Int J Adv Manuf Technol (2018) 99:73–83
DOI 10.1007/s00170-016-9123-1

ORIGINAL ARTICLE

Multi-objective optimization of the order scheduling problem in mail-order pharmacy automation systems

Husam Dauod[2] · Debiao Li[1] · Sang Won Yoon[2] · Krishnaswami Srihari[2]

Received: 24 November 2015 / Accepted: 28 June 2016 / Published online: 2 August 2016

Abstract This paper presents the multi-objective optimization problem (MOP) of minimizing collation delays and the makespan in mail-order pharmacy automation (MOPA) systems. MOPA is a high-throughput make-to-order (MTO) manufacturing system designed to handle thousands of prescription orders every day. Prescription orders are highly customized, and most of them consist of multiple medications that need to be collated before packaging and shipping. The completion time difference between the first and the last medications within the same order is defined as the order collation delay. This research mainly investigates the effects of machine flexibility and the proportion of multi-medication orders on the total collation delays. To solve this NP-hard problem, a genetic algorithm with a min-max Pareto objective function is used. The GA performance is compared to two industry heuristics: the longest processing time (LPT) and the least total workload (LTW). Experimental results indicate that a fully dedicated machine environment has 80 % more total collation delays as compared to a fully flexible machine environment and 25 % more total collation delays as compared to a multi-purpose machine environment. Experimental results also indicate that the GA can achieve the optimal makespan in most cases, while minimizing the total collation delays by 96 % when compared to LPT and LTW.

✉ Debiao Li
debiaoli@fzu.edu.cn

[1] Department of Management Science and Engineering, Fuzhou University, Fuzhou, Fujian 350116, China

[2] State University of New York at Binghamton, Binghamton, New York, USA

Keywords Order scheduling · Orders collation · Multi-objective optimization · Mail order pharmacy automation system · Multi-purpose machines scheduling · Machine flexibility

1 Introduction

As customer expectations for faster delivery and higher customization of products grow, many companies are changing their production strategy from make-to-stock (MTS) to make-to-order (MTO) [19]. MTO is the manufacturing system where all functions within the enterprise, from sales to product delivery, are triggered by the receipt of the customer order [16]. Although MTO systems provide high level of product customization and maintain continuous customer involvement, they are usually characterized by long lead times [6]. Several research streams have addressed product lead time minimization, in particular the manufacturing lead time, from different perspectives to optimize related processes and factors [25]. This research studies the order collation process, which significantly affects the throughput and the lead time of the MTO system.

Orders collation refers to the process where items within the same order are combined before packaging and shipping. Orders are usually associated with multiple items, and these items may be processed on different machines. When the first item is processed, it is usually sent to a collation process or a station where it awaits the remaining jobs from the same order. The design of the collation process or station depends mainly on the manufacturing environment and the material handling system used. The completion time difference between the first and the last item within the same order is defined as the order collation delay. Generally, large collation delays contribute to a reduced system

throughput, and may cause deadlocks to the material handling system. However, considering collation delays as a sole objective may contribute into a larger makespan. Therefore, both objectives must be considered in this scheduling problem.

This type of scheduling problems is formally defined in the literature as the order scheduling problem [10]. An order can be described as a set of jobs that is requested from one customer. Jobs within one order all have to be processed before the order can be packaged and shipped. In this type of scheduling model, the completion time of orders needs to be considered as a performance measure rather than the completion time of jobs [10]. This is related to the fact that higher costs are incurred if the order components are delivered separately. The order scheduling model can be found in many industrial applications other than MTO systems. Examples on applications include assemble-to-order (ATO) systems [23] and auto repair shops [27]. The problem studied in this research is motivated by a mail-order pharmacy automation (MOPA) system, which is a MTO high-throughput manufacturing system designed to handle thousands of prescription orders every day. Prescription orders received are highly customized and most of them contain more than one medication. For a multi-medication prescription order, the first dispensed medication is sent to a waiting station (collation loop), where it waits for the other medications within the same order to be dispensed. There are two main components in MOPA systems. The first component is the auto-dispenser, which is an electro-mechanical device that stores, counts, and dispenses medications. Each dispenser is assigned one type of medication, and cannot be used for other medications due to patient safety concerns. The second component is the robotic dispensing system (RDS) which contains up to 80 dispensers and utilizes a robotic arm to hold the vial while the medication is being dispensed. A high-throughput MOPA system consists of several RDS units that can produce thousands of prescriptions daily. In this research, we have considered the RDS units as parallel machines.

The machine environment in order scheduling models can be fully dedicated, which implies that each machine is only capable of processing one type of job, or flexible, which indicates that machines can process more than one type of job [10]. Flexible machines can either be fully flexible, which means they are capable of processing all types of jobs, or multi-purpose, which implies they are only capable of processing a specific subset of jobs. It is evident that the multi-purpose case lies between the fully flexible and the fully dedicated cases. Each job in this system can only be processed on specific set of eligible machines, which is called the job's processing set [8]. Due to these assignment restrictions, it is expected that this environment's performance will be less efficient than the fully flexible environment in terms of collation delays and the makespan. This research will study the effect of process flexibility on both the makespan and the total collation delays.

Previous research on the order scheduling model in MOPA systems has considered the RDS units to be fully flexible [13–15, 17]. However, because the RDS units are only capable of processing specific medications according to the auto-dispensers assigned to them, this problem becomes a multi-purpose machines scheduling problem. This paper will study the effect of the machine processing flexibility on both the makespan and the total collation delays. A classical optimization approach based on the min-max Pareto solution is used to combine both objectives into a single one, and a genetic algorithm is used to solve it. Results were compared to two heuristics: the longest processing time (LPT) and the least total workload (LTW). The remainder of this paper is organized as follows: Literature review is presented in Section 2; Methodology used to approach this MOP is illustrated in Section 3; Experimental results are discussed in Section 4; followed by conclusions and future work in Section 5.

2 Literature review

The order scheduling problem was first introduced by Julien and Magazine (1990) [7]. This type of problem can be found in many industry applications such as MTO manufacturing companies [7], pharmaceutical companies [7], automotive repair shops [27], and the multi-site order scheduling problems (MSOS) [5]. All these applications are similar in how jobs arrive in the system; however, they can be differentiated by the amount of flexibility their machines have. In this paper, we have expanded the classification found in the literature of the order scheduling problem [10] as shown in Fig. 1. This problem can be classified according to the machines processing flexibility to fully dedicated, where machines are only capable of processing one type of jobs, and flexible, where machines can process more than one type of jobs. Flexible machines can either be fully flexible, which makes them capable of processing all types of jobs, or multi-purpose, which makes them capable of processing a specific subset of jobs. In the real world, where processing restrictions and constraints are involved [18], it is more likely that the machine environment will follow the general multi-purpose case rather than the fully dedicated or the fully flexible cases.

Order scheduling with fully dedicated machines is considered the least complicated, because jobs assignment to machines is more restricted [26]. Problems with such characteristics have received considerable attention in the

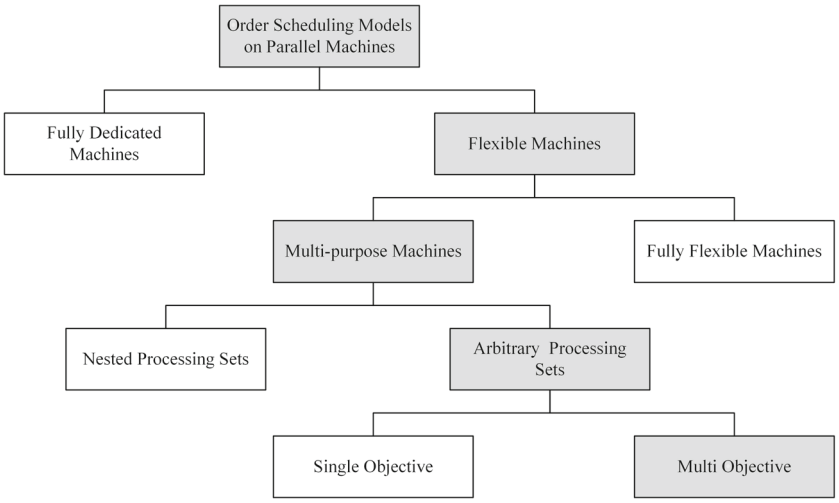

Fig. 1 Classification of the order scheduling problem on parallel machines

literature. It was proven that this problem is NP-hard when minimizing total completion times given that the number of machines $m \geq 3$ [9]. Therefore, several heuristics have been proposed to solve this problem [9, 12, 24]. It was shown that the weighted earliest completion time (WECT) is the most recommended rule for industry use, because it is simple to implement, produces good results, and requires a small amount of memory [12]. For the fully flexible case, it was proven that this problem is NP-hard given that the number of machines $m \geq 2$ [11]. Generally, heuristics integrated with the shortest processing time (SPT) dispatching rule are known to yield better results [2]. For the fully flexible machine environment, weighted shortest total processing time (WSTP)-based heuristics outperformed other heuristics in terms of solution quality and speed [12].

While order scheduling in fully dedicated and fully flexible machine environments has been studied extensively in the literature [12], the general case has not received much attention [10]. Multi-purpose parallel machine scheduling in general has been addressed under different names, such as "scheduling with processing sets" or "scheduling with eligibility constraints" [8]. Some authors prefer to use unrelated machines approach through setting the processing speed of ineligible machines very small. In this type of scheduling problems, each job has a predefined set of eligible machines called the job's processing set. The most general case of the multi-purpose machine scheduling problem is when the processing sets are arbitrary, which implies that each job processing set M_j can be an arbitrary subset of M, which is the set of all machines. This case is common when products with special characteristics are assigned to specific machines. Heuristics developed for solving the multi-purpose machine scheduling problem are usually the combination of two well-known dispatching rules: (i) least flexible job (LFJ) and (ii) least flexible machine (LFM) [18]. The effect of machine processing flexibility on the makespan of the system has been studied in [21]. They have shown that a small amount of processing flexibility is enough to achieve the performance of the identical parallel machines. They have also compared several heuristics and shown that the least average workload (LAW) and the least total workload (LTW) provide better results than other heuristics, such as the LPT. The order scheduling problem on multi-purpose machines has been studied in [26], where total completion time of orders has been considered as an objective function. Lower bound for the objective function was developed when the type splitting property is included, and several heuristics were proposed.

Multi-objective optimization has been receiving considerable attention in the literature in the past years [3]. This is related to the fact that solutions in real-world problems are evaluated based on multiple objectives. A few papers in the literature have addressed the multi-objective order scheduling problem. A non-dominated sorting algorithm-II (NSGA-II) was employed to solve a MSOS problem with three objective functions to minimize total tardiness, throughput time, and idle time; experimental results showed that the proposed model provided superior solutions compared to current industrial solutions [5]. The problem of minimizing collation delays in MOPA systems has been introduced in [13]. It was shown that this parameter significantly affects the throughput of the MOPA system, and it should be minimized when scheduling prescription orders [14]. An adaptive parallel tabu search (APTS) algorithm was proposed to solve this problem using the ϵ-constraint approach. The proposed algorithm outperformed the longest processing time (LPT) rule by 90–99 % and tabu search by 13–33 % in terms of collation delays [15]. This problem has been also studied in [17]. An NSGA-II was used and compared to other heuristics such as the vector evaluated

genetic algorithm (VEGA) and the multi-objective genetic algorithm (MOGA). It was shown that NSGA-II provides the best frontier and the most stable behavior in large job size problems. For the previous research on this problem in MOPA systems, an assumption was made that the machines are fully flexible. However, because the RDS units can only fill a specific set of medications according to the auto-dispensers assigned to them, machines in this scheduling model should be considered as multi-purpose.

3 Methodology

This research mainly investigates the effect of machine flexibility and the proportion of multi-medication orders on the total collation delays in the system. A multi-purpose machines environment, illustrated in Section 3.1, is adapted since the machines are subjected to eligibility constraints. The mathematical model developed for this problem is presented in Section 3.3. A min-max Pareto method, introduced in Section 3.4, is used to combine both objectives into a single one. The GA used in this problem is presented in Section 3.5, while LPT and LTW are illustrated in Section 3.6.

3.1 Multi-purpose machines approach

In this model, we have n medication jobs $\{j_1, \ldots, j_n\}$ and K parallel machines $\{m_1, \ldots, m_K\}$. Each medication job belongs to a prescription order i and has a predefined set of eligible machines, which is called the job's processing set. There are two types of orders: single medication orders and multi-medication orders. Medications that belong to multi-medication orders have to wait in a collation station until remaining medications within the same order are dispensed. It is assumed that there is no idle time, breakdown time, or setup time. Machines can process only one job at a time with no pre-emption allowed. The objective of this scheduling problem is to build a schedule that minimizes the makespan as well as the total collation delays of orders.

The makespan, denoted as $C_{\max}$, can be defined as the completion time of the last job leaving the system, and it depends on two factors: the number of jobs assigned to each machine and the processing times of these jobs. Let c_{ij} be the completion time of job j from order i. The schedule makespan can then be expressed as

$$C_{\max} = \max\{c_{11}, c_{12}, \ldots, c_{ij}\} \tag{1}$$

The order collation delay, which is defined as the completion time difference between dispensing the first and the last medication within one prescription order, can be mathematically expressed as

$$\Delta_i = L_i - E_i \tag{2}$$

where E_i and L_i are the completion times of the first and the last dispensed medications within the same order i.

To capture the multi-purpose machine environment, eligibility matrix A has been defined. In the real system, this matrix is defined through assigning dispensers to the RDS units. This process should be optimized based on the demand forecasting and the RDS unit planogram design. In this research, this matrix will be arbitrarily defined. An example of the eligibility matrix A is given in Table 1. As the table indicates, the value in this matrix is 1 if the job can be processed on this machine, 0 otherwise.

Table 1 Eligibility matrix A

Job	m_1	m_2	m_3
1	1	0	1
2	0	1	0
3	0	1	0
4	1	1	1

3.2 Notations

The list of notations used in this problem is presented in Table 2.

3.3 Mathematical model

The mathematical model developed for the problem is similar to the time indexed model in [20]. It was modified to capture the order scheduling problem on multi-purpose machine environment through adding the required constraints. In this model, it is assumed that there is no setup time, breakdown time, or idle time. Eligibility matrix A, which defines the processing sets of the machines, is assumed to be given as well. Time is considered as integer units and is given in seconds in subsequent sections.

$$\min C_{\max} \tag{3}$$

$$\min \sum_{i}^{I} (L_i - E_i) \tag{4}$$

s.t.

$$\sum_{m=1}^{K} \sum_{t=1}^{T} x_{ijmt} = 1 \qquad \forall i, j \tag{5}$$

Table 2 List of notations

Indexes	Description
i	Order index, $i \in \{1, 2, \ldots, I\}$
j	Job index, $j \in \{1, 2, \ldots, n_i\}$
m	Machine index, $m \in \{1, 2, \ldots, K\}$
t	Position index, $t, u \in \{1, 2, \ldots, T\}$
Data	**Description**
a_{ij}	1 if job j from order i can be processed on machine m, 0 otherwise
n_i	Number of jobs in order i
p_{ij}	Processing time of job j from order i
I	Number of orders
K	Number of machines
M	A very large number
Variables	**Description**
c_{ij}	Completion time of job j from order i
s_{ijmt}	Starting time of job j from order i in position t on machine m
x_{ijmt}	1 if job j from order i is processed in position t of machine m, 0 otherwise
$C_{\max}$	Schedule makespan
E_i	Earliest completion time of order i
L_i	Latest completion time of order i
Δ_i	Order collation delay

$$\sum_{i=1}^{I}\sum_{j=1}^{J} x_{ijmt} \leqslant 1 \qquad \forall m, t \qquad (6)$$

$$x_{ijmt} - a_{ijm} \leqslant 0 \qquad \forall i, j, m, t \qquad (7)$$

$$s_{ijm1} = 0 \qquad \forall i, j, m \qquad (8)$$

$$s_{ijm(t+1)} - \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{t} p_{ij} \cdot x_{ijmt} = 0 \qquad \forall i, j, m, t \qquad (9)$$

$$c_{ij} - x_{ijmt} \cdot (s_{ijmt} + p_{ij}) = 0 \qquad \forall i, j, m, t \qquad (10)$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T} p_{ij} \cdot x_{ijmt} - C_{\max} \leqslant 0 \qquad \forall m \qquad (11)$$

$$c_{ij} - L_i \leqslant 0 \qquad \forall i, j \qquad (12)$$

$$E_i - c_{ij} \leqslant 0 \qquad \forall i, j \qquad (13)$$

$$y_{mt}, c_{ij}, E_i, L_i \geqslant 0 \qquad \forall i, j, m, t \qquad (14)$$

$$x_{ijmt} \in \{0, 1\} \qquad \forall i, j, m, t \qquad (15)$$

Equation 3 is the objective function for minimizing the makespan, while Eq. 4 is the objective function for minimizing the total collation delays. Equation 5 ensures that all jobs are assigned to one position on one machine only, while Eq. 6 ensures that each position contains at most one job. Equation 7 guarantees that jobs are processed only on machines from their processing set. Equation 8 defines the starting time of the first position of each machine, while Eq. 9 specifies the starting time for jobs processed on other positions. Equation 10 establishes the completion time of each job. Equation 11 defines the schedule makespan. Equation 12 states the latest completion time of order i, meanwhile Eq. 13 specifies the earliest completion time of the order.

3.4 Min-max Pareto Approach

The min-max approach is a classical multi-objective optimization technique used to combine multiple objectives into a single one [1]. The goal is to choose a single best comprise solution through minimizing deviations from reference

Table 3 The list of GA parameters used

Population size	$10 \cdot n$
Number of iterations	$10 \cdot n$
Mutation method	2-opt swap
Mutation rate	0.05
Crossover method	2-point crossover
Crossover rate	1.0
Parent selection method	Binary selection

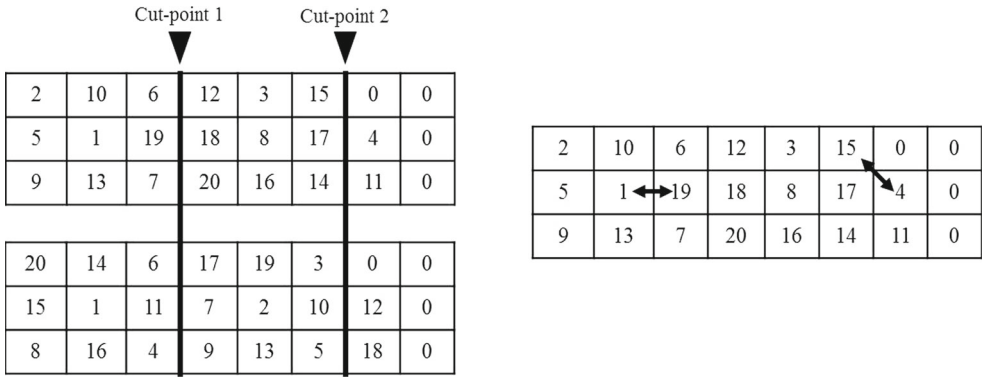

Fig. 2 GA crossover and mutation operators

values. To illustrate this concept, consider a problem with two objective functions f_1 and f_2. The deviations can be defined as $z_1 = f_1 - f_1^{\min}$ and $z_2 = f_2 - f_2^{\min}$, and the min-max objective function can be defined as

$$\min\{\max\{z_1, z_2\}\} \tag{16}$$

Because the makespan and the collation delays usually have different ranges of values, it is preferred to normalize deviations as

$$z_i = \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \tag{17}$$

For each new population in the genetic algorithm, the minimum and the maximum values are calculated from the solutions, and the objective function for each solution is calculated accordingly.

3.5 Genetic algorithm

A common method to design the chromosome in the parallel machines scheduling problem is to form an array that contains the jobs to be processed by each machine [22]. The array size is $K \cdot T$ and each cell represents a gene in the chromosome. The gene position represents the processing order of the job on the machine. In the first step, an initial population is created through randomly assigning jobs to eligible machines. Solutions are then evaluated using the min-max Pareto solution approach, and a binary selection process is conducted to select parents.

Table 4 Experimental results for different amounts of machine flexibility

			CPLEX (2 hours limit)		LPT		LTW		GA		
$F_p(\%)$	Job size	$C^*_{\max}$	$C_{\max}$	Δ	$C_{\max}$	Δ	$C_{\max}$	Δ	$C_{\max}$	Δ	CT
0	12	79	79	31	79	58	79	58	79.0	31.0	6.1
	24	140	140	30	140	222	140	222	140.0	30.8	30.7
	48	352	352	78	352	1044	352	1044	352.0	78.0	224.4
	96	728	728	335	728	4989	728	4989	728.0	267.0	1969.5
	120	771	771	780	771	8256	771	8256	771.0	335.6	3612.8
50	12	62	62	3	89	115	78	103	62.0	3.0	6.3
	24	123	123	2	136	234	125	268	123.0	4.4	35.2
	48	245	245	49	246	988	252	1203	245.0	26.8	283.5
	96	493	502	198	493	4264	496	4088	494.8	200.8	2351.1
	120	620	635	347	620	6293	624	6375	631.4	281.2	4571.3
100	12	61	61	2	61	73	61	73	61.0	2.0	6.2
	24	123	123	3	123	238	123	238	123.0	2.8	51.4
	48	245	248	56	245	1002	245	1002	245.0	23.8	434.4
	96	493	497	369	493	4223	493	4233	493.0	147.4	3667.3
	120	620	624	464	620	6280	620	6280	620.0	189.8	4736.4

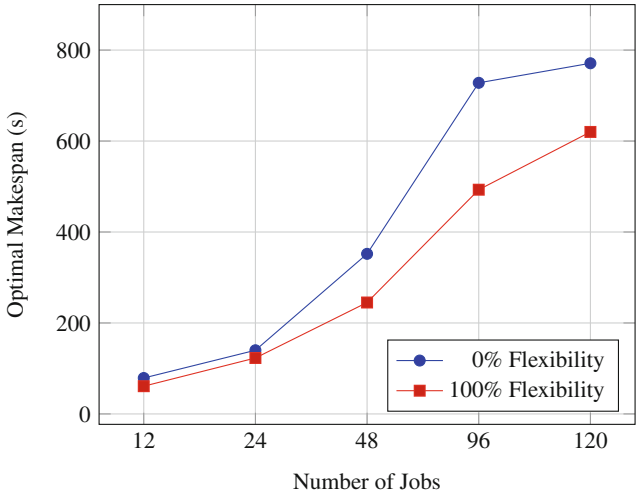

Fig. 3 Relationship between machine flexibility and the makespan

A two-point crossover operator is applied in this GA to produce offspring. Because the crossover process may result in corrupted offspring (repeated and missing jobs), a repair process was designed to transfer duplicate jobs from one child to the other. A job swap mutation is applied to the offspring in the next step to improve the diversity of the population. In this method, a job is swapped with another job either from the same or different machine. However, this process is constrained by the processing assignment restrictions to ensure the feasibility of the solution produced. The GA parameters, summarized in Table 3, were tuned using design of experiment concepts (n refers to the total number of the jobs in the problem). Figure 2 illustrates the chromosome design as well as the crossover and mutation operators.

Algorithm 1 Genetic algorithm

1. Create initial population P_0
2. Evaluate objective function for each solution in P_0
3. Binary selection to select parents
4. Two point crossover followed by repair process
5. Constrained mutation process to create offspring
6. Create new population. Go back to Step 2 if stopping criteria is not satisfied, else return best value.

3.6 LPT and LTW

Two popular heuristics are used to benchmark the effectiveness of the GA. The first heuristic is the longest processing time (LPT), which prioritizes the jobs in the list according to the length of their processing time. LPT is known to perform well in parallel identical machine environments and has a lower bound of $\frac{\sum_{j=1}^{n} p_j}{m}$. This algorithm can be summarized in the following steps:

1. Pick the job with the longest processing time.
2. Assign the job to the first available eligible machine.
3. Repeat until all jobs are assigned.

The second heuristic is the least total workload (LTW), which considers the current and the unassigned workload of each machine [21]. LTW chooses a machine first and then assigns a job, unlike LPT, which picks a job and then a machine. Let W_m represent the unassigned workload of machine m, and S_m represent the current workload of machine m. This algorithm can then be summarized in the following steps:

1. Find machine m with minimum $S_m + W_m$.
2. List all jobs that can be processed on machine m.
3. Pick the job with the LPT and assign it to machine m.
4. Repeat until all jobs are assigned.

4 Experimental results and analysis

The objective of this analysis is to understand how machine flexibility and multi-medication orders proportion affect the system performance, mainly in terms of collation delays. The measurement defined by Vairaktarakis and Cai (2003) to indicate the overall process flexibility of the system [21]. The value of this measure is between 0 and 100 %. A system with $F_p = 100$ % indicates a fully flexible system,

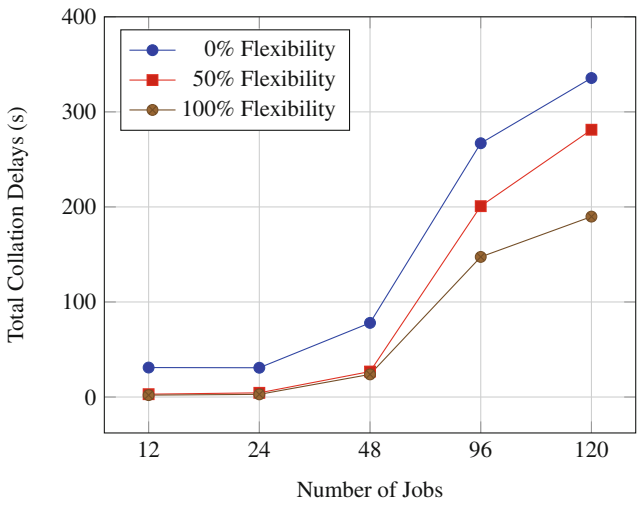

Fig. 4 Relationship between machine flexibility and the total collation delays

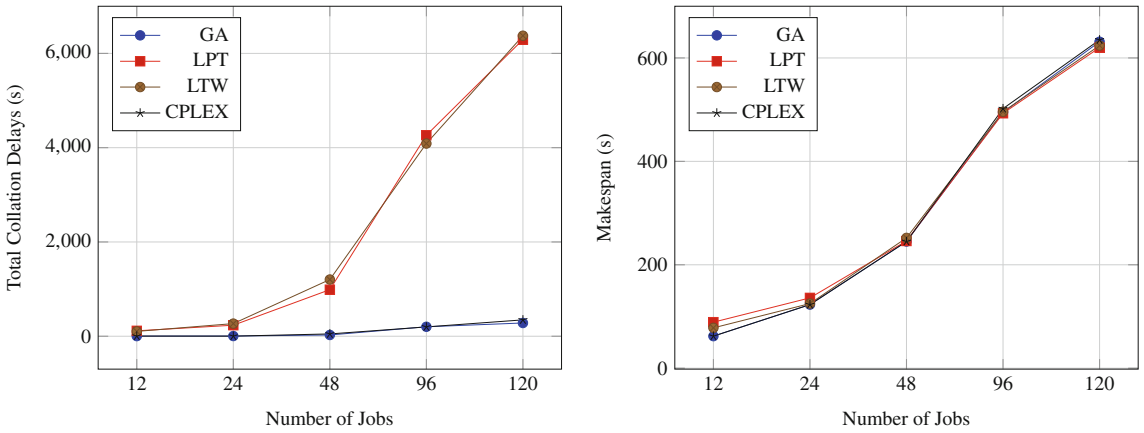

Fig. 5 GA performance compared to LPT and LTW ($F_p = 50\ \%$)

while a system with $F_p = 0\ \%$ indicates a fully dedicated system. This measure can be mathematically expressed as

$$F_p = \frac{\sum_{j,k} A_{jk} - n}{n(m-1)} \cdot 100\ \% \tag{18}$$

where n is the number of jobs, and m is the number of machines. Three different scenarios were tested: (1) fully dedicated machines scenario ($F_p = 0\ \%$), (2) multi-purpose machines scenario ($F_p = 50\ \%$), and (3) fully flexible machines scenario ($F_p = 100\ \%$).

To evaluate the effect of multi-medication orders proportion, three scenarios were tested, which depend on the ratio of jobs that belong to multi-medication orders (25, 50, and 75 %). Processing times were generated from a uniform random distribution ($U \backsim [14, 17]$), and the eligibility matrix A was defined arbitrarily. All scenarios were solved using CPLEX, LPT, LTW, and the GA. The optimal makespan was also calculated using the mathematical model without considering the total collation delays to provide a benchmark for the heuristics. Algorithms were developed using Matlab R2012a, and the experiments were conducted on a

Table 5 Experimental results for different proportions of multi-medication orders

			CPLEX (2 hours limit)		LPT		LTW		GA		
Multi-medication (%)	Job size	$C^*_{\max}$	$C_{\max}$	Δ	$C_{\max}$	Δ	$C_{\max}$	Δ	$C_{\max}$	Δ	CT
25	12	62	62	1	89	72	78	41	62.0	1.0	6.6
	24	123	123	0	136	161	125	178	123.0	0.4	37.2
	48	245	246	33	246	575	252	568	245.0	5.8	280.4
	96	493	505	76	493	1920	496	1690	493.0	74.8	2542.3
	120	620	633	346	620	3120	624	3184	620.0	137.0	3298.5
50	12	62	62	3	89	115	78	103	62.0	3.0	6.3
	24	123	123	2	136	234	125	268	123.0	4.4	35.2
	48	245	245	49	246	988	252	1203	245.0	26.8	283.5
	96	493	502	198	493	4264	496	4088	494.8	200.8	2351.1
	120	620	635	347	620	6293	624	6375	631.4	281.2	4571.3
75	12	62	62	5	89	145	78	119	62.0	7.4	5.8
	24	123	125	27	136	385	125	406	123.0	22.4	32.8
	48	245	247	137	246	1503	252	1648	245.0	64.6	232.7
	96	493	503	401	493	8226	496	8461	493.0	390.2	2521.2
	120	620	635	694	620	10257	624	10306	620.0	575.2	5182.8

PC with an Intel core i7-2600 CPU @ 3.4 GHz and 16 GB RAM.

4.1 Analysis of machine flexibility effect

Fully flexible machine environment is always desirable in manufacturing systems; however, there are many constraints in the real world that reduce the flexibility of the machines. In MOPA systems, the RDS unit is only capable of processing a specific subset of medications depending on the auto-dispensers assigned to it. To increase the system flexibility, the process of assigning medications to dispensers should be optimized to handle uncertainties in demand. Results illustrating the effect of machine flexibility on the makespan and the total collation delays are shown in Table 4.

Figure 3 shows the relationship between machine flexibility and the makespan. When the machine flexibility is 0 % (fully dedicated machine environment), the makespan increases by 25–50 % as compared to the case where the machine flexibility is 100 % (fully flexible machine environment). However, when the machine flexibility increases to 50 % in the multi-purpose machine environment, the model was able to achieve the same performance as in the fully flexible machine environment in terms of the makespan. In general, it was proven that small amounts of flexibility are required to achieve the performance of fully flexible environments with respect to the makespan [21].

Because LPT and LTW do not consider collation delays, only GA results were used to analyze the effect of machine flexibility on the total collation delays. As shown in Fig. 4, the total collation delays for the fully dedicated machine environment were 80 % higher than the fully flexible machines case, and 25 % higher than multi-purpose machine environment for large job size problems. The effect of machine flexibility can also be observed when comparing the multi-purpose machine environment to the fully flexible machine environment. When machine flexibility is 50 %, the total collation delays increase by 40 % on average for large job size problems. Generally, when the machine flexibility in the system decreases, the chances for jobs within the same order to be processed on the same machine increase, which subsequently leads to a larger order collation delay.

Figure 5 shows the GA performance compared to CPLEX, LPT, and LTW in terms of the makespan and the total collation delays. For the makespan, the GA achieved the optimal makespan in most of the cases. However in some cases, the GA favored solutions with fewer collation delays over solutions with optimal makespan and large collation delays. Compared to the CPLEX solution after the 2 hours limit, the GA generated 35 % less total collation delays in large size problems. LTW outperformed LPT for small job size problems by 10 % on average; however, for large size problems, both algorithms behaved similarly in achieving the optimal makespan. It can also be seen that the GA minimizes the total collation delays by 90–95 % in large job size problems as compared to LPT and LTW.

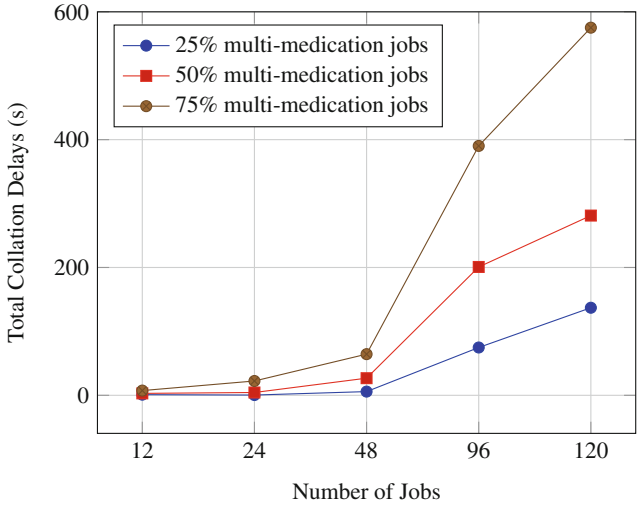

Fig. 6 Effect of multi-medication orders proportion on collation delays

4.2 Analysis of multi-medication orders proportion effect

In MOPA systems, prescription orders received can be single or multi-medication orders. The percentage of multi-medication orders varies from one MOPA system to another depending on the nature of the prescriptions demand received. To study the effect of multi-medication orders on the total collation delays, three scenarios were tested. In each scenario, the percentage of jobs belonging to multi-medication orders was varied (25, 50, and 75 %) and the machines flexibility was fixed at 50 %. Experimental results for these scenarios are shown in Table 5.

From the results, it can be seen that a higher percentage of multi-medication jobs leads to a larger total collation delays. Systems with 25 % of multi-medication jobs have 50–60 % fewer total collation delays than systems with 50 % of multi-medication jobs, and 75–80 % fewer collation delays than systems with 75 % of multi-medication jobs, as shown in Fig. 6.

5 Conclusions and future work

In this paper, we have studied the MOP of minimizing collation delays and the makespan in MOPA systems. This scheduling problem is formally defined in the literature as

the order scheduling problem. This model can be found in many applications in the industry, especially in MTO manufacturing systems. A special characteristic of this system is that orders have to be collated before packaging and shipping. The order collation delay is defined as the completion time difference between the first and the last items within the same order. This parameter significantly affects the throughput of the system and may cause deadlocks in the material handling system used. There are many factors in the system influencing orders collation delay. This research has studied the effect of machines flexibility and the proportion of multi-medication jobs.

To solve this NP-hard problem, a GA was used and compared with LPT and LTW. Experimental results indicate that both factors have a significant effect on the total collation delays in the system. A fully dedicated machine environment ($F_p = 0$ %) has 80 % more total collation delays compared to a fully flexible machine environment ($F_p = 100$ %), and 25 % more total collation delays compared to a multi-purpose machine environment ($F_p = 50$ %). When studying the effect of multi-medication jobs proportion, it was shown that a system with 25 % jobs of multi-medication jobs has 75–80 % fewer collation delays than a system with 75 % jobs of multi-medication jobs.

In manufacturing environments, it is often the case that the decision maker is interested in obtaining a set of alternative solutions rather than a single solution. This enhances the systems flexibility in adapting to different operational situations, especially when the number of objectives increases. Classical optimization methods have limitations in investigating the Pareto frontier set, and requires several simulation runs to find multiple solutions [4]. In this case, the Pareto-ranking approach is favorable since it provides a set of non-dominated solutions. The Pareto-ranking approach is based on the evolutionary algorithms (EA), which have been widely applied in multi-objective optimization problems in recent years, due to their population-based optimization characteristics. The future research could employ and compare several MOEAs such as the non-dominated sorting genetic algorithm II (NSGA-II) [4] and the strength Pareto evolutionary algorithm II (SPEA-II) [28].

Although the GA achieved optimal makespan in most of the scenarios while minimizing the total collation delays by 90–95 % as compared to LPT and LTW, the computational time for large job size scenarios was high. In the industry, computational efficient heuristics that generate near optimal solutions are preferred.. Future work would include designing computational efficient heuristics that take into consideration the unique characteristics of this problem. These heuristics will be designed based on several rules and factors, including machines flexibility (least flexible machine (LFM)), orders priority (assigning priorities based on number of items per order), and overall system performance measures (makespan, total completion time, tardiness).

In this research, the eligibility matrix A, which defines the processing set of each job, was arbitrarily defined. In the actual MOPA system, this matrix is defined through a two-stage assignment process. In the first stage, medications are assigned to an auto-dispenser which depends mainly on the prescriptions demand. In the second stage, the auto-dispensers are assigned to parallel RDS units. RDS units are only capable of processing a specific subset of medications depending on the auto-dispensers assigned. This assignment process can be optimized through efficient planogram techniques. The future research could include the planogram design optimization to increase the flexibility of the RDS, which is expected to significantly enhance the MOPA system performance.

Acknowledgments This study was supported by the Watson Institute of Systems Excellence (WISE) at Binghamton University and by Innovation Associates. The authors also wish to thank several colleagues who have inspired and provided valuable comments to improve this study: Boyer, T. and Lashier, A. from Innovation Associates and Poch, L. from WISE.

References

1. Belegundu A, Chandrupatla T (2011) Optimization concepts and applications in engineering Cambridge University Press
2. Blocher J, Chhajed D (1996) The customer order lead-time problem on parallel machines. Naval Research Logistics (NRL) 43(5):629–654
3. Deb K (2001) Multi-objective optimization using evolutionary algorithms, vol. 16 John Wiley & Sons
4. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evol Comput 6(2):182–197
5. Guo Z, Wong W, Li Z, Ren P (2013) Modeling and pareto optimization of multi-objective order scheduling problems in production planning. Computers & Industrial Engineering 64(4):972–986
6. Ioannou G, Dimitriou S (2012) Lead time estimation in mrp/erp for make-to-order manufacturing systems. Int J Prod Econ 139(2):551–563
7. Julien FM, Magazine MJ (1990) Scheduling customer orders: an alternative production scheduling approach. Journal of Manufacturing and Operations Management 3(3):177–199
8. Leung JY, Li CL (2008) Scheduling with processing set restrictions: a survey. Int J Prod Econ 116(2):345–364
9. Leung JY, Li H, Pinedo M (2005) Order scheduling in an environment with dedicated resources in parallel. J Sched 8(5):355–386
10. Leung JY, Li H, Pinedo M, Kendall G, Burke E, Petrovic S, Gendreau M (2005) Order scheduling models: an overview. Springer, U.S
11. Leung JY, Li H, Pinedo M (2006) Approximation algorithms for minimizing total weighted completion time of orders on identical

machines in parallel. Naval Research Logistics (NRL) 53(4):243–260
12. Leung JY, Li H, Pinedo M (2008) Scheduling orders on either dedicated or flexible machines in parallel to minimize total weighted completion time. Ann Oper Res 159(1):107–123
13. Li D, Yoon SW (2012) Minimizing fill-time window in central fill pharmacy. In: Industrial and systems engineering research conference (ISERC), orlando, FL
14. Li D, Yoon SW (2012) Simulation based manova analysis of pharmaceutical automation system in central fill pharmacy. In: International conference on industrial engineering and engineering management (IEEM) Hong Kong, China
15. Li D, Yoon SW (2015) A novel fill-time window minimisation problem and adaptive parallel tabu search algorithm in mail-order pharmacy automation system. Int J Prod Res 53(14):4189–4205
16. Li H, Womer K (2012) Optimizing the supply chain configuration for make-to-order manufacturing. Eur J Oper Res 221(1): 118–128
17. Mei K, Li D, Yoon SW, Ryu J (2015) Multi-objective optimization of collation delay and makespan in mail-order pharmacy automated distribution system. Int J Adv Manuf Technol:1–14
18. Pinedo M (2012) Scheduling: theory, algorithms, and systems Springer Science & Business Media
19. Piya S, Al-Hinai N (2015) Production planning for make-to-order flow shop system under hierarchical workforce environment. In: Transactions on engineering technologies, pp 317–330. Springer
20. Unlu Y, Mason S (2010) Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems. Comput Ind Eng 58(4):785–800
21. Vairaktarakis G, Cai X (2003) The value of processing flexibility in multipurpose machines. IIE Trans 35(8):763–774
22. Vallada E, Ruiz R (2011) A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. Eur J Oper Res 211(3):612–622
23. Wang B, Guan Z, Chen Y, Shao X, Jin M, Zhang C (2013) An assemble-to-order production planning with the integration of order scheduling and mixed-model sequencing. Frontiers of Mechanical Engineering 8(2):137–145
24. Wang G, Cheng T (2007) Customer order scheduling to minimize total weighted completion time. Omega 35(5):623–626
25. Weng ZK (1996) Manufacturing lead times, system utilization rates and lead-time-related demand. Eur J Oper Res 89(2):259–268
26. Xu X, Ma Y, Zhou Z, Zhao Y (2015) Customer order scheduling on unrelated parallel machines to minimize total completion time. IEEE Trans Autom Sci Eng 12(1):244–257
27. Yang J (2005) The complexity of customer order scheduling problems on parallel machines. Comput Oper Res 32(7): 1921–1939
28. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. Eidgenössische Technische Hochschule zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK)