine scheduling problem is formulated for flow lines.

II. PROBLEM FORMULATION

Consider the production system shown in Fig.1, assuming that m different product grades are required to be produced by n sequentially arranged machines. Each machine can process one grade at a time. The j^{th} machine ($1 < j < n$) start processing the material corresponding to the i^{th} grade ($1 < i < m$) only if it has finished the processing required for the previous job. Also, the j^{th} machine takes $t_j(i)$ time units to process the i^{th} grade. There is a material transfer delay of $d_{j(j+1)}$ between the j^{th} machine and its following machine. In addition, the processing time associated with the i^{th} grade can be increased by $\tau_j(i) > 0$ due to either raw material variations or operational failures in the j^{th} machine. It is assumed that $\tau_j(i)$ are unknown, yet bounded with a known Probability Density Function (PDF). The total product delivery time for the i^{th} grade is noted by $y(i)$ and each product has a delivery deadline of $r(i)$. It is desired to design a fixed-structure scheduler to control the processing start time of each grade (i.e. $u(i)$, the time when the i^{th} grade enters the flow line for processing). The objective of this scheduling controller is to minimise:

- the Integral of Production Time Squared Error (IPTSE),

- the Entropy of the Production Time Error (EPTE).

Define $x_j(i)$ as the time when the j^{th} machine starts processing the i^{th} job. This way, each machine will be assigned a so called 'state'. A simplified diagram of the flow line with non-Gaussian processing time variations is shown in Fig. 2.

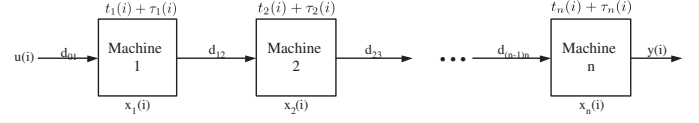


Fig. 2. Flow line with non-Gaussian processing time variations

A. Flow Line System Model Using ($Max, +$) Algebra

Based on the production policy explained, the time when the first machine can start processing the i^{th} job is when:

- it has finished processing the $(i - 1)^{th}$ job; and,
- the raw materials have arrived at its inlet.

Therefore, the state of the first machine can be defined as,

$$x_1(i) = \max \left(x_1(i-1) + t_1(i-1) + \tau_1(i-1), \right. \\ \left. u(i) + d_{01} \right), \quad (1)$$

where $u(i)$ is the time when processing the i^{th} begins. Similarly, the second machine starts processing the i^{th} grade when it has finished its $(i - 1)^{th}$ job and the first machine has finished its i^{th} job. By replacing (1) into resulting state equation, $x_2(i)$ can be obtained as,

$$x_2(i) = \max \left(x_2(i-1) + t_2(i-1) + \tau_2(i-1), \right. \\ \left. x_1(i-1) + t_1(i-1) + t_1(i) + \tau_1(i-1) + \right. \\ \left. \tau_1(i) + d_{12}, u(i) + t_1(i) + \tau_1(i) + d_{01} + d_{12} \right). \quad (2)$$

Applying the same approach, a nonlinear dynamics is obtained for the j^{th} machine. In addition, the product delivery time can be expressed as,

$$y(i) = x_n(i) + t_n(i) + \tau_n(i). \quad (3)$$

Clearly, the resulting state dynamic system is a nonlinear stochastic system. However, using the notations of ($max, +$) algebra, the dynamics can be expressed as a max-plus linear system with stochastic parameters. Defining

$$a \oplus b = \max(a, b), \quad a \otimes b = a + b; \quad (4)$$

the following dynamics can be obtained.

$$x(i) = A \otimes x(i-1) \oplus B \otimes u(i) \\ y(i) = C \otimes x(i) \quad (5)$$

where $x(i) = [x_1(i), x_2(i), \dots, x_n(i)]^T \in \mathbb{R}^n$ and system matrices A and B are expressed as the sum of their deterministic and stochastic parts, as,

$$A = A_d + A_s, \quad B = B_d + B_s. \quad (6)$$

It can be verified that, $A_d(p, q) = A_s(p, q) = 0$, $\forall p < q$, and

$$\begin{aligned} A_d(p, q) &= t_q(i-1) + \sum_{k=q}^{p-1} t_k(i) + d_{k,(k+1)} \\ &= t_q(i-1) \otimes \left[\bigotimes_{k=q}^{p-1} t_k(i) \otimes d_{k,(k+1)} \right], \end{aligned} \quad (7)$$

for $p \geq q$. In addition, it can be shown that $\forall p < q$

$$\begin{aligned} A_s(p, q) &= \tau_q(i-1) + \sum_{k=q}^{p-1} \tau_k(i) \\ &= \tau_q(i-1) \otimes \left[\bigotimes_{k=q}^{p-1} \tau_k(i) \right] \end{aligned} \quad (8)$$

Similarly, it can be shown that,

$$\begin{aligned} B_d(p) &= d_{01} + \sum_{k=1}^{p-1} t_k(i) + d_{k,(k+1)} \\ &= d_{01} \otimes \left[\bigotimes_{k=1}^{p-1} t_k(i) \otimes d_{k,(k+1)} \right] \end{aligned} \quad (9)$$

and

$$B_s(p) = \sum_{i=1}^{p-1} \tau_i(k) = \bigotimes_{i=1}^{p-1} \tau_i(k). \quad (10)$$

Also, the output matrix C can be written as

$$C(q) = \begin{cases} 0, & q \neq n; \\ t_n + \tau_n(k) = t_n \otimes \tau_n(k), & q = n. \end{cases} \quad (11)$$

where $p, q = 1, \dots, n$. The resulting flow line model is max-plus linear, non-Gaussian stochastic dynamic system represented by (5) to (11).

B. Objective Functions

By noting the delivery error time as $e(i) = r(i) - y(i)$; where $r(i)$ is the delivery deadlines, the PI controller dynamics is given by,

$$u(i) = K_p e(i) + K_i \xi(i) \quad (12)$$

$$\xi(i) = \xi(i-1) + T_s e(i), \quad (13)$$

where T_s is sampling time, ξ is the integrator, and $\{K_p, K_i\}$ are the PI controller parameters to be determined. As noted in section I, the first objective is to minimise the delivery time error for all product grades. This is formulated as the IPTSE expressed as

$$J_1 = \sum_{i=1}^m e^2(i) = \sum_{i=1}^m (r(i) - y(i))^2. \quad (14)$$

Furthermore, the second objective is to minimise the process variability caused by the non-Gaussian random processing time changes for each machine by choosing an optimal set of $\{K_p, K_i\}$. As for systems with non-symmetrically distributed noise characteristics, the spread area of the noise distribution cannot be described precisely by only using its

mean and variance. Therefore, entropy is used here to serve as a measure of uncertainty [11]. In this paper, the α -order Renyi's entropy definition is used to formulate the EPTE, as follows,

$$H(e) = \frac{1}{1-\alpha} \log(V_{R\alpha}(e(i))) \quad (15)$$

where $V_{R\alpha}(e)$ is often called the Information Potential (IP), denoted by

$$V_{R\alpha}(e) = \sum_i \gamma^\alpha(e(i)). \quad (16)$$

where $\gamma(e)$ is the PDF of the production time error, which can be estimated through Kernel Density Estimation (KDE) method [14]. Application of KDE yields the following PDF estimation,

$$\gamma(e) \approx \hat{\gamma}(e) = \frac{1}{Nh} \sum_{q=1}^N K_\sigma\left(\frac{e - e_q}{h}\right) \quad (17)$$

where K_σ is a Gaussian Kernel function and h is a bin-width. The choice of the Kernel function and bin-width depends on the required level of smoothness for PDF estimation. This means that the EPTE objective function has the following form,

$$J_2 = \frac{1}{1-\alpha} \log\left(\frac{1}{(Nh)^\alpha} \sum_{p=1}^N \left[\sum_{q=1}^N K_\sigma\left(\frac{e_p - e_q}{h}\right) \right]^\alpha\right). \quad (18)$$

Therefore, the task of scheduling is to find appropriate K_p and K_i parameters (see (12)) so that the following multi-objective cost function is minimised.

$$J(K_p^*, K_i^*) = \min\{J_1, J_2\} \quad (19)$$

where J_1 and J_2 are defined in (14) and (18), respectively. In the next section, a meta-heuristic solution based on multiobjective differential evolution is proposed for solving (19).

III. MULTI-OBJECTIVE DIFFERENTIAL EVOLUTION

Generally speaking, finding the best solutions to multi-objective optimisation problem (19) can be a very difficult problem, specially in the cases where the objectives are nonlinear, non-smooth and non-differentiable, or when it is not possible to determine initial estimates close to the global minimum. In these cases, meta-heuristic optimisation methods can be proven much more efficient as opposed to conventional optimisation techniques. These methods generate a so called *population* of possible solution and update a new population based on probabilistic rules. The aim of such generation updates is to converge to the global minimum of the objective functions with a high probability. The most significant advantage of meta-heuristic techniques is their simplicity as well as ability to optimise complex problems. In this paper, a multiobjective version of the DE algorithm (MODE) is chosen to tune the scheduling controller, as it has consistently been reported to be the most efficient direct search algorithm in several case studies [13].

A. MODE Algorithm

The DE algorithms consist of four main stages, namely, initialisation, mutation, crossover, and selection, which are briefly explained as follows [15]. In order to initialise the MODE, the initial population is chosen randomly to span the initial boundary of decision variables $\vec{x} = [K_p, K_i]$. That is,

$$P_p^{(0)} \leftarrow \vec{x}_p^{(0)} = \vec{x}^L + \vec{\rho}^n \cdot \mathbb{1}_0^1 (\vec{x}^U - \vec{x}^L),$$

$$p = [1, \dots, N_p], \quad (20)$$

where N_p is the total number of populations. Then the mutation takes place by using a difference vector calculated from members of the current populations as follows,

$$\vec{tr}_p^{s+1} = \vec{tr}_p^s + F(\vec{x}_v^k - \vec{x}_w^s), \quad (21)$$

$$\text{for } \vec{\delta} = \{v, w\}. \quad (22)$$

where \vec{tr}_p^{s+1} is the trial vector of the p^{th} member of the $(s+1)^{\text{th}}$ generation, F is the weighting factor and δ_j are randomly chosen integers such that $\delta_l \neq \delta_s \neq p, \forall s, l$. In the crossover stage, a random number is generated for each member of the trial vector. If this number is lower than the crossover rate C_r set by the user, then the gene of the new trial vector is used, if not, the original trial vector $\vec{x}_p^s = (x_{p1}^s, \dots, x_{pN}^s)$ is kept. In other words,

$$tr_{pq}^{s+1} = \begin{cases} x_{pq}^s, & \text{if } \rho \cdot \mathbb{1}_{0,pq}^1 < C_r \\ tr_{pq}^{s+1}, & \text{else} \end{cases}$$

$$q = [1, \dots, N] \quad (23)$$

Finally, to select the next generation members, the resulting trial vector will only replace the original parent if it has a lower objective function value for all objective functions. That is,

$$\vec{x}_p^{s+1} = \begin{cases} \vec{x}_p^s, & \text{if } J(\vec{tr}_p^{s+1}) > J(\vec{x}_p^s) \\ \vec{tr}_p^{s+1}, & \text{else} \end{cases} \quad (24)$$

It must be noted that as the population converges to an optimum, any randomly chosen difference vector becomes smaller in magnitude. Eventually when all members converge to a single solution, the difference vector will be zero and the mutation operator will be disabled all together. Therefore, the actual amount of mutation at each iteration is not only determined by F but also by the population diversity. In the next section, a simulation study is proposed to demonstrate the effectiveness of the method.

IV. SIMULATION RESULTS

Suppose a flow line shown in Fig. 2 comprised of four different machines. For simplicity, assume that the nominal processing time is fixed for all grades. Also, the following nominal processing times and transfer delays are set for the flow line,

$$t = [3 \quad 1 \quad 2 \quad 1], \quad d = [5 \quad 4 \quad 5 \quad 1]$$

Furthermore, due to non-Gaussian stochastic processing time variations, the processing time will be randomly changing. The τ values are not known, however, it is known that they

are a set of uniformly distributed independent additive noises with the following ranges.

$$0 \leq \tau_1(i) \leq 0.3, \quad 0 \leq \tau_2(i) \leq 0.2,$$

and,

$$0 \leq \tau_3(i) \leq 0.1, \quad 0 \leq \tau_4(i) \leq 0.25.$$

Therefore, according to (7), (9), and (11), the following deterministic system matrices can be obtained,

$$A_d = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 13 & 4 & 0 & 0 \\ 18 & 9 & 5 & 0 \\ 25 & 16 & 12 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} 5 \\ 13 \\ 18 \\ 25 \end{bmatrix} \quad C_d = [0 \quad 0 \quad 0 \quad 1]$$

There are 10 orders to be manufactured, and based on the customer orders, the following delivery deadline is also given (measured by sample time);

$$r(i) = [30, 94, 157, 220, 285, 347, 410, 474, 537, 600]$$

A PI controller is designed as the high level scheduler and is supposed to determine the times when each of the grades i must enter the flow line (i.e. when the processing starts). The controller is tuned by a multiobjective differential evolution algorithm. As noted in section II-B, the controller is optimised so that the two objectives IPTSE (14) and EPTE (18) are minimised simultaneously. For this purpose, the objective functions need to be calculated. As for the entropy, a second order Renyi's entropy measure ($\alpha = 2$) with the following Gaussian Kernel is chosen,

$$K_2(e) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{e^2}{2}\right).$$

To solve the multi-objective cost function (19) for K_p and K_i , a multiobjective differential evolution algorithm with population size 500, maximum algorithm iterations are 100 and upper and lower bounds of the controller parameters 1 and 5. Also, the lower and the upper bounds for the range from which the DE scaling parameter is taken are set to -1.5 and +1.5, respectively. Furthermore, the cross-over probability is set to 0.95. An experiment is set to compare a manually-tuned scheduler to the one tuned by the MODE. After 3 generations, the MODE yields the following values for the K_p and K_i ,

$$K_p = 2.6612, \quad K_i = 1.8696$$

A. Optimal Schedule

Fig.3 compares the delivery time performance of the flow line if either manually tuned or MODE-tuned schedulers are chosen. The delivery deadline is indicated with a solid line. As shown in the figure, the scheduler tuned by MODE represent superior performance delivering almost all grades on time. It must be noted that although by using the manually tuned scheduler some grades (4, 9, 10) are finished earlier than the required time, this might incorporate imposing new storage costs to the system which are not desirable. It is also worthwhile to examine the optimal schedule found by the MODE-tuned scheduler. Fig. 4 compares the production start times for each of the grades. As shown in the figure,

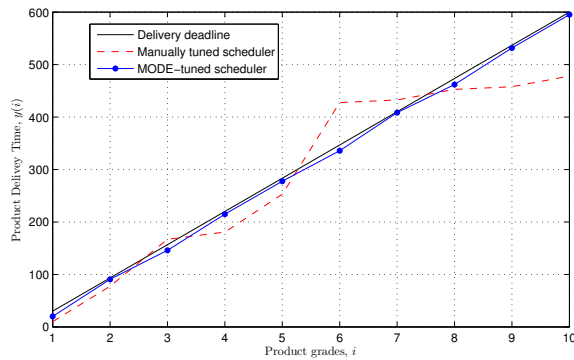


Fig. 3. Production delivery time for different grades

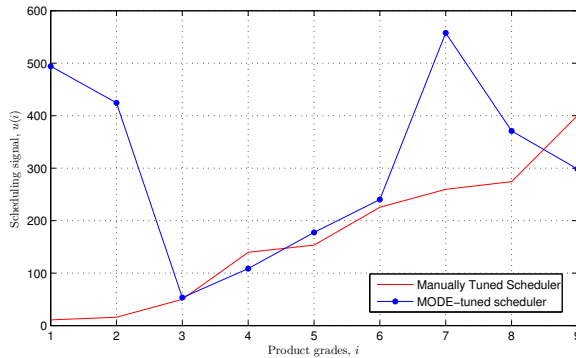


Fig. 4. Scheduling signal found: manual and MODE

the manual scheduler's policy is to start processing of each grade in the same order as they have been registered in the order book, i.e. $[1, 2, \dots, 10]$. However, the MODE-tuned scheduler has found the following optimal processing order,

$$opt_sched = [3, 4, 5, 6, 9, 8, 2, 1, 7, 10].$$

B. Objective Functions

The algorithm chosen must minimise both objective functions simultaneously. At each iteration, MODE finds members of the new generation belonging to the Pareto front. The best member (the point where both objectives are minimum) is found at the end of 100 iterations. As for the flow line of this study, the Pareto front found by MODE is illustrated in Fig. 5. Furthermore, the individual objective functions are examined to verify the effectiveness of the method. Firstly, in Fig. 6, the product delivery error (delay) is shown. Clearly, the scheduler tuned by MODE algorithm shows considerably less IPTSE value. Under the manually tuned scheduler, some grades, specially 6 require significant storage as they have been finished earlier than the required time. Also, to investigate the entropy minimisation, the PDF of the delivery time error, comparing the two schedulers is shown in Fig. 7. A narrowly-shaped PDF has significantly less entropy value compared to a widely distributed one. As shown in the figure, the scheduler tuned by MODE algorithm has efficiently reduced the process variability which confirms

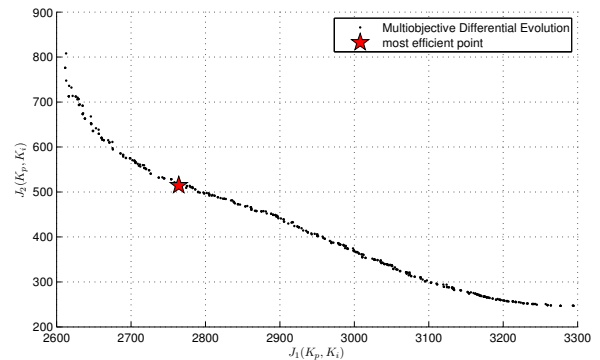


Fig. 5. Pareto front and the solution found by MODE

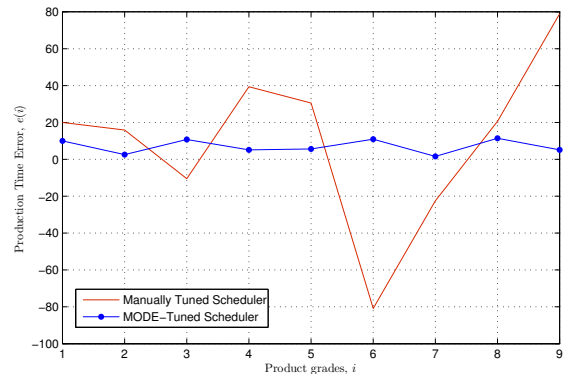


Fig. 6. Production delivery time error manual and automatic scheduling

the effectiveness of the algorithm proposed.

C. Comparison and Analysis

In this section, a comparison is performed to analyse the performance of the proposed optimiser, MODE. For this purpose, the same problem is solved using equivalent multiobjective Genetic Algorithm (GA), and multiobjective Particle Swarm Optimisation (PSO) methods. For all three algorithms, 500 initial populations and a maximum iterations 100 are considered. As for the multiobjective GA, the crossover probability and the mutation probability for each individual are set to 0.25 and 0.1, respectively. Also, for the GA algorithm, the coding (variable representation) is set as binary and 52 bits are used per decision variable. Furthermore, for the PSO algorithm the following parameters are set; social factor ($\eta_1=2$), cooperative factor ($\eta_2=2$), nostalgia factor ($\eta_3=0.5$), inertial constant ($\omega=0.5$), maximum number of neighbours is also set to 5 and a 'star' social network topology is chosen. Fig. 8 shows the resulting Pareto front found by all three algorithms. In terms of convergence speed, the PSO algorithm is relatively quick in the first generation. However, its performance deteriorates rapidly with proceeding to higher generations. The quickest algorithm appears to be MODE with 2.6 msec to evaluate a function. This figure is 3.3 msec for the GA and 2.93 msec for the PSO algorithms. Also, DE required 3 generations

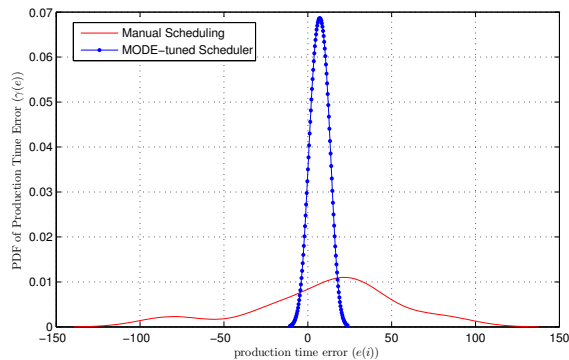


Fig. 7. PDF of delivery time error: Manual and MODE

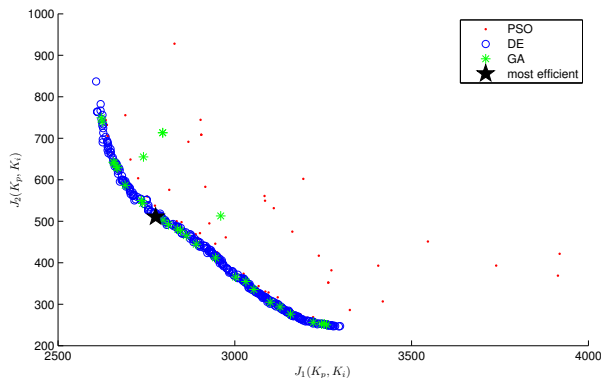


Fig. 8. Pareto front found by GA, DE and PSO

for convergence while this figure was 5 for the GA and 2 for PSO. In terms of convergence performance, the GA and DE show similar solutions, while PSO's convergence was relatively poor.

V. CONCLUSION

In this paper, a meta-heuristic algorithm is proposed for scheduling multi-product, multi-machine flow lines with non-Gaussian processing time variations. Existing scheduling methods incorporate nonlinear dynamic programming techniques which often result in complicated scheduler structures. Also, the effect of random variations in processing time has not been addressed comprehensively. The proposed framework employs a modelling strategy based on the $(max, +)$ algebra which assists to transfer the nonlinear product delivery time model to a $(max, +)$ linear dynamic system. This in turn facilitates the design of fixed-structure controlling schedulers. The proposed approach uses a PI controller which controls the processing time corresponding to each product grade. Therefore, the nonlinear dynamic programming problem is simplified to a closed loop control design problem. In this regard, the task of control is to tune the PI controller coefficients so that two objectives, namely the integral of the production time error, and the entropy of production time error are minimised simultaneously. Since the multiobjective optimisation problem is highly nonlinear,

a direct search algorithm based on differential evolution is used to find the global solution to the scheduling problem. Simulation results suggest that the proposed method can schedule the production such that on-time delivery is realised while the process variability is reduced which is a significant improvement compared to manually tuned scheduler. Future work incorporates studying the stability of the $(max, +)$ model as well as cases where parallel production lines exist.

VI. ACKNOWLEDGEMENTS

This work was jointly supported by the U.K. Leverhulme Trust (F/00 120/BC), Chinese National Fundamental Research Program (No. 2009CB320604), the National Natural Science of China (No. 61074071,60534010), UK EPSRC (EP/I016643/1, EP/E050441/1-CICADA), the Funds for Creative Research Groups of China (No.60821063), and 111 Project (No. B08015).

REFERENCES

- [1] F. Xhafa and A. Abraham, *Metaheuristics for Scheduling in Industrial and Manufacturing Applications*, J. Kacprzyk, Ed. Springer Berlin / Heidelberg, 2008.
- [2] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. Springer Science+Business Media, LLC, 2008.
- [3] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*. Prentice Inc., Englewood Cliffs, NJ., 1981.
- [4] R. Agrawal, "Planning and scheduling problems in manufacturing systems with high degree of resource degradation," Ph.D. dissertation, Georgia Institute of Technology, 2009.
- [5] J. Choi, M. J. Realf, and J. H. Lee, "A q-learning-based method applied to stochastic resource constrained project scheduling with new project arrivals," *Int. J. Robust Nonlinear Control*, vol. 17, no. 13, pp. 1214–1231, 2007.
- [6] P. Baptiste and C. Le Pape, "Scheduling a single machine to minimize a regular objective function under setup constraints," *Discrete Optimization*, vol. 2, no. 1, pp. 83–99, Mar. 2005.
- [7] F.-D. Chou, H.-M. Wang, and T.-Y. Chang, "Algorithms for the single machine total weighted completion time scheduling problem with release times and sequence-dependent setups," *Int J. Advanced Manufacturing Technology*, vol. 43, no. 7, pp. 810–821, 2009.
- [8] P. Pansuwan, "Identifying optimum artificial bee colony (abc) algorithms parameters for scheduling the manufacture and assembly of complex products," in *Computer and Network Technology (ICCNT), 2010 Second International Conference on*, 2010.
- [9] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat, *Synchronization and linearity*. New York: Wiley., 1992.
- [10] L. Tang, S. Jiang, and J. Liu, "Rolling horizon approach for dynamic parallel machine scheduling problem with release times," *Industrial & Engineering Chemistry Research*, vol. 49, no. 1, pp. 381–389, Jan. 2010.
- [11] P. Afshar, H. Wang, and T. Chai, "An ilc-based adaptive control for general stochastic systems with strictly decreasing entropy," *Neural Networks, IEEE Transactions on*, vol. 20, no. 3, pp. 471–482, 2009.
- [12] H. Yue and H. Wang, "Minimum entropy control of closed-loop tracking errors for dynamic stochastic systems," *Automatic Control, IEEE Transactions on*, vol. 48, no. 1, pp. 118–122, 2003.
- [13] P. Afshar, A. Nobakhti, H. Wang, and T. Chai, "Multi-objective minimum entropy controller design for stochastic processes," in *American Control Conference (ACC)*, 2010, pp. 355–360.
- [14] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CR, 1992.
- [15] R. Storn and K. Price, "A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.