An Evolutionary Approach to Cyclic Real World Scheduling

Dominik Żelazny

An evolutionary approach to cyclic real world scheduling

Dominik Żelazny

Department of Automatic Control and Mechatronics,
Wrocław University of Science and Technology,
Janiszewskiego 11-17, 50-372 Wrocław, Poland dominik.zelazny@pwr.wroc.pl

Abstract. Real world problems are often an inspiration for engineers to model and solve new scheduling problems. In this paper a cyclic scheduling of jobs with uncertain data is considered. A minimization of the cycle time represents an important economical factor considered by the companies. Proposed model was tested and data concerning processing times was obtained. Due to the nature of stations and human operator factor, we deal with uncertain data modeled using fuzzy numbers. For the modeled production station a fuzzy genetic algorithm was proposed and tested against deterministic algorithms. Proposed algorithm outperformed all deterministic algorithms.

Keywords: flow shop problem · cyclic scheduling · real world problem · nature-based algorithm.

1 Introduction

Maintaining competitive position in fast changing market requires companies to use new methods of optimization and drives scientists to develop more efficient algorithms. Due to that competitiveness, developing effective, advanced methods is extremely important. The so-called permutation flow shop scheduling problem (PFSP) represents a class of widely studied cases based on ideas derived from production engineering.

Most of the currently researched problems consider classic criterion and data, thus are easily adaptable to real world applications, but modern scheduling problems need more advanced models. This resulted in companies switching to cyclic scheduling models, where different products are manufactured as a *Minimal Part Set – MPS*.The most common optimization of such a system is minimization of the cycle time – the time between the beginning of one MPS and the beginning of the next one.

More complex models include uncertain data, which can represent systems with tasks performed by human operators or in conditions which don't guarantee deterministic processing times. Those factors add uncertain data to the analyzed systems.

In this paper a cyclic scheduling problem with uncertain data is considered. Moreover, a miniature production line (described in detail in[6]) was used to perform tests and model the problem. Fuzzy numbers are employed to identify and model the uncertain data. A meta-heuristic algorithm was developed to obtain production schedules for proposed model. The results are compared with deterministic algorithms.

2 Literature overview

Here we present a brief overview of research considering cyclic job scheduling and fuzzy scheduling problems. We start with cyclic scheduling which received considerable attention. In paper [7] theoretical and numerical properties of methods for obtaining the minimal cycle time are studied. The cycle time is estimated through several expressions which are then compared based on their convergence speed. The distribution of the number of MPS needed to ascertain the cycle time is examined, with 3 to 4 MPS being enough almost 98% of the time. In paper [2] graph model of the problem is employed to establish block elimination properties. Conducted research indicate improvement of the efficiency of the search process for cyclic flow shop problem. The solving methods using parallel computing have also appeared, for example, in paper [1] two parallel techniques – vector processing and multi-walk method – were employed to solve cyclic flexible job shop scheduling problem (CFJSSP). New method of computing the cycle time was also presented.

3 Problem description

A miniature production line is considered (see 1). The model consists of five separate machines. The flow of every detail through the system can be divided into separate phases called stations. Station A consists of transportation module, production line (A1) and a pneumatic arm (A2), which transports the details from station A to station B. Station B consists of four steps: B1) transportation module, which is supported by a human operator, B2) lift, which transports the detail to B3) measurement station, from where detail is taken by B4) transportation module, which moves it to station C. Station C is our main station, where details are processed. First, they are transported by module C1, then they are processed on C2, transported by module C3 and processed for the second time on C4. At last, details are transported by module C5 to station D. First, details are transported using pneumatic arm (module D1) to module the inclined slope (module D2), which transports them to station E. Last station consists of two modules, first (E1) transports details to one of the three existing gates, while the second moves the detail from the gate to the chosen exit.

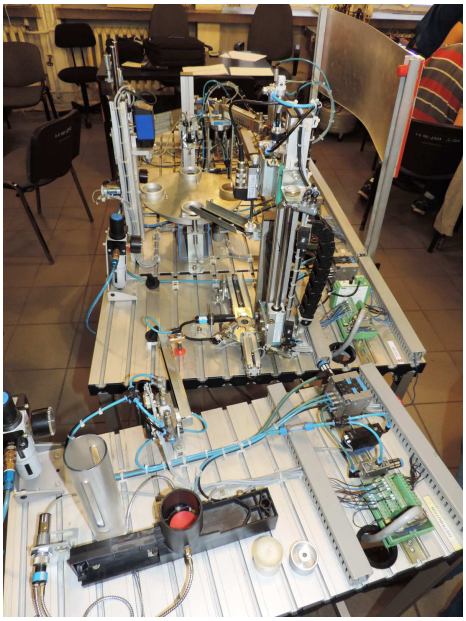

Fig. 1. Model of a production line

4 Mathematical model

Described system can be identified as a cyclic permutation flow shop scheduling problem (CPFSP). This is the manufacturing system with sequential structure of service stages (machines). In this system following elements are defined:

- $\mathcal{M} = 1, 2, \ldots, m$ – set of machines,
- $\mathcal{J} = 1, 2, \ldots, n$ – set of jobs (details),
- $p_{i,j}$ – processing time of job j on machine i,
- $s_{i,j,k}$ – refitting time of machine i from job j to job k,
- $t_{i,j}$ – transportation time of job j from machine i to machine $i + 1$,

where $i = 1, 2, \ldots, m$, $j, k = 1, 2, \ldots, n$.

In the modeled production system following processing steps are distinguished: physical measurement (step B3), first processing (step C2), second processing (step C4) and sorting of the details (steps E1 and E2). Operation processing times p_i depend on the detail to be processed. Between each pair of subsequent machines i and $i + 1$ exists a non-zero transportation time. Refitting takes place only before first and second processing (steps C2 and C4) and depends on the details to be processed before and after the fitting.

A flow shop production system with 4 machines $\mathcal{M} = \{1, 2, 3, 4\}$ is defined. Jobs from set $\mathcal{J}$, are to be processed in non-zero processing times. Set of operations is defined as $\mathcal{O} = \{1, 2, \ldots, 4n\}$ and each

job j consists of 4 operations $(\mathcal{O}_{1,j}, \ldots, \mathcal{O}_{4,j})$ processed on subsequent machines. Each machine can process only one job at the time and each job can be processed only at one machine at the same time. Moreover, interruption of processing of a operation is not allowed.

In cycle x all operations $\mathcal{O}$ for the x-th MPS must be performed. Moreover, this set can be decomposed into non-empty subsets $\mathcal{O}_k$, $k \in \{1, 2, \ldots, m\}$, containing operations from one machine k.

Schedule of jobs can be defined as a permutation $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$, where $\pi(i)$ is an i-th job from permutation π

Let $[S^x]_{m \times n}$ be a matrix of starting times of jobs on x-th MPS, where $S^x_{i,j}$ is a starting time of job j on machine i. Lets assume that the timetable of the system is fully cyclic. As such, there exists a constant $T(\pi)$ called period, such as:

$$S^{x+1}_{i,\pi(j)} = S^x_{i,\pi(j)} + T(\pi), \tag{1}$$

where $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ and $x \in \mathbf{N}$. Period $T(\pi)$ depends on permutation π and is called *cycle time*. Minimal value of $T(\pi)$ will be called *minimal cycle time* and will be denoted as $T^*(\pi)$. Optimizations goal is to minimize following function:

$$T^*(\pi^*) = \min\{T^*(\pi) : \pi \in \Phi\}, \tag{2}$$

Without the loss of generality, we can assume that start and completion times of the first details on first machines are as follows: $S_{1,\pi(1)} = 0$ and $C_{1,\pi(1)} = p_{1,\pi 1}$. Completion times of details on the machines are calculated using the following recursive equation:

$$\begin{aligned} C_{i,\pi(j)} = \max\{&C_{i,\pi(j-1)} + s_{i,\pi(j),\pi(k)}, C_{i-1,\pi(j)} \\ &+ t_{i-1,\pi(j)}\} + p_{i,\pi(j)}, \end{aligned} \tag{3}$$

where $i = 1, \ldots, m$, $j = 1, \ldots, n$ and $k = 1, \ldots, n$.

Goal is to optimize following objective function:

$$T(\pi) = \max_{i \in \mathcal{M}} \{C^x_{i,\pi(n)} - S^x_{i,\pi(1)}\}. \tag{4}$$

5 Uncertain data – fuzzy numbers

Fuzzy numbers [5] are a specific case of fuzzy sets [8] – which are defined through membership function $f : \mathbf{R} \to [0, 1]$ – and can be viewed as generalization of the real numbers. One of the most common membership functions defines fuzzy number x as a triple (a, b, c), where $a, b, c \in \mathbf{R}$, resulting in triangular visualization.

Lets consider step B4 for some job j. The processing time of that step for job j was measured a number of times (12 in this case), creating a vector $\bar{p_j}$ with 12

Table 1. Results of measurement of $\bar{p}$ for step B4 (in seconds)

$p_{1,j}$	$p_{2,j}$	$p_{3,j}$	$p_{4,j}$	$p_{5,j}$	$p_{6,j}$	$p_{7,j}$	$p_{8,j}$	$p_{9,j}$	$p_{10,j}$	$p_{11,j}$	$p_{12,j}$
1.48	7.20	4.72	1.56	1.20	6.24	2.92	4.76	1.68	2.64	2.48	2.72

Table 2. Fuzzy numbers obtained for the proposed production system

Fuzzy number	a	b	c
C^5	3.44	3.44	3.45
D^1	17.52	27.20	47.52
D^2	0.69	1.51	2.92
$t_{0,j}$	13.76	17.61	24.44
$t_{1,j}$	4.64	6.75	10.65
$t_{2,j}$	3.44	3.44	3.45
$t_{3,j}$	21.65	32.15	53.89
$p_{1,j}$	$2.32 \cdot \gamma_j$	$2.79 \cdot \gamma_j$	$3.44 \cdot \gamma_j$
$p_{2,j}$	$1.40 + \alpha_j$	$1.40 + \alpha_j$	$1.40 + \alpha_j$
$p_{3,j}$	$1.40 + \beta_j$	$1.40 + \beta_j$	$1.40 + \beta_j$
$p^A_{4,j}$	1.40	2.71	5.04
$p^B_{4,j}$	2.44	4.03	7.32
$p^C_{4,j}$	3.32	4.36	7.84

elements as shown in Table 1. The estimated processing time for step B4 can be now modeled using fuzzy number $B^4 = (B^4_a, B^4_b, B^4_c)$ as follows:

$$B^4_a = \min(p_{1,j}, p_{2,j}, \ldots, p_{11,j}) = 1.20, \tag{5}$$

$$B^4_b = \frac{p_{1,j} + p_{2,j} + \cdots + p_{11,j}}{12} = 3.30, \tag{6}$$

$$B^4_c = \max(p_{1,j}, p_{2,j}, \ldots, p_{11,j}) = 7.20, \tag{7}$$

yielding $B^4 = (1.20, 3.30, 7.20)$.

The above procedure can be applied to all steps in the production process and then be used to define all processing and transport times through fuzzy numbers. For example, transport time $t_{3,j}$ is the sum of times of steps C5, D1, D2. Thus, it can be represented as a fuzzy number computed as a sum of 3 fuzzy numbers *i.e.* $t_{3,j} = C^5 + D^1 + D^2$. The resulting fuzzy numbers are presented in Table 2. Transport times are identical for all jobs. Fuzzy number $p_{i,j}$ represents fuzzy processing time of job j on machine i. This time is dependent on the scaling parameter γ_j which is different for various jobs. It is easily observed that the uncertainty in the considered system is significant, as the ratio $\frac{c}{a}$ can be as high as 2 or 3 for some of the presented fuzzy numbers.

6 Proposed method

For the purpose of this article an evolutionary method was proposed. Parameters of the algorithm were automatically adjusted (self-set parameters). Moreover, a second stop rule was implemented. When execution reaches certain (predetermined by tests) run time, the algorithm stops at current iteration.

6.1 Genetic Algorithm

Genetic Algorithm (GA) is a multi-agent method based on the evolution process found in the nature to find better solutions. Evolutionary algorithms use techniques inspired by natural occurring factors, such as inheritance, mutation, selection and crossover to generate solutions to optimization problems. Usually, the evolution starts from random initial population, which consists of individuals. In each iteration, called generation, those individuals are modified (by means of mutation and/or crossover) and their fitness is evaluated in order to select best solutions for next generation. Over the years, different approaches to the GA were proposed and tested for a variety of optimization problems. Our GA uses external Pareto archive in order to maintain non-dominated solutions through successive iterations. The individuals in population are represented by the following: jobs permutation, values of criteria functions and relative closeness indicator calculated by the TOPSIS method. Initial population includes solutions obtained from certain constructive algorithms, prepared to optimize one of the criteria. Such initialization allows faster designation of the approximation of Pareto front. Mutation is performed by interchanging two random jobs in schedule, while crossover uses a partially matched crossover (PMX) scheme. Fitness values are evaluated using the TOPSIS technique and are then used in tournament selection. After the selection, half of parent and child population is combined into new parent population. Moreover, when relative closeness values converge to zero, an anti-stagnation function is employed. Similar genetic algorithms have been proposed in [4, 3].

7 Computer experiment

All algorithms were implemented in C++ and compiled with *Embarcadero C++ Builder XE7 Professional*. The programs were tested on Intel® Core i7-3770 3400MHz machine (8 concurrent threads) with 8GB of RAM under the *Microsoft Windows 8.1 64b* operating system. In order to compare the results, 3 deterministic algorithms were implemented, resulting in tests of following algorithms:

1. F – fuzzy algorithm. Based on the fuzzy problem instances with times in the form of fuzzy numbers *e.g.* $\{a, b, c\}$.
2. D_{MIN} – deterministic algorithm working on real numbers. Numbers are generated from the fuzzy instance using a value of fuzzy number.
3. D_{MID} – deterministic algorithm. Numbers are generated from the fuzzy instance using b value of fuzzy number.

4. D_{MAX} – deterministic algorithm. Numbers are generated from the fuzzy instance using c value of fuzzy number

Table 3. Summary of results, instance sizes 10-25

Instance	Algorithm	Best run [%]	Average run [%]	Worst run [%]
10	F	100,00	113,91	144,43
	D_{MIN}	220,51	256,44	306,31
	D_{MID}	200,64	234,50	283,59
	D_{MAX}	202,76	234,76	278,15
15	F	100,00	113,14	148,71
	D_{MIN}	256,75	298,16	354,26
	D_{MID}	188,42	219,84	266,55
	D_{MAX}	213,44	244,54	289,41
20	F	100,00	114,58	139,31
	D_{MIN}	284,74	315,46	357,21
	D_{MID}	185,23	212,78	247,60
	D_{MAX}	211,88	241,71	278,66
25	F	100,00	124,36	155,24
	D_{MIN}	280,57	311,85	351,78
	D_{MID}	203,25	228,54	261,83
	D_{MAX}	215,68	244,37	280,67

It can be easily observed, that fuzzy genetic algorithm (henceforth called FGA) outperformed deterministic GAs considerably. In every test, best run of the fuzzy version had best results from all of the algorithms. Its average run was better in most cases than the best runs of deterministic algorithms. Moreover, worst run of the fuzzy algorithm was better than best runs of D_{MIN} and D_{MAX} algorithms and, in some cases, D_{MID} deterministic algorithm.

There are 8 groups of 5 instances each, yielding 40 instances of problems for proposed problem. Groups are dependant on the number of jobs to be performed, starting from 10 up to 45. Proposed algorithm, as well as deterministic versions of GA, were tested 10 times per instance. The summarized results of first 4 groups are shown in Tab. 3 and last 4 groups in Tab.4.

8 Conclusions and further research

In this paper a fuzzy genetic algorithm for a real world model of cyclic scheduling problem was proposed. The mathematical model of the problem was based on the cyclic flow shop scheduling system with uncertain (fuzzy) data. Algorithms, fuzzy

Table 4. Summary of results, instance sizes 30-45

Instance	Algorithm	Best run	Average run	Worst run
30	F	100,00	136,01	182,08
	D_{MIN}	348,17	386,79	437,72
	D_{MID}	201,77	232,37	270,41
	D_{MAX}	272,22	308,10	354,31
35	F	100,00	149,14	204,07
	D_{MIN}	352,34	388,99	437,76
	D_{MID}	138,15	177,69	227,37
	D_{MAX}	252,77	289,37	332,77
40	F	100,00	128,40	174,28
	D_{MIN}	346,05	381,01	425,99
	D_{MID}	115,62	146,73	190,44
	D_{MAX}	220,48	262,26	311,40
45	F	100,00	117,05	144,44
	D_{MIN}	276,09	297,31	325,59
	D_{MID}	105,11	129,60	156,19
	D_{MAX}	199,96	217,35	240,55

and deterministic, were tested using instances with number of jobs ranging from 10 to 45. The results indicate that the proposed FGA significantly outperforms the deterministic algorithms. Each algorithm was tested under the same test conditions (maximal worktime and maximal number of iterations), so that the results were not affected by differences in test procedure.

Acknowledgements

This work is co-financed by the Młoda Kadra project, B50304.

References

1. Bożejko W., Pempera J., Wodecki M., Parallel Simulated Annealing Algorithm for Cyclic Flexible Job Shop Scheduling Problem, Lecture Notes in Computer Science (9120), pp 603–612 (2015)
2. Bożejko W., Uchroński M., Wodecki M., Block approach to the cyclic flow shop scheduling, Computers & Industrial Engineering (81), pp 158–166 (2015)
3. Bożejko W., Kacprzak Ł., Wodecki M., Parallel Coevolutionary Algorithm for Three-Dimensional Bin Packing Problem, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science (9119), pp 319–328 (2015)

4. Bożejko W., Kacprzak Ł., Wodecki M., Parallel packing procedure for three dimensional bin packing problem, proceedings of 20th International Conference on Methods and Models in Automation and Robotics (MMAR), pp 1122–1126 (2015)
5. Dubois D., Prade H., Operations on fuzzy numbers, International Journal of Systems Science (9) 6, pp 613–626 (1978)
6. Rudy J., Cyclic Scheduling Line with Uncertain Data, Lecture Notes in Artificial Intelligence (9692), pp 1–10 (2016)
7. Smutnicki C., An efficient algorithm for finding minimal cycle time in cyclic job shop scheduling problem, 16th International Conference on Intelligent Engineering Systems, pp 381–386 (2012)
8. Zadeh L. A., Fuzzy sets, Information and Control (8) 3, pp 338–353 (1965)