Integrated Preventive Maintenance Scheduling Model with Redundancy for Cutting Tools on a Single Machine

Leyla Sadat Tavassoli
Department of Industrial Manufacturing
and Systems Engineering
University of Texas at Arlington
Arlington, TX, USA
leylasadat.tavassoli@mavs.uta.edu

Nahal Sakhavand
Department of Industrial Manufacturing
and Systems Engineering
University of Texas at Arlington
Arlington, TX, USA
nahal.sakhavand@mavs.uta.edu

Seyed Sajjad Fazeli
Department of Industrial and System Engineering
Wayne State University
Detroit, MI, USA
sajjad.fazeli@wayne.edu

***Abstract*-In this paper, we present an integrated multi-objective framework of a single machine for a single cutting tool problem. Our maintenance policy is based on performing minimal repairs in case of a minor failure and Preventive Maintenance (PM) to avoid a major failure that results in the replacement of the tool. This framework allows simultaneous optimization of the two conflicting time and cost objectives. A redundant system is proposed as a part of the model to assist the production line under a major failure. In addition, the tool's preventive maintenance time is synchronized with the completion of the machine tool's work cycle to reduce the machine's set-up time. The model was optimized using a customized Non-dominated Sorting Genetic Algorithm (NSGA-II). An experimental study based on real-market data was conducted and the results were compared with the ones obtained from classical methods.**

Keywords-maintenance scheduling; preventive maintenance; redundancy; non-dominated sorting genetic algorithm; multi-objective optimization; time and cost trade-off

I. Introduction

Addressing the trade-off between time and cost brings forth several challenges. Notably, any change in the time schedule has an impact on project cost and ultimate profit. This is reflected in the efficiency of the tools and machines in the production line. Cutting tools are widely used in almost all lines of industrial systems. However, due to their constant use they are subjected to gradual depreciation and/or severe damages. These complications attribute to the costs of storage and procurement as well as the production disruption during operation. In order to reconcile these challenges, it is necessary to consider the system characteristics before implementing the production process. In a manufacturing system, all the equipment parts, including machines and tools, have an estimated lifetime. In order to prevent the line from going idle as a result of an unexpected minor or major breakdown of a piece of equipment, a common approach is to incorporate Preventive Maintenance (PM) as proposed in [1, 2] using linear programming algorithms. PM is regularly performed to decreasethe likelihood of system failure under two alternatives. A repair action allows the equipmentto be operable again or a replacement to a new tool. For a comprehensive review on integrated maintenance and production planning models, we refer the enthusiastic reader to [3]. In general, there is a possibility of an excessive amount of time to repair a tool in addition to the fact that replacement with a new element might not always be to the best profit interest. Therefore, PM actions alone may not be as effective in planning decisions. Another technique to improve system performance, is to use redundancy in the processing operation [4]. The redundant system is designed to automatically take over the equipment operation if a failure occurs. Some works have considered a joint redundancy and maintenance strategy optimization [5]. This model determines the optimal redundancy levels in addition to the minimal repair simultaneously. While there are clear advantages of incorporating PM and redundant systems (e.g. reduction in time and cost of failures), these systems can create other difficulties such as extra operational costs.

In an optimal PM scheduling process, there is an essential role in resource and planning management to meet the conflicting user-driven requirements. The objective of such a system is to minimize the time between the essential PM actions and the corresponding cost. Although in a competitive environment minimizing time and cost is less likely to happen simultaneously, an optimal trade-off between time and cost is

Corresponding author: Leyla Sadat Tavassoli achievable. Time and cost trade-off models have long been addressed using traditional exact optimization algorithms where the cost function is linear [6–9] or nonlinear [10]. In several works, these problems have been formulated using dynamic programming [11]. However, given the NP-hard nature of the trade-off problems, using exact algorithms may become computationally intractable for large-size nonlinear problems. In order to improve the computational performance over the exact algorithms, several works suggest near-optimal heuristics such as Genetic Algorithms (GA) [12-14], Simulated Annealing (SA) [15], and Ant Colony Optimization (ACO) [16].

In all of these models, the problem is considered as a single objective time and cost trade-off. Alternatively, time and cost trade-off problems can be formulated as multi-objective models. In conventional multi-objective optimization problems, weighted LP and NLP algorithms yield a single optimal point based on the random or fixed weights [17, 18]. In single-objective optimization models and multi-objective problems, the endeavor is to find a set of optimal solutions for conflicting objectives which are not necessarily superior corresponding to all objectives, called the non-dominated Pareto optimal solutions [19-21]. Utilizing evolutionary algorithms such as Non-dominated Sorting Genetic Algorithm (NSGA) [22-25], Multi-Colony ACO (MOACO) [26], and Multi-Objective Particle Swarm Optimization (MOPSO) [27, 28], establishes a set of solutions which are subject to decision-makers' selection from the optimum. Although single-objective maintenance scheduling problems have been broadly addressed in the literature, the number of studies on multi-objective maintenance problems is quite limited. Authors in [29] use goal programming to solve the PM and replacement scheduling problem while suggesting heuristic solution procedures for large-scale problems. For large-scale multi-objective maintenance scheduling problems, NSGA is commonly used in search heuristics to optimize these problems based on operators of biologically inspired principles like mutation and crossover. Using GA, we can obtain optimal maintenance actions for the multi-objective maintenance scheduling problem of deteriorating structures. Authors in [14] proposed a bi-objective model for electromechanical products where they considered soft and hard failure as well as an imperfect PM approach. Their goal was to maximize the availability and minimize the cost of the system. In terms of solution methodology, they considered NSGA-II with the single mutation and crossover operators.

Compared with the literature, the main contributions of this paper are:

- We propose an integrated multi-objective preventive maintenance scheduling for a cutting tool system on a single machine with redundancy that captures the existing time and cost trade-off. Our model inherits the merits of providing a more realistic set of solutions which are tested for a large-scale case study.
- Our model provides a well-suited customized redundancy option to decide which tools require a redundant system to assist the production line in addition when conducting PM actions.
- We present an innovative customized crossover and mutation operators in the NSGA-II algorithm and we contrast our set of Pareto front solutions with conventional NSGA-II algorithms. Our numerical results attest the faster convergence of our modified NSGA-II algorithm over the techniques used in the literature.

II. PROBLEM DESCRIPTION

In order to formulate our scheduling problem, we begin by stating the following assumptions.

A. Minor Failure Problem

Let us consider an operating cutting tool assembled on a machine. As soon as a minor failure occurs, we perform minimal repair so that the tool is in an operable status again. In this case the tool's useful life does not change. Generally, tools' life deteriorates with time which necessitates regular tools inspection to replace the tool or perform a minimal repair. To address this, we schedule reoccurring PM actions to avoid unpredictable failures which affect the efficiency of the production line. In this case, the tool's age is reset. In our problem, as a part of the optimization process, we would like to minimize the process time of PM actions which leads to an optimized cost and time of the process.

B. Major Failure Problem

From time to time, cutting tools are subjected to major failures in the middle of a cycle. These failures are caused by either human error or a tool life depreciation which leads to complete tool breakage. In order to keep the production line active, a new replacement tool needs to be available. However, inventory and holding equipment is costly. Therefore, in addition to scheduling PM actions in a timely manner, including redundant systems can have a significant impact on a cost-effective and time-saving system performance.

C. Setup Time

PM actions implementation entails stopping the machine tool and extracting the cutting tool out of it. If PM actions do not take place at the end of the machine's cycle time, we have to interrupt the machine in the middle of the cutting operation. After PM actions are performed, the machine requires to be set up again. We define a penalty indicator for the setup time in the middle of the production process which we tend to minimize in our objective function.

D. The Integrated Problem

In the cutting process, in addition to performing PM actions in case of a minor failure, we consider a redundant system in the form of a backup when a major failure occurs. Moreover, as a part of our objective, we minimize the setup time after each process disruption. In the next section, the full model formulation is provided.

III. PROBLEM FORMULATION

Our formulation is inspired by [30]. The failure rates for minor and major failures are determined as:

$$r_1(y) = \frac{f_1(y)}{\int_y^{\infty} f_1(y)} \quad (1a)$$

$$r_2(y) = \frac{f_2(y)}{\int_y^{\infty} f_2(y)} \quad (1b)$$

where $f_1(y)$ and $f_2(y)$ are the probability density functions for minor and major failure variables.

Let z_t , $t \in T$ be 1 if PM actions are performed before time t and 0 otherwise. We can formulate the effective age of the cutting tool as:

$$a_t = (1 - z_j)(a_{t-1} + L) \ \forall t \in T \quad (2)$$

Equation (2) suggests that when z_t = 0, no PM action is applied before time t. Therefore, the tool's current age is its age at the previous time period t-1 in addition to some period length denoted by L. On the other hand, if PM actions are performed on the tool before t, its age is reset and the tool is considered as new. Next, we present the maintenance costs which consist of the expected Cost of Minimal Repair (*CMR*) and the Cost of PM (*CPM*) for each tool.

$$CM_i = \sum_{t=1}^{T} CPM_i \, z_t + CMR_i \int_{a_t}^{a_t+L} r_1(y)dy \quad \forall i \in I \quad (3a)$$

We can immediately write the cycle time for PM actions as:

$$TM_i = \sum_{t=1}^{T} TPM_i \, z_t + TMR_i \int_{a_t}^{a_t+L} r_1(y)dy \quad \forall i \in I \quad (3b)$$

where *TPM* and *TMR* are the process time for PM actions and minimal repair which are considered as continuous variables. If a major failure occurs, the production line will be stopped until the tool is replaced. Most of the equations described here can be applied to every cutting tool $i \in I$. We will explicitly mention when this is not the case. We denote the penalty associated with the major failure by CF1 and define it as:

$$CF_1 = (TIC_m)(\int_{a_t}^{a_t+L} r_2(y)dy) + CT \quad (4a)$$

where C_m is defined as the cost of production of machine tool per time and *TI* and *CT* are the idle time of the machine due to a major failure and cost of replacing a broken tool with a new one respectively. The time it takes for replacing the tool can be computed as:

$$TF_1 = TI \int_{a_t}^{a_t+L} r_2(y)\mathrm{d}y) \quad (4b)$$

We define b to be 1 if the redundant system is applied and 0 otherwise. When redundancy is considered in the process, the major failure occurrence penalty is defined as in (5a) which is the cost of applying a redundant system denoted by *CRS* in addition to *CT*.

$$CF_2 = CRS + CT \quad (5a)$$

In this case, we assume that the process time of replacing the tool is ignored as it does not affect the overall production time as a result of a backup system which results to:

$$TF_2 = 0 \quad (5b)$$

Consequently, (6a) and (6b) construe the cost and time penalty functions respectively. Let T_{set} be the machine setup time after being interrupted in the middle of its cycle time to conduct PM. We note that if the process time of the PM actions is equal to a coefficient of the cycle time of the machine tool, a_t = 0, which results in no time penalty. The cost penalty function can be interpreted in a similar manner.

$$C_{penalty} = \sum_{t=1}^{T} a_t \mathrm{Tset} C_m \quad (6a)$$

$$T_{penalty} = \sum_{t=1}^{T} a_t Tset \quad (6b)$$

where a_t can be computed as:

$$\alpha_t = 1 \Leftrightarrow \ \nexists n \in \mathbb{N}\text{: } n \times \text{Cycle } time \ = \ TPM \quad (7a)$$

$$\alpha_t = 0 \Leftrightarrow \ \exists n \in \mathbb{N}\text{: } n \times \text{Cycle } time \ = \ TPM \quad (7b)$$

Our cost and time objective functions can be formulated as:

$$Min\ Cost = \sum_{i=1}^{I} CM_i + (TM_i C_m) + \beta_i \mathrm{CF}_{1,i} + (1-\beta_i)CF_{2,i} + C_{Penalty,i} \quad (8a)$$

$$Min\ Time = \sum_{i=1}^{I} TM_i + \beta_i CF_{1,i} + (1-\beta_i)TF_{2,i} + T_{Penalty,i} \quad (8b)$$

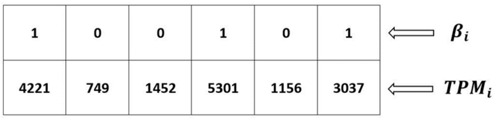
where β_i is binary $\forall i \in I$ and $TPMi \geq 0 \ \forall i \in I$, where I is the set of cutting tools in the process.

IV. Methodology

On a production basis, we often use a lot of cutting tools in the process which affects the decisions about time and cost of PM actions and redundant systems inclusion. To mitigate the computational difficulties of solving the resulting large scale problem due to the non-linear objective functions, we apply a modified NSGA-II heuristic algorithm. We apply these modifications in the crossover and mutation operators which allow a faster convergence. We ultimately assess our solutions by introducing an index penalty in the algorithm.

A. Initialization

Before creating the crossover and mutation operators, we require maintaining a diverse population of the candidate solutions which we refer to as chromosome strings. For starters, we generate 100 random solution alternatives. For each of our two variables of β_i and *TPMi*, we assign a row which is extended for each tool *i*. Figure 1 demonstrates one set out of the randomly generated 100 solutions.



1	0	0	1	0	1	⟸ β_i
4221	749	1452	5301	1156	3037	⟸ TPM_i

Fig. 1. A chromosome string of set of candidate solutions.

B. Mutation

We define 3 mutation operators where we generate the parents using the roulette wheel selection. We design the first mutation operator to change the value of a random β cell picked from its row. In order to address the possibility of finding better optimal solutions whether redundancy is applied or not, the proposed operator switches the values of the cell from 0 to 1 or from 1 to 0. Our second operator picks a cell from the second row of the chromosome string. This operator sums the value of the cell with a random number x in order to avoid selecting local solutions. The third operator is set to select solutions of the second row with the minimum penalty.

C. Crossover

For our experiment, we consider two crossover operators where the first one generates a parent using roulette wheel selection. It then switches two randomly selected parents and creates two offspring. The second operator generates uniformly distributed solutions in the Pareto front. In this approach, the operator chooses consecutive parents from the Pareto front which maximize the crowding distance. We use the following equations to create new solution candidates.

$$C_1' = C_1 + (1-\lambda)(C_2 - C_1) \quad (9a)$$

$$C_2' = C_1 + \lambda(C_2 - C_1) \quad (9b)$$

where $0 \leq \lambda \leq 1$ and C_1' and C_2' are the updated new cell values.

D. Non-dominated Sorting

To sort the solutions, we use the convexity crowding distance method. For every chromosome we have:

$$d = \sum_{i=n}^{N} \frac{g_n(x) - G}{g_n^{max} - g_n^{min}} \quad (10)$$

where $g_n(x)$ is the n^{th} objective function and g_n^{max} and g_n^{min} are the maximum and minimum values of the n^{th} objective function value. G is the closest value obtained to the considered objective value.

E. Penalty Algorithm

As explained above, we consider a penalty for each PM action that requires disturbing the production line in the middle of its cycle. In order to find the number of each solution's penalty α, we conduct a penalty algorithm which is illustrated in Figure 2.

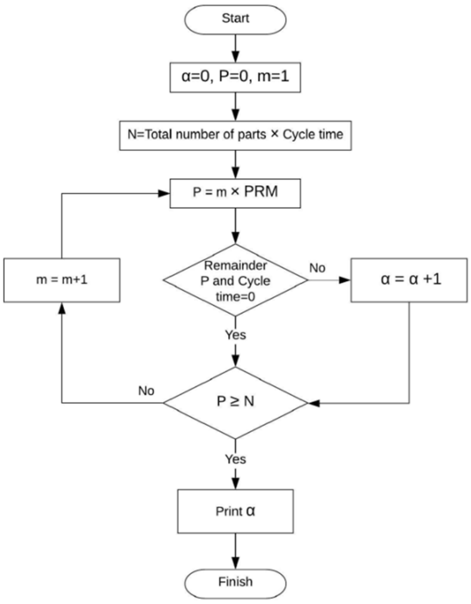


Fig. 2. Penalty function algorithm.

F. Numerical Example

For the computational experiments, we used real data collected from [31]. In our numerical experiment, we consider 6 cutting tools which perform machining operation on an aluminum part. Using a vertical milling machine, it produces 10000 parts with a cost of $2.5 per minute. We consider the idle time of the machine tool in case of a major failure to be one complete shift of 8 hours and the cycle time of the machine tool to be 375 seconds. Figure 1 demonstrates the time and cost of PM for each cutting tool. We note that the cost and time of PM actions are assumed under screw and tool replacement after a breakage.

V. NUMERICAL RESULT

A. Results for Modified-NSGA-II vs. NSGA-II

We initialize the algorithm by randomly generating a population of 100. The number of the initial population and the solutions provided by mutation and crossover differ based on their importance. Table I demonstrates the population for each operator.

TABLE I. PM COST AND TIME RESULTS

	Tool #1	Tool #2	Tool #3	Tool #4	Tool #5	Tool #6
CPR ($)	12	47	14	25	23	17
CMR ($)	12	9	8	7	8	7
TPR (min)	4	5	4	6	4	4
TMR (min)	4	2	2	2	2	2

TABLE II. ALGORITHM SPECIFICATIONS AND POPULATION

Algorithm specification	Number of produced population
Primary population	100
Mutation Type1	30
Mutation Type2	30
Mutation Type3	30
Crossover 1	30
Crossover 2	30

In order to assess the performance of our modified NSGA-II algorithm, we define a reference based on 100 best candidate solutions obtained by the modified NSGA-II, NSGA-II with redundancy, and NSGA-II without redundancy, each replicated 10 times. We conduct the experiment using MATLAB platform on a Windows-based server with 16 GB RAM. Our case study is based on a real data-set. Figure 3 contrasts the Pareto front results for the three algorithms. It can be seen that the modified NSGA-II provides a wider range of non-dominated optimal solutions. To provide better intuition on how the implemented algorithms perform, the detailed results are exhibited in Table III. Each replication was run for 100 iterations. We observe that due to the computational complexity of the modified NSGA-II, it takes relatively more time to solve each replication. Moreover, since we incorporated a mutation operator that seeks candidate solutions with no penalties, there are fewer solutions with penalty for the Modified NSGA-II than for NSGA-II with and without redundancy. Applying our mutation operators select solutions with no penalty in our proposed NSGA-II. However, there is no such operator incorporated to generate these solutions and may randomly select these solutions. We also notice that

NSGA-II with redundancy provides better computational performance confirming the effectiveness of including redundancy in the system. The results of the comparison between the NSGA-II and the modified NSGA-II can be seen in Table III.

TABLE III. RESULTS FOR NSGA-II AND MODIFIED NSGA-II

		Time			**Penalty**			**NNS**		
No.	**Iteration**	**Modified-NSGA2**	**NSGA2**		**Modified-NSGA2**	**NSGA2**		**Modified-NSGA2**	**NSGA2**	
			Without the redundant system	**With the redundant system**		**Without the redundant system**	**With the redundant system**		**Without the redundant system**	**With the redundant system**
1	100	128.6	94.2	86.5	131.4	161.2	178.5	0	19	1
2	100	134.3	87.8	87.2	116.0	176.4	150.6	0	0	6
3	100	127.2	88.1	88.5	74.0	115.5	142.5	13	8	5
4	100	127.0	88.0	88.6	125.8	177.1	167.3	0	1	5
5	100	126.8	100.2	90.0	68.7	171.7	160.3	2	0	0
6	100	128.2	113.2	87.4	75.6	156.3	157.3	31	4	4
7	100	128.2	87.8	87.4	68.2	167.5	120.6	15	0	14
8	100	126.0	86.5	87.3	119.1	121.7	159.4	1	14	7
9	100	129.8	86.0	88.1	92.7	136.2	183.2	7	6	20
10	100	128.0	85.9	87.9	97.2	163.7	156.8	1	0	2
Ave.	-	128.4	91.8	87.9	96.8	154.7	157.6	7	5.2	6.4
Std. Dev.	-	2.20	8.31	0.92	23.44	21.25	16.86	9.59	6.35	5.85
		Inverted Generated Distance			**Delta Index**			**Spacing Metric**		
No.	**Iteration**	**Modified-NSGA2**	**NSGA2**		**Modified-NSGA2**	**NSGA2**		**Modified-NSGA2**	**NSGA2**	
			Without the redundant system	**With the redundant system**		**Without the redundant system**	**With the redundant system**		**Without the redundant system**	**With the redundant system**
1	100	268990	670680	1782100	4618	2499	465	185	796	291
2	100	107210	651130	1772100	4974	3051	478	237	704	278
3	100	82140	656510	1774300	5716	3118	583	332	204	362
4	100	252240	601480	1782100	4606	3368	249	232	1273	83
5	100	130800	675760	1777100	5296	2706	607	258	479	572
6	100	81027	600140	1765000	5600	2221	332	184	695	202
7	100	119160	677110	1785900	5153	3214	257	248	429	143
8	100	177600	639400	1779800	5609	3163	414	241	354	249
9	100	160170	677310	1771300	3978	2551	195	212	515	74
10	100	50396	674010	1784400	6373	3401	442	225	483	396
Ave.	-	142970	652353	1777410	5192	2929	402	235	593	265
Std. Dev.	-	68860	28442	6324	649	384	133	39	281	145

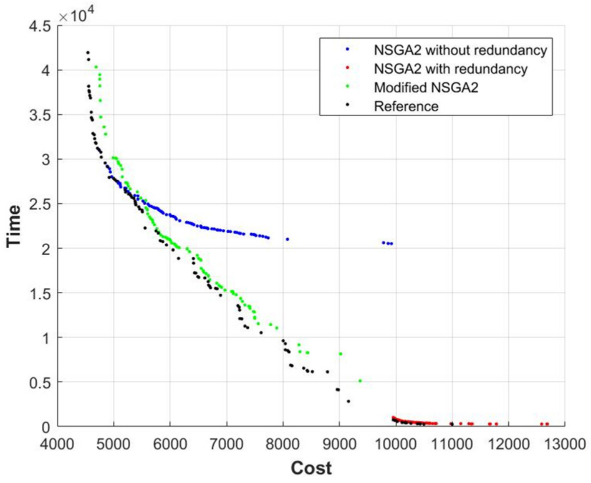


Fig. 3. Optimal Pareto front for NSGA-II vs Modified NSGA-II.

B. Inverted Generated Distance

This approach is designed to evaluate the diversity of the optimal solutions computed by:

$$IGD = \frac{\sum_{v \in P^*} d(v,P)}{|P^*|} \quad (11)$$

where P is a set of non-dominated solutions obtained by the algorithm and P^* is the set of uniformly distributed points in a true Pareto front. $d(v,P)$ is the minimum Euclidean distance between v and the points in P. In Table III, we observe that modified NSGA-II provides far less IGD numbers over the other NSGA-II algorithms across all the replications which prove the significant effectiveness of the algorithm in finding the optimal solutions.

C. Delta Index

In this approach, we examine the distribution of the Pareto front considering the following equation:

$$\Delta(s) = \sum_{i=1}^{|s|-1} \frac{|d_i - \bar{d}|}{|s|-1} \quad (12)$$

where d_i is the Euclidean distance of two consecutive solutions of the Pareto front with s optimal solutions and $\bar{d}$ is the average distance. The less the delta is, the more uniformly the Pareto front will be distributed. However, the range of Pareto front solutions affect the delta index as is demonstrated in Table III. Since the modified NSGA-II selects a wider range of solutions in the Pareto front, its index is more than other reported indices for NSGA-II algorithms.

D. Spacing Metrics

In this technique, we opt for analyzing the uniformity between the Pareto front and the reference solutions which can be computed as in (13):

$$SP(s) = \sqrt{\frac{1}{|S|-1}\sum_{i=1}^{|s|}\left(\bar{d}-d_i\right)^2} \quad (13)$$

where d_i is the minimum distance of optimal solution i from the reference point and $\bar{d}$ is defined as the average distance for s optimal solutions. Notice that the spacing metrics across all the replications for modified NSGA-II are less than NSGA-II with and without redundancy which suggests better uniformity to reference solutions.

VI. Conclusion

In this study, we developed an integrated multi-objective preventive maintenance scheduling model for a single machine cutting tool system with the option of redundancy in one framework. Redundancy performs as a backup and major failure safe system. Our model optimizes the redundancy requirements and the time duration of the PM process. We propose a modified genetic algorithm with customized mutation and crossover operators which allow finding a better Pareto front compared to the existing algorithms. We implemented our experiments on a case study drawn from real data. Our results confirm the better performance of the modified NSGA-II algorithm which suggests that evolutionary algorithms alone do not necessarily provide more viable optimal solutions and require customized operators for better computational efficiency. Moreover, integrating the PM process with redundancy outperforms the conventional models by providing more realistic solutions. Furthermore, defining the appropriate mutation and crossover operators in GA avoids selecting local optima which improves significantly the convergence. Analyzing PM process times provides valuable information to manufacturing companies in order to determine meticulous planning and scheduling decisions. In addition, these industries have the privilege of economic justification for applying redundancy in the production line before implementing the project, not to mention the ability to design, produce and procure their products based on customers' priorities and demand requirements. Utilizing the modified NSGA-II, one can consider more objective functions such as the quality of roughness amount of the cutting surface as well as machine energy consumption or labor requirements in order to provide better scheduling results.

For future work, the incorporation of the PM process and redundancy for multiple tools and machines is considered. In this case, we study a framework which includes multiple machines, each operating with different tools. This will result in error computation for a more complicated system which naturally necessitates defining appropriate mutation and crossover operators associated with the tools and the machines. Another future work could be implementing other evolutionary algorithms such as the NSGA-III to further improve the accuracy and performance of the solutions in large-scale instances.

References

[1] S. P. Canto, "Application of Benders' decomposition to power plant preventive maintenance scheduling," *European Journal of Operational Research*, vol. 184, no. 2, pp. 759–777, Jan. 2008, https://doi.org/10.1016/j.ejor.2006.11.018.

[2] J. A. C. Duarte, J. C. T. A. Craveiro, and T. P. Trigo, "Optimization of the preventive maintenance plan of a series components system," *International Journal of Pressure Vessels and Piping*, vol. 83, no. 4, pp. 244–248, Apr. 2006, https://doi.org/10.1016/j.ijpvp.2006.02.016.

[3] G. Budai, R. Dekker, and R. P. Nicolai, "Maintenance and Production: A Review of Planning Models," Econometric Institute Research Papers, EI 2006-44, 2006. https://doi.org/10.1007/978-1-84800-011-7_13.

[4] T. Sooktip, N. Wattanapongsakorn, and D. W. Coit, "System reliability optimization with k-out-of-n subsystems and changing k," in *The Proceedings of 2011 9th International Conference on Reliability, Maintainability and Safety*, Guiyang, China, Jun. 2011, pp. 1382–1387, https://doi.org/10.1109/ICRMS.2011.5979487.

[5] Y. Liu, H. Huang, Z. Wang, Y. Li, and Y. Yang, "A Joint Redundancy and Imperfect Maintenance Strategy Optimization for Multi-State Systems," *IEEE Transactions on Reliability*, vol. 62, no. 2, pp. 368–378, Jun. 2013, https://doi.org/10.1109/TR.2013.2259193.

[6] D. R. Fulkerson, "A Network Flow Computation for Project Cost Curves," *Management Science*, vol. 7, no. 2, pp. 167–178, 1961.

[7] J. E. Kelley, "Critical-Path Planning and Scheduling: Mathematical Basis," *Operations Research*, vol. 9, no. 3, pp. 296–320, 1961.

[8] Xiaodong Yao, M. Fu, S. I. Marcus, and E. Fernandez-Gaucherand, "Optimization of preventive maintenance scheduling for semiconductor manufacturing systems: models and implementation," in *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01) (Cat. No.01CH37204)*, Sep. 2001, pp. 407–411, https://doi.org/10.1109/CCA.2001.973899.

[9] X. Yao, E. Fernandez-Gaucherand, M. C. Fu, and S. I. Marcus, "Optimal preventive maintenance scheduling in semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 17, no. 3, pp. 345–356, Aug. 2004, https://doi.org/10.1109/TSM.2004.831948.

[10] A. H. Shirmohammadi, Z. G. Zhang, and E. Love, "A Computational Model for Determining the Optimal Preventive Maintenance Policy With Random Breakdowns and Imperfect Repairs," *IEEE Transactions on Reliability*, vol. 56, no. 2, pp. 332–339, Jun. 2007, https://doi.org/10.1109/TR.2007.896747.

[11] R. V. Canfield, "Cost Optimization of Periodic Preventive Maintenance," *IEEE Transactions on Reliability*, vol. 35, no. 1, pp. 78–81, Apr. 1986, https://doi.org/10.1109/TR.1986.4335355.

[12] C. R. Cassady and E. Kutanoglu, "Integrating preventive maintenance planning and production scheduling for a single machine," *IEEE Transactions on Reliability*, vol. 54, no. 2, pp. 304–309, Jun. 2005, https://doi.org/10.1109/TR.2005.845967.

[13] M. Vanhoucke and D. Debels, "The discrete time/cost trade-off problem: extensions and heuristic procedures," *Journal of Scheduling*, vol. 10, no. 4, pp. 311–326, Oct. 2007, https://doi.org/10.1007/s10951-007-0031-y.

[14] C. Su and Y. Liu, "Multi-objective imperfect preventive maintenance optimisation with NSGA-II," *International Journal of Production Research*, vol. 58, no. 13, pp. 4033–4049, Jul. 2020, https://doi.org/10.1080/00207543.2019.1641237.

[15] J. T. Saraiva, M. L. Pereira, V. T. Mendes, and J. C. Sousa, "A Simulated Annealing based approach to solve the generator maintenance scheduling problem," *Electric Power Systems Research*, vol. 81, no. 7, pp. 1283–1291, Jul. 2011, https://doi.org/10.1016/j.epsr.2011.01.013.

[16] M. Samrout, F. Yalaoui, E. Châtelet, and N. Chebbo, "New methods to minimize the preventive maintenance cost of series–parallel systems using ant colony optimization," *Reliability Engineering & System Safety*, vol. 89, no. 3, pp. 346–354, Sep. 2005, https://doi.org/10.1016/j.ress.2004.09.005.

[17] S. Rajendran, A. Ansaripour, M. K. Srinivasan, and M. J. Chandra, "Stochastic goal programming approach to determine the side effects to be labeled on pharmaceutical drugs," *IISE Transactions on Healthcare Systems Engineering*, vol. 9, no. 1, pp. 83–94, Jan. 2019, https://doi.org/10.1080/24725579.2018.1488157.

[18] E. Taghizadeh, M. Abedzadeh, and M. Setak, "A Multi Objective Reliable Location-Inventory Capacitated Disruption Facility Problem with Penalty Cost Solve with Efficient Meta Historic Algorithms," *arXiv:1711.09400 [cs, stat]*, Nov. 2017, Accessed: Dec. 04, 2020. [Online]. Available: http://arxiv.org/abs/1711.09400.

[19] A. Ansaripour, A. Mata, S. Nourazari, and H. Kumin, "Some Explicit Results for the Distribution Problem of Stochastic Linear Programming," *Open Journal of Optimization*, vol. 5, no. 4, pp. 140–162, Dec. 2016, https://doi.org/10.4236/ojop.2016.54014.

[20] P. Abbasian, N. Mahdavi-Amiri, and H. Fazlollahtabar, "Multiple utility constrained multi-objective programs using Bayesian theory," *Journal of Industrial Engineering International*, vol. 14, no. 1, pp. 111–118, Mar. 2018, https://doi.org/10.1007/s40092-017-0211-0.

[21] H. Fazlollahtabar, P. Abbasian, and N. Mahdavi-Amiri, "Modelling and Optimization of a Non-Constrained Multi-objective Problem having Multiple Utility Functions using Bayesian Theory," *Journal of Information Sciences and Computing Technologies*, vol. 4, no. 3, pp. 332–342, Oct. 2015.

[22] P. Ghoddousi, E. Eshtehardian, S. Jooybanpour, and A. Javanmardi, "Multi-mode resource-constrained discrete time–cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm," *Automation in Construction*, vol. 30, pp. 216–227, Mar. 2013, https://doi.org/10.1016/j.autcon.2012.11.014.

[23] H. Jafarzadeh, N. Moradinasab, and M. Elyasi, "An Enhanced Genetic Algorithm for the Generalized Traveling Salesman Problem," *Engineering, Technology & Applied Science Research*, vol. 7, no. 6, pp. 2260–2265, Dec. 2017, https://doi.org/10.48084/etasr.1570.

[24] K. Soleimani and J. Mazloum, "Designing a GA-Based Robust Controller For Load Frequency Control (LFC)," *Engineering, Technology & Applied Science Research*, vol. 8, no. 2, pp. 2633–2639, Apr. 2018, https://doi.org/10.48084/etasr.1592.

[25] M. Tavana, K. Khalili-Damghani, D. Di Caprio, and Z. Oveisi, "An evolutionary computation approach to solving repairable multi-state multi-objective redundancy allocation problems," *Neural Computing and Applications*, vol. 30, no. 1, pp. 127–139, Jul. 2018, https://doi.org/10.1007/s00521-016-2676-y.

[26] A. Afshar, A. K. Ziaraty, A. Kaveh, and F. Sharifi, "Nondominated Archiving Multicolony Ant Algorithm in Time–Cost Trade-Off Optimization," *Journal of Construction Engineering and Management*, vol. 135, no. 7, pp. 668–674, Jul. 2009, https://doi.org/10.1061/(ASCE)0733-9364(2009)135:7(668).

[27] M. Rahimi and H. Iranmanesh, "Multi Objective Particle Swarm Optimization for a Discrete Time, Cost and Quality Trade-off Problem," *World Applied Sciences Journal*, vol. 4, no. 2, pp. 270–276, 2008.

[28] H. Zhang and H. Li, "Multi-objective particle swarm optimization for construction time-cost tradeoff problems," *Construction Management and Economics*, vol. 28, no. 1, pp. 75–88, Jan. 2010, https://doi.org/10.1080/01446190903406170.

[29] K. S. Moghaddam, "Multi-objective preventive maintenance and replacement scheduling in a manufacturing system using goal programming," *International Journal of Production Economics*, vol. 146, no. 2, pp. 704–716, Dec. 2013, https://doi.org/10.1016/j.ijpe.2013.08.027.

[30] M.-C. Fitouhi and M. Nourelfath, "Integrating noncyclical preventive maintenance scheduling and production planning for a single machine," *International Journal of Production Economics*, vol. 136, no. 2, pp. 344–351, Apr. 2012, https://doi.org/10.1016/j.ijpe.2011.12.021.

[31] "Products," *Sumitomo Electric Hardmetal*. https://www.sumitool.com/en/products/ (accessed Dec. 04, 2020).