Research Journal of Applied Sciences, Engineering and Technology 6(9): 1566-1572, 2013
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: November 08, 2012 Accepted: January 05, 2013 Published: July 15, 2013

Two-Dimensional Scheduling: A Review

Zhuolei Xiao and Hao Shao
School of Computer and Information, Fuyang Teachers College, China
School of WTO Research and Education, Shanghai Institute of Foreign Trade, China

Abstract: In this study, we present a literature review, classification schemes and analysis of methodology for scheduling problems on Batch Processing machine (BP) with both processing time and job size constraints which is also regarded as Two-Dimensional (TD) scheduling. Special attention is given to scheduling problems with non-identical job sizes and processing times, with details of the basic algorithms and other significant results.

Keywords: Algorithm, batching, complexity, scheduling, review

INTRODUCTION

As a literature review focusing entirely on one particular aspect of scheduling theory, this study offers detailed information and guidance for specific area of scheduling research. In recent years, several reviews of batch scheduling are presented. For example, Mathirajan and Sivakumar (2006) provides a literature review, classification schemes and a simple meta-analysis for Scheduling of Batch Processors (SBP) research in Semiconductor Manufacturing (SM) and summarizes current research results for different problem types. Chris *et al*. (2000) provides an extensive literature review on models of integrate scheduling with batching decisions. However, as far as our research scope concerns, there exists no such review which put dominant emphasis on batch scheduling with constrains of both job sizes and processing time. We regard this kind of issue as “Two-Dimensional Scheduling”. The idea of “two-dimensional” had been introduced by Gilmore and Gomory (1965) to deal with the cutting stock problem. In two-dimensional cutting stock problems, stock units have not only length but also width restrictions. This model resembles our scheduling problems where both processing time and job sizes should be considered. The purpose of this study is to provide specific literature review, classification schemes and research methodologies of this aspect. In our study, details of the basic algorithms and references of other significant results are presented along with our elaborate analysis.

Batch scheduling is of great significance in many manufacturing industries. The manufacturing process of Integrated Circuit (IC) chips is often composed of four major steps including wafer fabrication, wafer probe, assembly and final testing. The burn-in operation in the final testing step is the critical working procedure in the whole process. Chips are loaded on boards, placed in an oven and exposed to high temperature (generally 120C) for an extended period of time in order to bring out any chips with latent defects leading to infant mortality that might otherwise surface in the operating environment. Commonly, the processing time would be longer than 120 h, which is several times more than other procedures. Thus the burn-in operations is generally considered as the bottleneck process in the final testing step. The burn-in time for each chip is specified by customers (e.g., military purpose or civil use). However, in some circumstances, the time could be extended but no shrinking is allowed. Except the difference between processing time of each chip, the size may also be different. It is possible for an oven to be loaded with a number of chips and the processing time of a batch is equal to the longest processing time among all chips in the batch. Once the processing begins on a batch, no chip can be removed from the oven until the whole process is complete. For abstract, consider the chips as jobs and the ovens as machines. The whole process then can be regarded as scheduling a single batch processing machine with non-identical jobs sizes and processing time.

These issues are often encountered in practice such as port handling, lathe machining and ceramic making. Based on practical scheduling problems in manufacturing, this specific scheduling issue then can be abstracted as a Two-Dimensional model. Two-dimensional means that the jobs have not only constraints of processing time, but also limitations of machine capacity. In this scheduling model, machines are batch processors that can handle several jobs simultaneously and the processing time of a batch is represented by the longest job processing time among those of all the jobs contained in the batch. All jobs belonging to the same batch are completed at the same time. Different jobs may have different sizes and the sum of the sizes of the jobs contained in a batch cannot

Corresponding Author: Zhuolei Xiao, School of Computer and Information, Fuyang Teachers College, China

exceed the batch capacity. Then the question is how to divide the jobs into batches and arrange the sequence in order to satisfy the objective function. The significance of this scheduling model lies in two aspects as following:

- The economic prospect is promising
- In practical aspect, contradiction emerges between utilize ratio of machines and completion time of jobs. Thus, it is of great significance to deal with contrary requirements of each aspect in practical scheduling which is more complicated.

PROBLEM SPECIFICATIONS

In this study, we put emphasizes on the TD scheduling model. A typical TD scheduling model consists of the following assumptions:

- The collection of jobs is J = {1, 2 … n} and the processing time of job j is p_j , the size of job j is sj
- The processing of jobs is in the form of batches, b_k, k = {1 … m}, where 1 ≤ m ≤ n, which contains one or more jobs. The capacity of the machine is B, that is, the total size of any batch cannot exceed B
- The processing of a batch cannot be interrupted and the processing time of a batch b_k is P_k, which equals to the longest time of all the jobs in bk.

According to the above description, the mathematical model of minimizing makespan (Cmax) is as follows:

$$Min \quad C_{\max} = \sum_{k=1}^{m} P_k \tag{1}$$

$$s.t. \quad \sum_{k=1}^{m} Y_{jk} = 1 \quad j = 1, 2 \dots, n \tag{2}$$

$$\sum_{j=1}^{n} s_j Y_{jk} \le B \quad k = 1, 2 \dots, m \tag{3}$$

$$P_k \ge P_j Y_{jk} \quad j = 1, 2 \dots n \; ; \; k = 1, 2 \dots m \tag{4}$$

$$Y_{jk} = \{0,1\} \quad j = 1, 2 \dots n \; ; \; k = 1, 2 \dots m \tag{5}$$

$$P_k \ge 0 \tag{6}$$

$$\sum_{j=1}^{n} \frac{s_j}{B} \le m \le n \tag{7}$$

Objective (1) indicates that the make span is the sum of processing time of all batches. Y_{jk} is a 0-1 variable and Y_{jk} equals to 1 when j ∈ bk, or else, Y_{jk} equals to 0. In constraint set (2) and (5), we can see that any job j can belong to at most one batch. Constraint set (3) ensures that the capacity of the machine cannot be exceeded. Constraint set (4) gives the definition of P_k and (6), (7) are general restrictions of the variables.

LITERATURE REVIEW

Single batch processing machine:

- **Identical job size:**
- **No family or compatible job family:** The batch scheduling issues was probably firstly raised and studied by Ikura and Gimple (1986). An O(n2) rule was proposed to determine whether feasible schedule exists considering jobs with identical processing time where release times and due dates were also encountered. The algorithm could find a feasible schedule to minimize finishing time.

Lee *et al*. (1992) studied batch scheduling problems where different jobs can be batched together and the processing time of a batch equals to the longest processing time of all jobs in that batch. Chandru *et al*. (1993a) proposed branch and bound algorithms to solve scheduling problem with single batch processing machines exactly. The model was also extended to parallel machine scheduling. Chandru *et al*. (1993b) showed that if the number of job families is fixed and all jobs in the same family have identical processing time, the problem to minimize total completion time could be solved in polynomial time.

Lee and Uzsoy (1999) studied scheduling problem of single batch processing machine with dynamic job arrives (the release time of jobs are not same) and proposed polynomial and pseudo-polynomial algorithms under some special instances. Wang and Uzsoy (2002) considered single batch scheduling where jobs have release times.

- **Incompatible job family:** When we refer incompatible job families scheduling problems, we mean that jobs in different family cannot be assigned to the same batch.

Uzsoy (1995) studied batch scheduling problems of incompatible job families under different objective functions. Kim *et al*. (2000) considered the problem of minimize total tardiness on a single batch processor and proposed an effective heuristic algorithm.

- **Non-identical job sizes:**
- **No Family or compatible job family:** Dobson and Nambimadom (2001) addressed batch scheduling problems with non-identical job sizes. Parsa *et al*. (2010) considered the scheduling problem on a

single batch processing machine with nonidentical job sizes for minimizing the makespan. Kashan *et al*. (2010) studied the problem of scheduling jobs with non-identical sizes on a single batch processing machine.

- **Incompatible job family:** Monch *et al*. (2005) addressed problems on parallel batch processing machines where jobs have non-identical ready times.

Perez *et al*. (2005) focused on scheduling problems on a single batch processing machine with non-identical job sizes. Monch *et al*. (2006) studied scheduling problem in the presence of dynamic job arrives in diffusion and oxidation process of semiconductor manufacturing. Malve and Uzsoy (2007) considered scheduling problems on parallel batch processing machines with several parallel incompatible job families where jobs have non-identical release time and due date. Kashan *et al*. (2008) developed an ant colony framework in two versions, depending on the type of embedded heuristic information, to minimize total weighted completion time on a single batch-processing machine with incompatible job families and arbitrary job sizes.

Parallel batch processing machine: Lee *et al*. (1992) studied worst-case error bound of any list scheduling algorithms and parallel machine problems to minimize makespan, they also proposed a worst-case error bound of a list scheduling algorithm to minimize the maximum lateness.

Balasubramanian *et al*. (2004) provided two different genetic algorithms based on decomposition approaches of incompatible job families to minimize total weighted tardiness. Monch *et al*. (2005) extended their algorithms to situations when jobs have non-identical ready times. Damodaran and Velez-Gallego (2010) studied the scheduling problem of parallel identical batch processing machines to minimize the makespan. Chung *et al*. (2009) considered the parallel batch processing machine scheduling problem which involves the constraints of unequal ready times, non-identical job sizes and batch dependent processing times. Wang and Chou (2010) studied the scheduling problem of parallel batch-processing machines in the presence of dynamic job arrives and non-identical job sizes to minimize the makespan.

Sabouni *et al*. (2010) considered scheduling problems on parallel machines in batches for minimizing the total completion time together with the maximum lateness.

Many researchers also put emphasis on scheduling problems of parallel batch processing machines such as Chandru *et al*. (1993a), Uzsoy (1995), Hochbaum and Landy (1997) and Chandra and Gupta (1997). However, they all considered jobs have identical sizes. Jobs with non-identical sizes are studied by Dobson and Nambimadom (2001), Uzsoy (1994) and Kempf *et al*. (1998).

METHODOLOGY AND ANALYSIS

General description: This is the main part of our study. A typical classification of batch processing problems is proposed by Mathirajan and Sivakumar (2006), as shown in Fig. 1.

Complexity analysis: Uzsoy (1994) gave proposition for special case of the problem:

Proposition 1: When all jobs require identical processing times, C_{max} is equivalent to a bin-packing problem with bin capacity Band item sizes α_i.

And this lead to directly the following corollary:

Corollary 1: C_{max} is strongly NP hard.
The author also proved that:

Proposition 2: ΣC_i with identical processing times is NP hard.

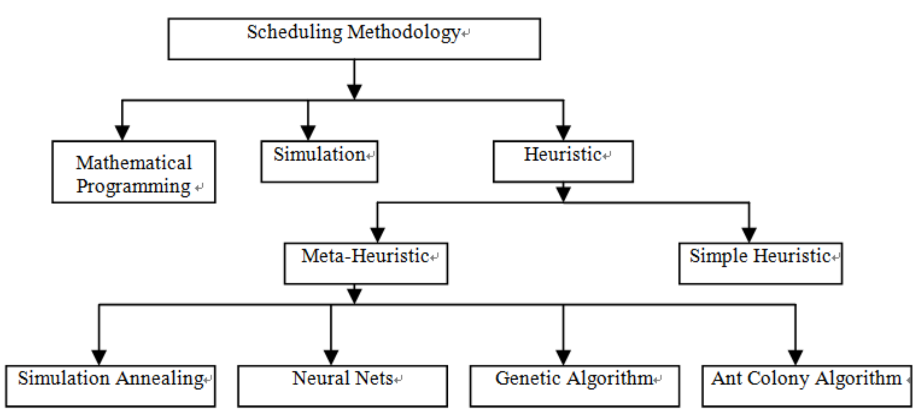

Fig. 1: A typical classification of batch processing problems

For parallel batch processing machines, (Lee *et al*., 1992) proved that P| B| C_{max} is NP hard in the strong sense even for B =1.

Competitive ration analysis: An algorithm A is a ρ-approximation algorithm for a minimization problem if it produces a solution which is at most ρ times the optimal one, in time that is polynomial in the input size. We also say that ρis the worst-case ratio of algorithm A. The worst-case ratio is the usual measure for the quality of an approximation algorithm for a minimization problem: the smaller the ratio is, the better results the approximation algorithms can get.

For 1|B, S_i|C_{max}, based on FF (First Fit) algorithm of bin packing problem (Coffman *et al*., 1984; Uzsoy, 1994) studied the problem and proposed BFF (Batch First Fit) algorithm. Using the characteristics of bin packing problems, they provided competitive ratio as:

$$C_{max}(BFF) \le \frac{17}{10} C^*_{max} + p$$

The FFLPT heuristic based on BFF is proved to have a competitive ratio of 17/10.

For 1|B, ri, pi, si| C_{max}, Shuguang *et al*. (2005) considered scheduling problem on a single batch processing machine with consideration of job release times and non-identical job sizes. They proposed an approximate algorithm with competitive ratio 2 + ε (ε tends to 0).

For P|pi, si, di| C_{max}, Lingfa and Jinjiang (2008) studied scheduling problems on parallel batch processing machine with unbounded batch capacity where jobs have delivery time.

Branch and bound: These Lower Bounds (LB) are proposed to be benchmarks for the results of other algorithms.

For 1|B, pi, si| C_{max}, Uzsoy (1994), Dupont and Dhaenens-Flipo (2002) and Kashan *et al*. (2006) studied such problems and in Dupont and Dhaenens-Flipo (2002), a lower bound on the optimal C_{max} can be calculated by relaxing the problem and allowing jobs to be split and processed in different batches. This is done by constructing an instance of C_{max} where each job i, of the original problem, is replaced with s jobs of unit size and processing time p_i. This can easily be solved by sorting jobs in the LPT order (decreasing order of their processing time), successively grouping the non-added jobs with longest processing times into the same batch and then processing batches in any order.

When there are some jobs that cannot be grouped with any other jobs in the same batch, the LB is modified as following:

- Put the jobs satisfying following relation in the set J and remove them from the set of whole jobs: J = -{k| B – αk < min $\{\alpha i\}_{ie\{1, \dots ,n\}}$}
- For the reduced problem, construct an instance of C_{max} (C_{max}^{LB}) with unit job sizes where each job m is replaced by A_m numbers jobs of unit size and processing time $t_m$which is the processing time of job m. This can be solved by ordering jobs in decreasing order of processing times, successively grouping the B jobs with longest processing times into the same batch.
- The modified lower bound, $C_{\max}^{MLB}$ can be obtained as:

$$C_{\max}^{MLB} = \sum_{i \in J} p_i + C_{\max}^{LB}$$

Uzsoy (1994) and Jolai and Dupont (1998) considered 1|B, pi, si | ΣC_i. In Jolai and Dupont (1998), to develop a lower bound, the author relaxed the problem by splitting each job I of size si into si unit-size jobs ij, j = 1, …, si and replaced the batch processing machine with B parallel identical unit-capability machines. The following expression provides a lower bound on the optimal value of the problem. C_1 denotes the optimal $\Sigma W_i C_i$ value of the problem of scheduling all Σsi unit-sized jobs ij with weights 1/si on a single unit-capacity machine:

$$\frac{1}{B}[C_1 - \frac{1}{2}\sum_{i=1}^{n} p_i] + \frac{1}{2}\sum_{i=1}^{n} p_i$$

For 1|B, p_i, s_i| $\Sigma W_i C_i$, Azizoglu and Webster (2000) provided two lower bounds. One is the LB_{AK} :

$$LB_{AK} = \sum_{j=1}^{n} [\omega_{\tau\omega(n-j+1)}(C_j - p_j) + \omega_i p_j]$$

$$Cj = \begin{cases} p_j, & j \le m \\ C_{j-m} + p_j, & J > m \end{cases}$$

where, m is the maximum number of jobs a batch could accommodate. τω(n – j + 1) is the (n-j+1) weight order by increasing.

The other lower bound is LBs:

$$LB_s = F_1^* / B + F_n^* / 2 - R / 2$$

where, F^*_1 is the optimal total weighted completion time of a single processor problem with job processing time $\alpha_j p_j$ and weights ωj. F*n = $\sum_j \omega_j p_j$ and R = $\sum_j \omega_j \alpha_j p_j$/B.

Simple heuristic: Uzsoy (1994) proposed the typical model for scheduling single batch machine with non-identical job sizes. Several simple heuristics are also presented in that study to solve problems with objectives of C_{max} and ΣC_i.

Another type of heuristic algorithm is based on a greedy ratio procedure which takes both processing time and batch utilization information into account. Jolai and Dupont (1998) made some modification on GR algorithm and proposed another BFGR algorithm. The DYNA algorithm in this study, which obtains the result by optimizing partial schedule, was proved very effective on solving ΣC_i problems in considerable scale.

Meta-heuristic (intelligent algorithm): The first intelligent algorithm used for TD scheduling is the simulation annealing (SA) algorithm proposed by Melouk *et al*. (2004).

Researches on these intelligent algorithms mainly focus on the development of GA (Genetic Algorithm). This algorithm was introduced in the 1970s by Holland. Wang and Uzsoy (2002) apply GA on a batch processing machine in the presence of dynamic job arrivals to minimize maximum lateness. Sevaux and Peres (2003) employ GA to minimize weighted number of late jobs on a single batch processing machine. Koksalan and Keha (2003) have minimized the flow time, the number of tardy jobs and the maximum earliness by employing GA. Readers could see Reeves (1997) to get an overview of successful application of GA in business and industry.

Genetic algorithms are stochastic search algorithms based on the mechanism of natural selection and natural genetics. The GA is applied to spaces that are too large to be exhaustively searched (Goldberg, 1989).

To the best of our knowledge, only hybrid GA could be possible approaches to solve such kind of issues. The addition of simple heuristics makes many problems easier to solve. In Purushothaman *et al*. (2006), GA was proposed to solve TD scheduling problems.

We can conclude that the construction is still based on heuristics. By evolution from generation to generation, GA could obtain optimal solution finally. GA was extensively studied in recent years and many developments are achieved. Thus applicable techniques and methods could be used in hybrid GA algorithms. Ali *et al*. (2006), the author proposed a hybrid GA with two different encoding schemes to optimize current GA. One is sequence based GA that generates random sequences of jobs and applies the Batch First Fit (BFF) heuristic to group the jobs. The other one is a batch based hybrid GA that generates random batches of jobs and ensures feasibility through using knowledge of the problem. The second method uses pair wise swapping heuristic based on the problem characteristics which consider the space constrains of jobs in GA sufficiently and improves the traditional GA effectively.

Random key method was often used in crossover procedure in Bean (1994). It represents order of jobs with a sequence of uniformly distributed random numbers. The advantages of this method consist in that in crossover procedure, it is no need to pair wise dramatically but only to arrange numbers with their value to obtain the solution. Readers are referred to Malve and Uzsoy (2007) for detailed procedure of this method.

For parallel machine scheduling problems, a hybrid GA was propose by Ali *et al*. (2008) where heuristics are applied in. Random Batches Procedure (RBP) was presented in batch construction scheme, which constructs a feasible batching plan through simultaneously minimizing the residual batch capacity. In hybrid GA, two local search heuristics are proposed. One is to swap jobs between batches when constructing batches, the other is to make interchange of batches between machines to reduce the make span. The basic procedure of GA does not change much.

However, the efficiency of GA depends on the scale of the population. With the scale becomes larger, the astringency of GA gets worse. Furthermore, the embedment of heuristics makes the contribution of GA not as brilliant as expected. For example, the initialization of population still rely on simple heuristics and BWSPT is used for sequence the batches in ΣCi. commonly.

To the best of our knowledge, no other intelligent algorithms are introduced in this kind of issue except those achievements made by our lab. The key point of this kind of problems is the Two-Dimensional characteristics. Therefore, the encoding scheme would determine the quality of the solutions. Appropriate encoding methods, along with other intelligent algorithms such as ant colony algorithms, particle swarm algorithms, neural nets and fuzzy logic, would be possible to find better solutions to the problems.

CONCLUSION

This study reviews research works on batch scheduling problems especially on the Two-Dimensional scheduling problems where jobs have processing time and size constraints. The problem discussed in our study is known to be NP hard. For the NP-hard problems, the number of studies on the design of branch and bound and approximation algorithms is limited.

ACKNOWLEDGMENT

This study was supported by the technology of video stag analysis for Internet regulation, under the Natural Science Foundation of Fuyang Teachers college (2012FSKJ10) and Dynamic optimization of multi-agent networked systems in the complex environment, under the Anhui province natural science foundation (1208085MF111).

REFERENCES

Ali, H.K., K. Behrooz and J. Fariborz, 2006. Minimizing Makespan on a single batch processing machine with non-identical job sizes: A hybrid genetic approach. In: Gottlieb, J. and G.R. Raidl (Eds.), Proceedings of the 6th European Conference on Evolutionary Computation in Combinatorial Optimization, Springer-Verlag Berlin, pp: 135-146.

Ali, H.K., K. Behrooz and J. Masoud, 2008. A hybrid genetic heuristic for scheduling parallel batch processing machines with arbitrary job sizes. Comp. Oper. Res., 35(4): 1084-1098.

Azizoglu, M. and S. Webster, 2000. Scheduling a batch processing machine with non-identical job sizes. Int. J. Prod. Res., 38(10): 2173-2184.

Balasubramanian, H., L. Monch, J. Fowler and M. Pfund, 2004. Genetic algorithm based scheduling of parallel batch machines with incompatible job families to minimize total weighted tardiness. Int. J. Prod. Res., 42: 1621-1638.

Bean, J.C., 1994. Genetic algorithms and random keys for sequencing and optimization. ORSA J. Comp., 6: 154-160.

Chandru, V., C.Y. Lee and R. Uzsoy, 1993a. Minimizing total completion time on batch processing machines. Int. J. Prod. Res., 31: 2097-2121.

Chandru, V., C.Y. Lee and R. Uzsoy, 1993b. Minimizing total completion time on a batch processing machine with job families. Oper. Res. Lett., 13: 61-65.

Chandra, P. and S. Gupta, 1997. Managing batch processors to reduce lead time in a semiconductor packaging line. Int. J. Prod. Res., 35: 611-633.

Chris, N.P. and M.Y. Kovalyov, 2000. Scheduling with batching: A review. Europ. J. Oper. Res., 120: 228-249.

Chung, S.H., Y.T. Tai and W.L. Pearn, 2009. Minimising makespan on parallel batch processing machines with non-identical ready time and arbitrary job sizes. Int. J. Prod. Res., 47(18): 5109-5128.

Coffman, J.E.G., M.R. Garey and D.S. Johnson, 19840. Approximation Algorithms for Bin Packing-an Updated Survey. In: Ausiello, G. *et al*. (Eds.), Algorithms Design for Computer System Design. Springer-Verlag, New York, pp: 49-106.

Damodaran, P. and M.C. Velez-Gallego, 2010. Heuristics for makespan minimization on parallel batch processing machines with unequal job ready times. Int. J. Adv. Manuf. Technol., 49(9): 1119-1128.

Dobson, G. and R.S. Nambimadom, 2001. The batch loading and scheduling problem. Oper. Res., 49: 52-65.

Dupont, L. and C. Dhaenens-Flipo, 2002. Minimizing the makespan on a batch machine with non-identical job sizes: An exact procedure. Comp. Oper. Res., 29: 807-819.

Gilmore, P.C. and R.E. Gomory, 1965. Multistage cutting problems of two and more dimensions. Oper. Res., 13: 94-119.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co. Inc., Boston, MA.

Hochbaum, D.S. and D. Landy, 1997. Scheduling semiconductor burn-in operations to minimize total flow time. Oper. Res., 45: 874-885.

Ikura, Y. and M. Gimple, 1986. Efficient scheduling algorithms for a single batch processing machine. Oper. Res. Lett., 5: 61-65.

Jolai, G.F. and L. Dupont, 1998. Minimizing mean flow time on a single batch processing machine with non-identical job sizes. Int. J. Prod. Econ., 55(3): 273-280.

Kashan, A.H., B. Karimi and F. Jolai, 2006. Minimizing makespan on a single batch processing machine with non-identical job sizes: A hybrid genetic approach. Proceeding of 6th European conference on Evolutionary Computation in Combinatorial Optimization, Springer-Verlag Berlin, pp: 135-146.

Kashan, A.H., B. Karimi and M. Jenabi, 2008. A hybrid genetic heuristic for scheduling parallel batch processing machines with arbitrary job sizes. Comp. Oper. Res., 35: 1084-1098.

Kashan, A.H., B. Karimi and F. Jolai, 2010. An effective hybrid multi-objective genetic algorithm for bi-criteria scheduling on a single batch processing machine with non-identical job sizes. Eng. Appl. Artif. Intell., 23(6): 911-922.

Kempf, K.G., R. Uzsoy and C.S. Wang, 1998. Scheduling a single batch processing machine with secondary resource constraints. J. Manuf. Syst., 17: 37-51.

Kim, H.U., Y.D. Kim and J.G. Kim, 2000. Batching and scheduling at a batch processing workstation in a semiconductor fabrication facility producing multiple product types with distinct due dates. Proceeding of IC Modelling and Analysis of Semiconductor Manufacturing, pp: 151-156.

Koksalan, M. and A.B. Keha, 2003. Using genetic algorithms for single machine bicriteria scheduling problems. Europ. J. Oper. Res., 145(3): 543-556.

Lee, C.Y. and R. Uzsoy, 1999. Minimizing makespan on a single batch processing machine with dynamic job arrivals. Int. J. Prod. Res., 37: 219-236.

Lee, C.Y., R. Uzsoy and L.A. Martin-Vega, 1992. Efficient algorithms for scheduling semiconductor burn-in operations. Oper. Res., 40: 764-765.

Lingfa, L. and Y. Jinjiang, 2008. Unbounded parallel batch scheduling with job delivery to minimize makespan. Oper. Res. Lett., 36(4): 477-480.

Malve, S. and R. Uzsoy, 2007. A genetic algorithm for minimizing maximum lateness on parallel identical batch processing machines with dynamic job arrivals and incompatible job-families. Comp. Oper. Res., 34: 3016-3028.

Mathirajan, M. and A.I. Sivakumar, 2006. A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. Int. J. Adv. Manuf. Technol., 29: 990-1001.

Melouk, S., P. Damodaran and P.Y. Chang, 2004. Minimizing make span for single machine batch processing with non-identical job sizes using simulated annealing. Int. J. Prod. Europ., 87: 141-147.

Monch, L., H. Balasubramanian, J.W. Fowler and M.E. Pfund, 2005. Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times. IIE Trans., 32: 2731-2750.

Parsa, N.R., B. Karimi and A.H. Kashan, 2010. A branch and price algorithm to minimize makespan on a single batch processing machine with non-identical job sizes. Comp. Oper. Res., 37(10): 1720-1730.

Perez, I.C., J.W. Fowler and W.M. Carlyle, 2005. Minimizing total weighted tardiness on a single batch process machine with incompatible job-families. Comp. Oper. Res., 32: 327-341.

Purushothaman, D., K.M. Praveen and S. Krishnaswami, 2006. Minimizing makespan on a batch-processing machine with non-identical job sizes using genetic algorithms. Int. J. Prod. Europ., 103: 882-891.

Reeves, T., 1997. Evaluating What Really Matters in Computer-Based Education, Retrieved from: http:// books. google. com. pk/ books?id = YiOW x1x OD JoC & pg = PA96&dq = Evaluating + What + Really + Matters + in + Computerased + Education & hl = en & sa = X & ei = Us MA Udrl G Mbm4QSN7I Gg DA & ved = 0 CE Q Quw Uw BA # v = onepage & q = Evaluating % 20 What % 20 Really % 20 Matters % 20 in % 20 Computer-Based % 20 Education&.

Sabouni, M., F. Jolai and A. Mansouri, 2010. Heuristics for minimizing total completion time and maximum lateness on identical parallel machines with setup times. J. Int. Manuf., 21(4): 439-449.

Sevaux, M. and S.D. Peres, 2003. Genetic algorithms to minimize the weighted number of late jobs on a single machine. Europ. J. Oper. Res., 151: 296-306.

Shuguang, L., L. Guojun, W. Xiaoli and L. Qiming, 2005. Minimizing makespan on a single batching machine with release times and non-identical job sizes. Oper. Res. Lett., 33: 157-164.

Uzsoy, R., 1994. Scheduling a single batch processing machine with non-identical job sizes. Int. J. Prod. Res., 32: 1615-1635.

Uzsoy, R., 1995. Scheduling batch processing machines with incompatible job families. Int. J. Prod. Res., 33: 2685-2708.

Wang, C. and R. Uzsoy, 2002. A genetic algorithm to minimize maximum lateness on a batch processing machine. Comp. Oper, Res., 29: 1621-1640.

Wang, H.M. and F.D. Chou, 2010. Solving the parallel batch-processing machines with different release times, job sizes and capacity limits by metaheuristics. Exp. Syst. Appl., 37(2): 1510-1521.