International Journal of Production Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Multi-objective optimisation of tool indexing problem: a mathematical model and a modified genetic algorithm

Kaveh Amouzgar, Amir Nourmohammadi & Amos H.C. Ng

To cite this article: Kaveh Amouzgar, Amir Nourmohammadi & Amos H.C. Ng (2021): Multi-objective optimisation of tool indexing problem: a mathematical model and a modified genetic algorithm, International Journal of Production Research, DOI: 10.1080/00207543.2021.1897174

To link to this article: https://doi.org/10.1080/00207543.2021.1897174

Published online: 13 Mar 2021.

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
https://doi.org/10.1080/00207543.2021.1897174

Taylor & Francis
Taylor & Francis Group

Multi-objective optimisation of tool indexing problem: a mathematical model and a modified genetic algorithm

Kaveh Amouzgar [a,b], Amir Nourmohammadi [b] and Amos H.C. Ng [a,b]

[a]Division of Industrial Engineering and Management, Uppsala University, Uppsala, Sweden; [b] School of Engineering Science, University of Skövde, Skövde, Sweden

ABSTRACT

Machining process efficiencies can be improved by minimising the non-machining time, thereby resulting in short operation cycles. In automatic-machining centres, this is realised via optimum cutting tool allocation on turret-magazine indices – the "tool-indexing problem". Extant literature simplifies TIP as a single-objective optimisation problem by considering minimisation of only the tool-indexing time. In contrast, this study aims to address the multi-objective optimisation tool-indexing problem (MOOTIP) by identifying changes that must be made to current industrial settings as an additional objective. Furthermore, tool duplicates and lifespan have been considered. In addition, a novel mathematical model is proposed for solving MOOTIP. Given the complexity of the problem, the authors suggest the use of a modified strength Pareto evolutionary algorithm combined with a customised environment-selection mechanism. The proposed approach attained a uniform distribution of solutions to realise the above objectives. Additionally, a customised solution representation was developed along with corresponding genetic operators to ensure the feasibility of solutions obtained. Results obtained in this study demonstrate the realization of not only a significant (70%) reduction in non-machining time but also a set of tradeoff solutions for decision makers to manage their tools more efficiently compared to current practices.

ARTICLE HISTORY
Received 22 April 2020
Accepted 14 February 2021

KEYWORDS
Tool indexing; genetic algorithm; non-machining time; multi-objective optimisation; SPEA2; mathematical model

1. Introduction

In flexible manufacturing systems (FMS), the production time can be significantly reduced by performing a sequence of machining operations on a single *Computerised Numerical Control (CNC)* lathe machine. Depending on the process plan, these operations require the use of different tool types in a predefined order. The turret magazine or automatic tool changer (ATC) can hold multiple tools in different tool indexes (i.e. pockets, slots, or stations) to facilitate automatic execution of the sequence of operations. The turret magazine delivers the required tools in the cutting position of a CNC machine by rotating about its vertical axis.

The rotation of the turret magazine from one tool index to the next is called *unit rotation*, whereas the time required by the turret to switch from one tool to another is referred to as the *turret-* or *tool-indexing time*. Because no cutting operation can be performed during turret rotations to switch between different tools, the turret-indexing time is also referred to as the *non-machining time*. The total processing time equals the sum of the cutting and tool-indexing times. Although the cutting time can be shortened by optimising the machining process, we assumed this to remain constant because such optimizations are beyond the scope of this study. However, the tool-indexing time can be significantly reduced by determining the best strategy for allocating appropriate index positions to cutting tools on the turret magazine of a CNC machine. This problem of indexing the tools in appropriate slots to minimise the total number of unit rotations (or the tool-indexing time) of the turret is referred to as the *tool indexing problem (TIP)* or *ATC indexing problem* (Dereli and Filiz 2000).

Turret magazine can rotate in any one or both directions about its vertical axis. In bidirectional rotating turrets, the sum of unit rotations from the current to the target index is always less than or equal to half the turret capacity. Consequently, bi-directional turrets are more preferred owing to the ability to determine the smallest angle of rotation required to switch between two tools.

If the number of required tools exceeds that of available index positions, some tools on the magazine may need to be removed and replaced by another tool, thereby resulting in an additional optimisation problem aimed at

CONTACT Kaveh Amouzgar kaveh.amouzgar@angstrom.uu.se Division of Industrial Engineering and Management, Uppsala University, PO Box 534, Uppsala, 75121 Sweden; School of Engineering Science, University of Skövde, Skövde 541 28, Sweden

minimising the number of tool switches required over time. This is referred to as the tool-switching problem (ToSP).

In real-world industrial applications, the cutting tools placed on ATC indexes are used to perform solitary operation or their sequences on several workpieces or parts. Accordingly, life of a cutting tool is determined in terms of the number of parts it can machine prior to being no longer usable. That is, if a tool can machine 100 parts, it has a lifespan of 100 parts, whereas another tool with twice that lifespan would require replacement after machining 200 parts. Thus, it is crucial to resolve TIP whilst considering the lifespan of all tools placed in a turret magazine. This implies that the index positions of tools placed in a magazine must be so determined that the total tool-indexing time for all parts could be minimised. In other words, a magazine must be removed from a CNC machine only when all tools placed in it need replacement.

Most prior TIP studies have assumed that either the tools do not undergo wear or their operating life is sufficiently long to operate on a *single part*, thereby simplifying the problem to that involving single-step optimisation.

In contrast to prior research, this study aims at solving TIP for a real-world industrial case pertaining to the automotive industry by accounting for the entire lifespan of tools used in the machining of *1304 crankshafts*. Thus, we intend to resolve TIP encountered in a more realistic scenario involving an actual machining centre.

Most studies have identified TIP and ToSP as NP-hard problems (Dereli and Filiz 2000; Baykasoğlu and Dereli 2004; Baykasoğlu and Ozsoydan 2016). Furthermore, in Atta, Sinha Mahapatra, and Mukhopadhyay (2019), TIP has been formally defined and modelled as a *quadratic assignment problem* (QAP) (Loiola et al. 2007), an NP-hard problem. This implies that finding an optimal solution using exact methods is computationally expensive. Consequently, resorting to metaheuristic methods is the main option, especially in our case with a large decision space.

In contrast to the literature in which the TIP is simplified into a single-objective optimisation problem by only minimising the total tool-indexing time in this study, inspired by observation in a real case, one more criterion, namely the number of changes required from the current industrial setting is found to be of importance for the industrial partner, which transforms the TIP to a multi-objective optimisation problem (MOOP), or henceforth referred to as the *multi-objective optimisation of tool indexing problem (MOOTIP)*.

In the preference-based or priori technique for solving MOOPs, the preference vector transforms MOOP into a single-objective problem by considering high-level information in the beginning of the optimisation process (Deb 2001). Consequently, the ideal or posteriori technique, wherein a solution set is obtained in the form of Pareto-optimal solutions, is generally preferred over the priori technique for solving MOOPs. This affords decision makers the freedom to analyse the obtained results with a better understanding of the involved variables, objectives, and the relationship between them prior to selecting the desired solution based on high-level information alone.

The characteristics of evolutionary algorithms (EA), by virtue of which a solution population can evolve in generations, makes them suitable for use in solving MOOPs within the ideal-approach framework. However, solutions obtained using EA might not be truly Pareto-optimal. Moreover, it is impossible to validate the Pareto optimality of solutions when the problem at hand lacks analytical expressions involving objectives and constraints. Therefore, *tradeoff* or *non-dominated solutions* are considered conventional when referring to solutions obtained using EAs. Although *MOOTIP* has not yet been investigated in detail, previous studies have reported a range of MOOPs solved using EAs in several FMS applications (Reddy and Rao 2006; Tseng et al. 2008; Zhang et al. 2010; Soolaki 2013; Shen and Yao 2015).

Having said the above-mentioned gaps, this paper proposes a novel mathematical model for MOOTIP. Moreover, an implementation of the well-known strength Pareto evolutionary algorithm (SPEA2) (Zitzler, Laumanns, and Thiele 2001) to solve a complex industry-inspired MOOTIP is presented. The said SPEA2 was modified using a customised *environment-selection* mechanism to obtain a uniform distribution of solutions with regard to the second objective. In addition, to address the specific nature of the problem at hand, a customised solution representation and its corresponding genetic operators were developed to ensure the feasibility of the solutions obtained.

The major contributions of this study can be summarised as follows.

- This study represents an attempt to solve a complex real-world problem inspired by the automotive industry
- Tool wear has been considered for the first time in TIP
- A new objective function is proposed to transform TIP into an MOOP with a view to improve existing industrial practices
- A novel mixed-integer non-linear programming model for MOOTIP is proposed for the first time
- A well-known MOO algorithm has been modified by customising the solution representation,

environmental selection, and genetic operators to solve MOOTIP

The remainder of this paper is organised as follows. Section 2 provides a review of the related literature. Section 3 describes an actual case study performed in an automobile-engine production plant. A mathematical model for MOOTIP is presented in Section 4. Details concerning the modified SPEA2 and related mechanism are described in Section 5. Section 6 discusses the results obtained in this study. Lastly, major conclusions drawn from this study are presented in 7.

2. Literature review

The three possible ATC scenarios can be summarised as follows (Baykasoğlu and Dereli 2004).

(1) *ToSP*: $NT > NS$,
(2) *TIP*: $NT < NS$,
  (a) (a)without tool duplicates
  (b) (b)with tool duplicates
(3) *TIP*: $NT = NS$,

Here, *NS* denotes the number of index positions available on the turret magazine, and *NT* denotes the number of tools required to execute a given sequence of operations.

The first study concerning ToSP was performed by Hertz et al. (1998), and it has since attracted increased research attention. Several ToSP solution methodologies proposed to date include the genetic algorithm (GA) (Keung, Ip, and Lee 2001), branch & cut and branch & bound (Laporte, Salazar-González, and Semet 2004), and multiple start algorithm (Salonen, Raduly-Baka, and Nevalainen 2006). Additionally, a multi-objectives algorithmic framework for ToSP has been proposed by Furrer and Mütze (2017). Tool positions with arbitrary production sequences have been pre-optimised in another study (Ménard, Quimper, and Gaudreault 2019).

The consideration of more than one objectives in ToSP optimisation has featured in several prior studies. Keung, Ip, and Lee (2001) optimised the number of tool switches in ToSP using a GA, albeit in combination with the preference-based approach. In addition, the unit-switching problem in stock keeping, which is closely related to ToSP in FMS, has been optimised via the use of a heuristic decomposition approach (Schwerdfeger and Boysen 2017).

ToSP related to FMS scheduling have also been reported in several studies. A biased random-key GA to solve a scheduling problem with tool constraints has been reported in Soares and Carvalho (2020), whereas a mathematical modelling and multi-attribute rule mining approach for job-shop scheduling has been described in Zhang et al. (2019). The realization of consecutive block minimisation via use of heuristic methods has been reported in Soares et al. (2020).

In a study by Wang et al. (2020), the total non-productive time of the machining process, including tool travelling time, tool switching time and Z-axis compensation moving time is minimised by presenting a hybrid variable neighbourhood search/tabu search and neighbourhood generation strategy. Wang, Zou, and Wang (2020) found the optimal solutions to a parallel machine scheduling problem combining operation scheduling, tool scheduling and restrained resources are obtained by a Tabu-Genetic algorithm. Dang et al. (2021) proposed a mathematical model and combination of a genetic algorithm and an integer linear programming formulation to solve industry-size instances of parallel machine scheduling with tool replacements.

The provision of job sequencing as an input to FMS has been reported in several studies concerning ToSP (Crama et al. 2007; Konak, Kulturel-Konak, and Azizoğlu 2008; Raduly-Baka and Nevalainen 2015). Additionally, the use of a heuristic approach to solve the job sequencing and tool-switching problem (SSP) has been reported in Paiva and Carvalho (2017). In another study (Ahmadi et al. 2018), SSP has been modelled as a second-order travelling-salesman problem, and the same has been solved using a learning-based GA.

A multi-commodity flow mathematical model for SSP is presented (da Silva, Chaves, and Yanasse 2020). In the most recent study (Mecler, Subramanian, and Vidal 2021), SSP is solved by a hybrid genetic search based on a generic solution representation.

However, despite TIP's significant influence on the FMS machining efficiency, very few studies have been performed to minimise the non-machining time.

The first attempt to address TIP was made by Dereli et al. (1998) and Dereli and Filiz (2000). They developed a GA-based optimisation software to facilitate optimal allocation of cutting tools on ATC/turret-magazine index positions. Baykasoğlu and Dereli (2004) proposed a meta-heuristic optimisation system to minimise the total tool-indexing time. The possibility of allocating more than one instance of some or all tools indexed on the turret magazine was considered. The number of duplicates for each tool type and optimum index location for each tool on the ATC was obtained using the simulated-annealing (SA) algorithm, thereby resulting in the realization of minimum tool-indexing time. The objective function was calculated by constructing a tree of all possible options and evaluating all alternatives to determine the best possible index-routing strategy.

A similar study proposed the use of an ant colony (AC) algorithm to determine the optimum index position for cutting tools (Gopala and Rao 2006). Baykasoğlu and Ozsoydan (2016) improved their prior study (Baykasoğlu and Dereli 2004) by evaluating the objective function by integrating the shortest-path algorithm as a sub-optimisation problem of the larger ATC indexing problem to address TIP (Baykasoğlu and Ozsoydan 2016). In another study (Baykasoğlu and Ozsoydan 2017), the SA algorithm was used to solve the TIP and ToSP problems simultaneously. In addition, a solution strategy for solving TIP and ToSP simultaneously in a dynamic environment has been proposed by Baykasoğlu and Ozsoydan (2018). Changes in lot sizes and those in operating personnel represent two types of dynamic operating conditions considered in that study.

A study (Atta, Sinha Mahapatra, and Mukhopadhyay 2019) aimed to solve TIP via the use of a customised harmony search (HS) algorithm in combination with a harmony-refinement strategy. Tool duplicates were ignored, and it was considered that if the number of slots on the turret magazine exceeds the number of required tools, some slots would remain unoccupied.

In a recent work (Amouzgar, Ng, and Ljustina 2020) a GA is proposed to solve a real-world case considering tool duplicates and tool wear. Palubeckis (2020) presented an approach integrating SA and variable neighbourhood search algorithm for bidirectional loop layout problem. The algorithm was tested on two sets of tool indexing problem instances.

Noteworthy research in this direction has been summarised in Table 1.

As it can be observed from the Table, this study is the first attempt to fill the existing gap in the literature by considering tool wear and multi-objective optimisation while addressing TIP. Furthermore, a novel mathematical model for MOOTIP is proposed and a real-world complex industrial case is solved by utilising an efficient meta-heuristic algorithm.

3. Problem description

TIP is the allocation of cutting tools on the indexes of turret magazine to reduce the tool indexing time or total number of unit rotations. Figure 1 (adopted from Dereli and Filiz 2000) illustrates a turret magazine with eight indexing stations and four different cutting tools (T1–T4) allocated to the slots.

As already mentioned, the proposed study is inspired by the sequence of machining operations, referred to as Operation 30 (OP30), performed in the engine production unit of a major automotive company. In OP30, a biaxial CNC lathe is used to machine the crankshafts of 4-cylinder engines. OP30 comprises 19 turning operations that involve use of 15 different tools. The CNC lathe machine considered in this study was equipped with two circular bidirectional turret magazines each capable of holding 45 tools (inserts). Both magazines measured more than 500 mm in diameter with a turret-indexing time of approximately 0.2 s. The turrets are mounted on

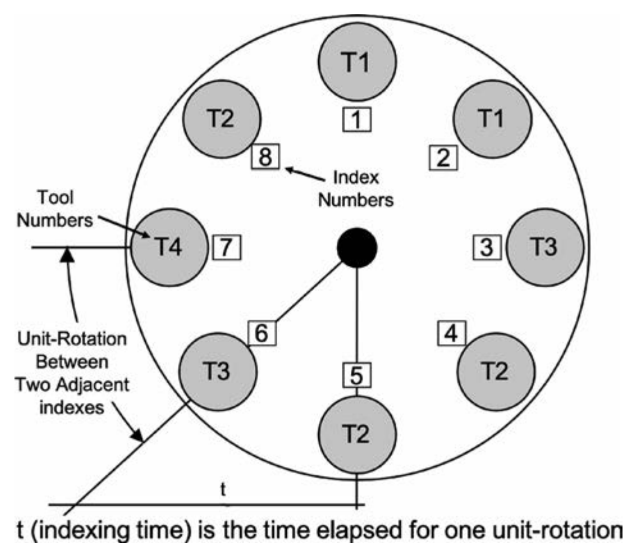
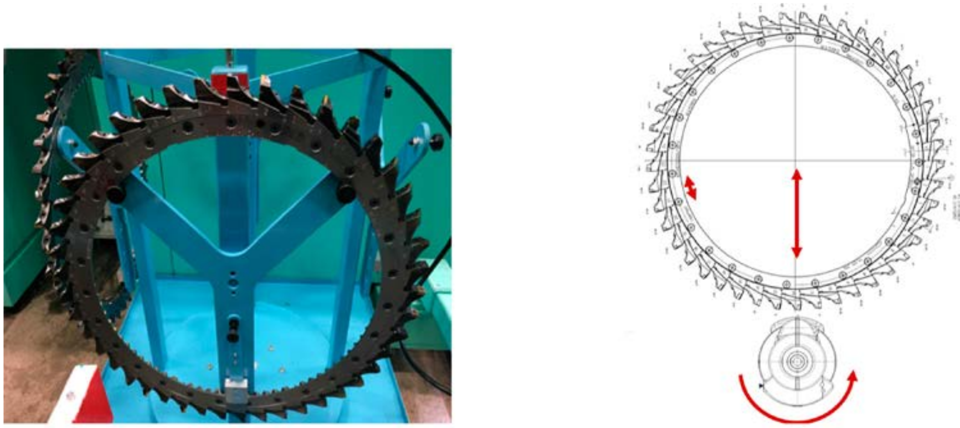


Figure 1. Tool indexing on an 8-index turret magazine (Dereli and Filiz 2000).

Table 1. Summary of research performed to address tool-indexing problems.

study	Tool Duplicate	Empty Index	Tool Wear	Mathematical Model	Solution Approach	TIP	ToSP	Number of objectives SOO	MOO
Dereli et al. (1998)					GA	o		o	
Dereli and Filiz (2000)					GA	o		o	
Baykasoğlu and Dereli (2004)	o				SA	o		o	
Gopala and Rao (2006)		o			AC	o		o	
Baykasoğlu and Ozsoydan (2016)	o				SA,VNS	o		o	
Baykasoğlu and Ozsoydan (2017)	o	o			SA	o	o	o	
Baykasoğlu and Ozsoydan (2018)	o	o			SA	o	o	o	
Atta, Sinha Mahapatra, and Mukhopadhyay (2019)		o		o	HS	o		o	
Amouzgar, Ng, and Ljustina (2020)	o	o	o		GA	o		o	
Palubeckis (2020)		o			SA,VNS	o		o	
This study	o	o	o	o	GA	o			o

Note: Genetic algorithm (GA), simulated annealing (SA), ant colony (AC), harmony search (HS), tool-indexing problem (TIP), tool-switching problem (ToSP), single-objective optimisation (SOO), multi-objective optimisation (MOO), variable neighbourhood search (VNS).



(a) (b)

Figure 2. (a) Turret magazine with 45 index positions used in OP30; (b) schematic of OP30 turret.

Table 2. OP30 operations with required tools and tool life (i.e. the number of crankshafts each tool can machine before complete wear out).

Operation number	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Tool number	T1	T2	T6	T5	T7	T5	T6	T7	T4	T3
Tool life	145	131	326	326	326	326	326	326	131	326

Table 3. Number of duplicates for each tool and magazine-index positions allocated to each tool based on the current factory settings.

Tool No.	No. of Duplicates	Index positions
T1	9	1,3,5,8,12,16,20,24,28
T2	10	2,4,6,10,14,18,22,26,30,32
T3	4	7,19,21,33
T4	10	9,11,13,15,17,23,25,27,29,31
T5	4	34,36,40,42
T6	4	35,37,41,43
T7	4	38,39,44,45

two axes of a CNC machine that are operated in parallel. The right and left turrets can perform 10 and 9 different operations, respectively, by using 7 and 8 different cutting tools. It is noteworthy that although this study focuses on the right turret magazine, the proposed methodology and algorithm can be readily applied to other magazines as well. Figure 2 depicts a schematic of the turret magazine used in OP30. As can be seen, the figure illustrates the relative positions of the crankshaft and turret.

Table 2 lists the sequence of OP30 operations along with the type of tools requires to execute them. The said operations and tool types are denoted by symbols $O1$–$O10$ and $T1$–$T7$, respectively. The number of crankshafts a tool can machine before wearing out (fixed tool life) is also stated in the table.

To reduce the frequency of performing the complex and time-consuming task of removing turret magazines from the CNC lathe and replacing worn-out tools with new ones, it is considered apt to maximise the time for which each magazine remains in operation prior to tool replacement. To this end, the number of duplicates required for each tool type can be determined by solving a simple optimisation problem, the objective of which is to maximise the number of crankshafts that can be machined without the need to remove the turret magazine. In this study, this maximum number of crankshafts equals 1304.

Table 3 describes the current setting of the OP30 sequence, including the different tool types, number of duplicates, and station (or index) number corresponding to the placement of each tool on the magazine. Figure 3 illustrates the tool-allocation strategy currently employed in the factory. In accordance with the objective of this study, the product of the number duplicates for each tool (Table 3) and the corresponding tool life (Table 2) is equal or greater than 1304 for all tools. For instance, if one considers tool T1, it is obvious that $9 \times 145 = 1305 > 1304$.

The current tool-allocation strategy is trial-and-error-based and depends on the experience of the production and tooling engineers. To the best of our knowledge, no prior research has attempted to minimise the tool-indexing time whilst optimising existing tool-allocation strategies. Furthermore, altering the index positions of the cutting tools from the current setting will require additional feasibility studies. For instance, a collision detection analysis must be performed prior to adoption of an optimum tool-allocation strategy in the production line to detect any possible intersection of the tool engaged in the cutting operation with adjacent tools or that between individual tools and the crankshaft. Likewise, a load-balancing analysis of the turret can be undertaken after altering the tool-index positions.

Due to the above-mentioned complexities and based on the decision makers' suggestion, a second objective

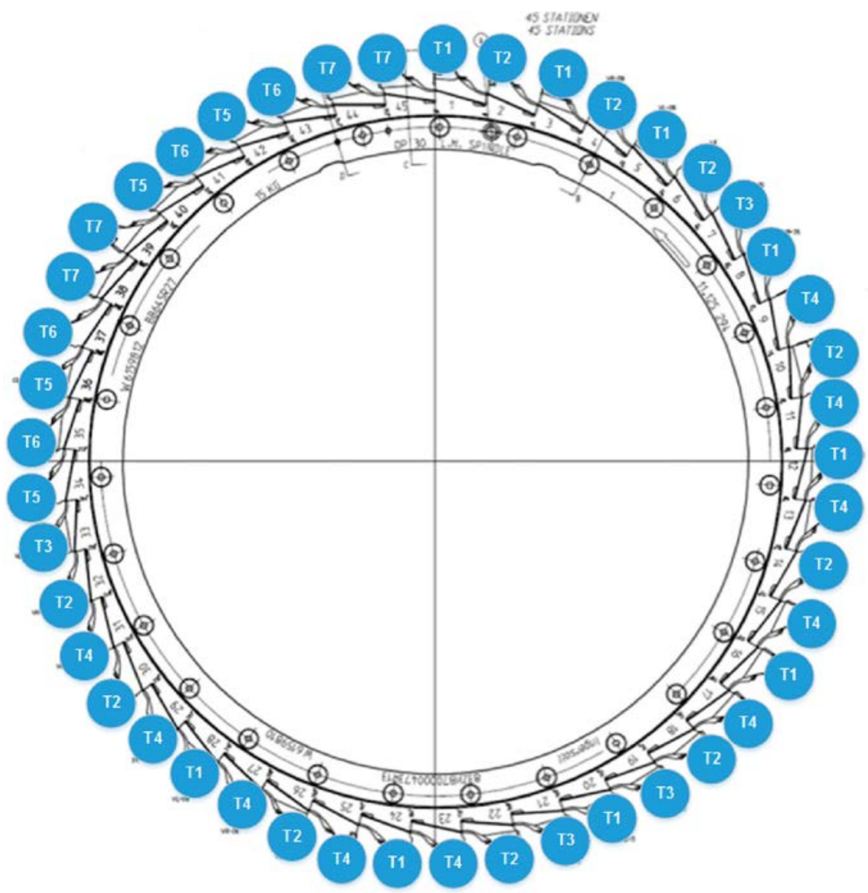


Figure 3. Factory-settings-based tool allocation on 45-index turret.

was added to the problem to be minimised, i.e. the number of changes in index positions from the current setting in the factory. Therefore, inspired by this industrial requirement, the proposed investigates the possibility of converting a classical TIP into an MOOP with the following objectives.

- *Objective1:* Minimizing the total tool-indexing time (hereinafter referred to as the *number of rotations (NoR)*) required for machining 1304 crankshafts
- *Objective2:* Minimizing the number of changes in the cutting-tool index (hereinafter referred to as the *number of changes (NoC)*)

It is believed that the set of non-dominated solutions obtained in this study would enhance the decision-making process in industrial settings. In fact, the proposed reduction in the number of changes has been considered in prior studies (Pehrsson, Ng, and Bernedixen 2016; Ng, Siegmund, and Deb 2018) aimed at resolving problems concerning efficiency improvement production lines. However, consideration of the same to improve the machining process has not yet been reported.

The following assumptions are made while solving this problem:

- All tools have fixed lifespans based on the number of crankshafts they can machine before replacement
- The number of tool duplicates is pre-determined based on the tool lifespan and operation sequence
- No tool breakage occurs prior to the expiry of the turret lifespan – i.e.before machining 1304 crankshafts
- There exist no collisions between tools placed in the two magazines nor between individual tools and the crankshafts being machined

4. Mathematical formulation of MOOTIP

In this study, a mixed-integer non-linear programming model *(MINLP)* is proposed to solve MOOTIP. The MINLP formulation was used to minimise two objectives: (1) total number of rotations required to perform a sequence of operations on crankshafts and (2) total number of changes required to modify the current tool allocation on the magazine in the company. The MINLP model used in this study was developed based on the following assumptions.

- The number of slots ($s = 1, \ldots, NS$) on a turret magazine needs to be filled using a given number of tools ($t = 1, \ldots, NT$)
- The number of tools required to perform a sequence of operations is less than or equal to the number of slots on the magazine, $NT \leq NS$
- It is possible to have duplicates for each tool type on the magazine
- Unlike the literature that have assumed the tools do not undergo wear or the tool life is long enough for carrying out cutting operation for a single part, this study takes into account the tool wear by solving TIP for the entire tool life

Tables 4–6 define the notations of sets, parameters, and decision variables used in this study to formulate TIP.

$$\text{Minimize} \quad NoR = \sum_{p=1}^{NP} \sum_{o=1}^{NO-1} \sum_{s=1}^{NS} \sum_{s'=1}^{NS} z_{pos} \times z_{p,o+1,s'} \times d_{ss'} \tag{1}$$

$$\text{Minimize} \quad NoC = \sum_{s=1}^{NS} \sum_{t=1}^{NT} \frac{1}{2} \left| y_{st} - y^{c}_{st} \right| \tag{2}$$

$$\text{Subjected to} \quad \sum_{s=1}^{NS} z_{pos} = 1 \quad \forall p, \forall o, \tag{3}$$

Table 4. TIP sets and their definitions.

Sets	Definition
$t : 1, \ldots, NT$	Tool index
$s, s' : 1, \ldots, NS$	Slot index
$o : 1, \ldots, NO$	Operation index
$p : 1, \ldots, NP$	Part index

Table 5. TIP parameters and their definitions.

Parameters	Definition
NT:	Number of tools
NS:	Number of slots
NO:	Number of operations
NP:	Number of parts
l_t:	Tool life of tool t
dup_t:	Duplication of tool t
fo_{ot}:	$\begin{cases} 1: & \text{if operation o uses tool t} \\ 0; & \text{otherwise} \end{cases}$
y^{c}_{st}:	$\begin{cases} 1: & \text{if tool t is placed on slot s in current magazing status} \\ 0; & \text{otherwise} \end{cases}$
$d(s, s')$:	minimum number of rotations between slot s and s'; $\min\{\lvert s' - s\rvert, \lvert s + NS - s'\rvert\}$

Table 6. TIP decision variables and their definitions.

Decision variables	Definition
y_{st}:	$\begin{cases} 1: & \text{if tool t is placed on slot s} \\ 0; & \text{otherwise} \end{cases}$
z_{pos}:	$\begin{cases} 1: & \text{if operation o uses slot s for machining part p} \\ 0; & \text{otherwise} \end{cases}$

$$\sum_{o=1}^{NO} \sum_{s=1}^{NS} z_{pos} = NO \quad \forall p, \tag{4}$$

$$y_{st} \geqslant fo_{ot} \times z_{pos} \quad \forall p, \forall s, \forall o, \forall t \in \{fo_{ot} = 1\}, \tag{5}$$

$$\sum_{t=1}^{NT} y_{st} = 1 \quad \forall s, \tag{6}$$

$$\sum_{s=1}^{NS} y_{st} = dup_t \quad \forall t, \tag{7}$$

$$\sum_{p=1}^{NP} z_{pos} \leqslant l_t \quad \forall s, \forall o, \forall t \in \{fo_{ot} = 1\}. \tag{8}$$

In the proposed model, Equation (1) aims to minimise the total number of rotations (NoR) required by the magazine to perform the operations, whereas Equation (2) minimises the total number of tool replacements to be made to the current setting (NoC). In addition, Equation 3 ensures that for machining of each part at each operation only one slot can be used, while Equation (4) determines that the total number of visited slots used for machining each part is equal to the number of operations. Equation (5) implies that in order to machine part p in operation o by tool t and slot s, the corresponding tool for that operation (i.e. $fo_{ot} = 1$), must be positioned on that slot as well. Equations (6) ensures that each slot is occupied by a single tool, whereas Equation (7) guarantees that the total number of each tool placed in the slots equals the number of duplicates for that tool (dup_t). Equation (8) deals with tool life in which the total number of parts that can be machined by each tool cannot exceed the pre-specified tool life.

To provide a better insight of the problem, a small scale MOOTIP in a turning operation for 12 parts consisting of 5 operations and 4 tool types using a 8-index turret magazine is solved using the proposed mathematical model. Figure 1 shows the current tool allocation of cutting tools on the magazine which is considered as the basis for the calculation of the second objective.

Table 7 lists the sequence of operations along with the type of tools required to execute them. The number of parts a tool can machine before wearing out (fixed tool life) is also stated in the table.

Table 7. Operations with required tools and tool life (i.e. the number of parts each tool can machine before complete wear out).

Operation Number	O1	O2	O3	O4	O5
Tool number	T2	T3	T4	T1	T2
Tool life	4	6	12	6	4

Table 8. Number of duplicates for each tool and magazine-index positions allocated to each tool based on the settings illustrated in Figure 1.

Tool no.	No. of duplicates	Index positions
T1	2	1,2
T2	3	4,5,8
T3	2	3,6
T4	1	7

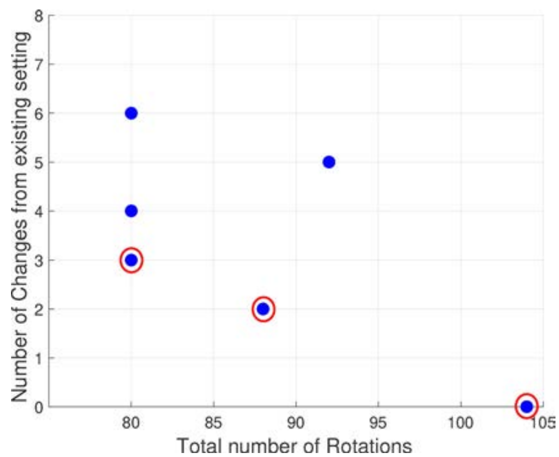


Figure 4. Solutions in the objective space for the 8-index turret magazine solved by the mathematical model.

Table 8 describes the current setting of operations' sequence, including the different tool types, number of duplicates, and station (or index) number corresponding to the placement of each tool on the magazine.

The two objective optimisation problem is converted to a single-objective optimisation by employing the ϵ-constrained method. The solutions in objective space are illustrated in Figure 4. It should be noted that the 3 solutions marked by the red circles are the non-dominated solutions of the MOOTIP in this example, as they dominate the other 3 solutions.

Due to the complexity of the TIP which has been argued to be NP-hard in the literature solving the MOOTIP using the proposed mathematical model in computationally expensive. Therefore, a MOO genetic algorithm for addressing the large-scale real-world MOOTIP is developed and presented in the next section.

5. Proposed modified SPEA2 (m-SPEA2)

In this study, we modified the implementation of the well-known (SPEA2), which has been previously used to optimise real-world engineering problems (Rezaei and Davoodi 2012; Amouzgar, Rashid, and Stromberg 2013; Tang et al. 2016; Amouzgar et al. 2018; Rao et al. 2019; Amouzgar et al. 2019). Figure 5 describes the SPEA2 process workflow.

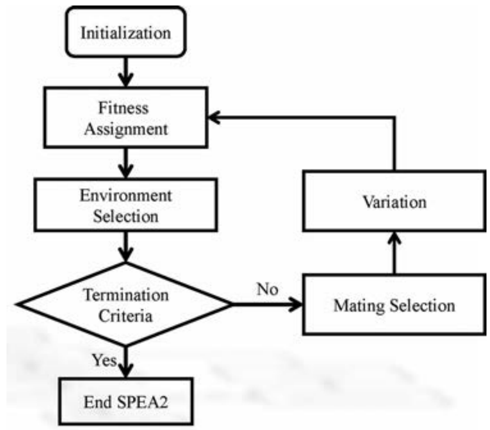


Figure 5. Flowchart of the SPEA2 algorithm.

The algorithm can be described as follows Zitzler, Laumanns, and Thiele (2001).

Input:	N	(populationsize)
	$\overline{N}$	(archivesize)
	T	(maximumnumberofgenerations)
Output:	A	(non − dominatedset)

Step 1: *Initialization*: Generate an initial population P_0 and archive (external set) $\overline{P}_0$. Set $t = 0$.

Step 2: *Fitness assignment*: Calculate fitness values of individuals in P_t and $\overline{P}_t$.

Step 3: *Environment selection*: Copy all non-dominated individuals in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. If the size of $\overline{P}_{t+1}$ exceeds $\overline{N}$, a reduction in $\overline{P}_{t+1}$ can be realised in terms of the truncation operator. On the contrary, if the size of $\overline{P}_{t+1}$ is less than $\overline{N}$, $\overline{P}_{t+1}$ is filled with dominated individuals in P_t and $\overline{P}_t$.

Step 4: *Termination*: If $t > T$ or another stopping criterion is satisfied, the non-dominated individuals in $\overline{P}_{t+1}$ create the output set A.

Step 5: *Mating Selection*: The binary tournament selection with replacement is performed on $\overline{P}_{t+1}$ to populate the mating pool.

Step 6: *Variation*: Apply the recombination (crossover) and mutation operators to the mating pool and set the resulting population to P_{t+1}; increment the generation counter ($t = t + 1$) and return to Step 2.

The difference between the algorithm used in this and the original SPEA2 lies in the environmental-selection step (step 3) and the use of customised genetic operators. A detailed explanation of customised elements pertaining to the proposed algorithm is provided in the following discussions.

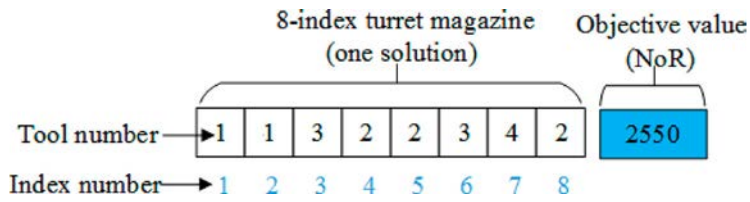


Figure 6. Solution representation of 8-index turret magazine depicted in Figure 1.

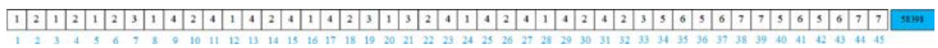


Figure 7. Solution representation of current tool-allocation strategy pertaining to the 45-index turret used in factories.

5.1. Solution representation

A vector-based solution representation method that can be readily employed in other similar cases has been developed to address the problem considered in this study.

In the proposed method, the set of obtained solutions can be represented as a row vector with corresponding columns representing index positions on the turret magazine. The top station is identified as the first index (no. 1) with subsequent indexes being numbered in the clockwise direction until index no. 45. Figure 6 depicts the solution representation corresponding to the 8-index turret magazine illustrated in Figure 1. As can be seen, the tool T1 is placed in index numbers 1 and 2 followed by tool T3 placed in index number 3, until tool T2 in the last index(number 8). The proposed method facilitates the representation of duplicate tool allocation via insertion of applicable tool numbers in several columns.

Owing to its current use in the production line, OP30 can be considered a feasible solution representation, as depicted in Figure 7. Therefore, other alternative solutions can be easily generated by modifying the current tool-allocation (shuffling) and using the generated solutions as the initial population provided as input to the evolutionary algorithm.

5.2. Objective functions

In this section, the two objective functions dealt with while addressing MOOTIP are discussed in detail.

5.2.1. First objective function – noR

The primary objective of TIP is to reduce the non-machining time, thereby increasing the productivity of the machining process. This results in lower production costs and increased profitability. The above benefits can be realised by minimising the tool-indexing time (objective-function value). The said indexing time can be calculated by multiplying the catalog indexing time of the CNC machine (0.2 s in this case) by the total number of unit rotations in accordance with the given operation sequence. Because the catalog indexing time remains constant, one can simply minimise the total number of unit rotations in a typical TIP.

Contrary to published literature, the authors in this study extended the objective function to minimise the total number of unit rotations the turret magazine must perform to machine 1304 crankshafts.

Consider the process to evaluate the objective function pertaining to the simple example illustrated in Figure 1, wherein 5 cutting operations are using 4 different tools in the sequence T4–T2–T3–T4–T1. The tools are placed on a turret magazine with 8 index positions. The number of duplicates and corresponding lifespans of tools T1 to T4 are 2, 3, 2, and 1 and 150, 100, 150, and 300 parts, respectively.

In accordance with the given operation sequence, we evaluated the total number of rotations required to machine the first workpiece. The optimisation objective is to determine the smallest angle through which the turret must rotate to position the tools T4, T2, T3, T4, and T1 in that sequence. Such an objective-function evaluation of each solution represents a separate optimisation subproblem of the master problem (i.e.optimum tool allocation on the turret), which can be converted to the shortest-path problem. To this end, we adopted the methodology proposed in Baykasoğlu and Ozsoydan (2016), which employs the Dijkstra shortest-path algorithm (Dijkstra 1959) with additional *dummy-in* (D_{in}) and *dummy-out* (D_{out}) nodes. Details concerning this methodology can be found in the cited references.

As observed, the objective value (the shortest path that can be transformed into the number of unit rotations) for machining the first workpiece using the allocation strategy depicted in Figure 1 equals *6 unit rotations*, whereas the shortest turret-index-rotation route is D_{in}-7-8-6-7-1-D_{out}.

The utility of all methods and algorithms proposed in previous studies ends here, and the obtained objective function value of 6 is reported as the specific solution. Previously proposed optimisation algorithms search for alternative allocation strategies with the aim of determining the minimum objective-function value. However, this approach is not realistic because the tools placed on the turret cannot be replaced with a new one after each part (crankshaft) is machined. Hence, the lifespan of tools included in the shortest path used for machining one crankshaft reduces by one. For instance, after machining 100 parts by considering the above-mentioned shortest route, tool T2 placed at index 8 cannot be used for machining any more parts owing to the expiry of its 100-part lifespan. In this situation, the Dijkstra algorithm must determine another shortest path without consideration of the tool placed at index 8. This process would continue until all tools on the turret magazine are worn-out. Thus, 300 parts can be machined using this approach. The corresponding objective-function value can be calculated as the sum of the product of the shortest distance and the number of parts that can be machined with that distance. Thus, the objective-function value for this simple 8-index turret can be calculated as follows. $NoR = (6 \times 100) + (6 \times 50) + (11 \times 50) + (11 \times 100) = \mathbf{2550}$. Here, the first and second numbers inside each pair of parentheses denote the smallest number of required turret rotations and the number of parts that could be machined in those rotations.

Similarly, the objective-function (unit rotation) value corresponding to the current tool-allocation strategy (Figure 7) based on the OP30 sequence and tool lifespans (Table 2) can be calculated as follows. $NoR = (23 \times 131) + (26 \times 131) + (27 \times 131) + (31 \times 14) + (32 \times 64) + (34 \times 67) + (35 \times 117) + (35 \times 14) + (35 \times 14) + (39 \times 36) + (40 \times 67) + (40 \times 61) + (45 \times 3) + (48 \times 61) + (48 \times 34) + (48 \times 14) + (49 \times 19) + (67 \times 3) + (69 \times 11) + (71 \times 53) + (76 \times 78) + (76 \times 14) + (76 \times 14) + (78 \times 14) + (79 \times 11) + (79 \times 64) + (92 \times 3) + (93 \times 31) + (93 \times 3) + (94 \times 14) + (95 \times 13) = \mathbf{58,398}$. This implies that the turret must perform 23 rotations to machine the first 131 crankshafts. Likewise, machining the last 13 crankshafts would require 95 unit rotations to be performed.

The above calculation implies that the current tool-allocation strategy used in the factory is far optimum. For instance, for the first batch of crankshafts when none of the tools are worn out 23 unit rotations is required which is *13 unit rotations* more than the optimum.

5.2.2. Second objective function – noC

The second objective function (i.e. the number of changes (NoC) required to be made to the tool-index positions) can be easily obtained by counting the number of discrepancies in the tool-index positions pertaining to a candidate solution and the current index positions in the factory. For example, a random feasible solution can be generated by modifying the current tool-allocation strategy, as illustrated in Figure 8. As can be seen, the figure highlights the tools positioned differently compared to the current factory setting. By adding the cells highlighted in the solution representation, the value of the second objective function can be calculated as $NoC = 6$ for the random solution.

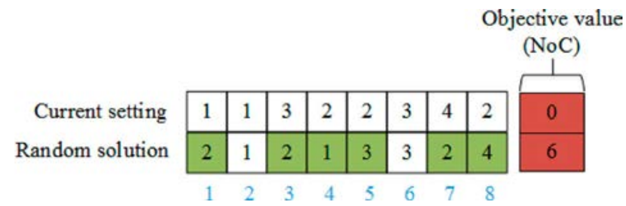


Figure 8. Calculating value of second objective function (NoC) for a random solution.

5.2.3. Environment selection

In this study, the archive-update process (step 3 in SPEA2) was modified to ensure the inclusion of at least one solution from each NoC in the archive set during each algorithm iteration.

The first step, similar to the environment-selection step in the original SPEA2, involves copying all non-dominated solutions to the next generation external set. In other words, solutions from the archive and population with fitness values less than unity are copied to $\overline{P}_{t+1}$ as follows.

$$\overline{P}_{t+1} = \{i | i \in P_t + \overline{P}_t \wedge F(i) < 1\}. \quad (9)$$

If the size of non-dominated solutions is identical to the archive size ($|\overline{P}_{t+1}| = \overline{N}$), the environment-selection step is completed. Otherwise, the two scenarios wherein the size of the archive set are either smaller or greater than the archive size ($\overline{N}$) are handled as follows.

Case 1: *The archive is too small* ($|\overline{P}_{t+1}| < \overline{N}$): In this case, if the archive set misses an NoC result in the non-dominated solutions, a dominated solution with the corresponding missing NoC corresponding to the lowest number of rotations (NoR) is first copied to the archive set from P_t and $\overline{P}_t$. Subsequently, the remaining $\overline{N} - |\overline{P}_{t+1}|$ are populated with the best-dominated solutions from the previous external set and population by sorting the multi-set $\overline{P}_t + \overline{P}_{t+1}$ in the increasing order of fitness values. Next, the first $\overline{N} - |\overline{P}_{t+1}|$ solutions with $F(i) > 1$ from the sorted list are copied to $\overline{P}_{t+1}$. This 3-step process is illustrated in Figure 9.

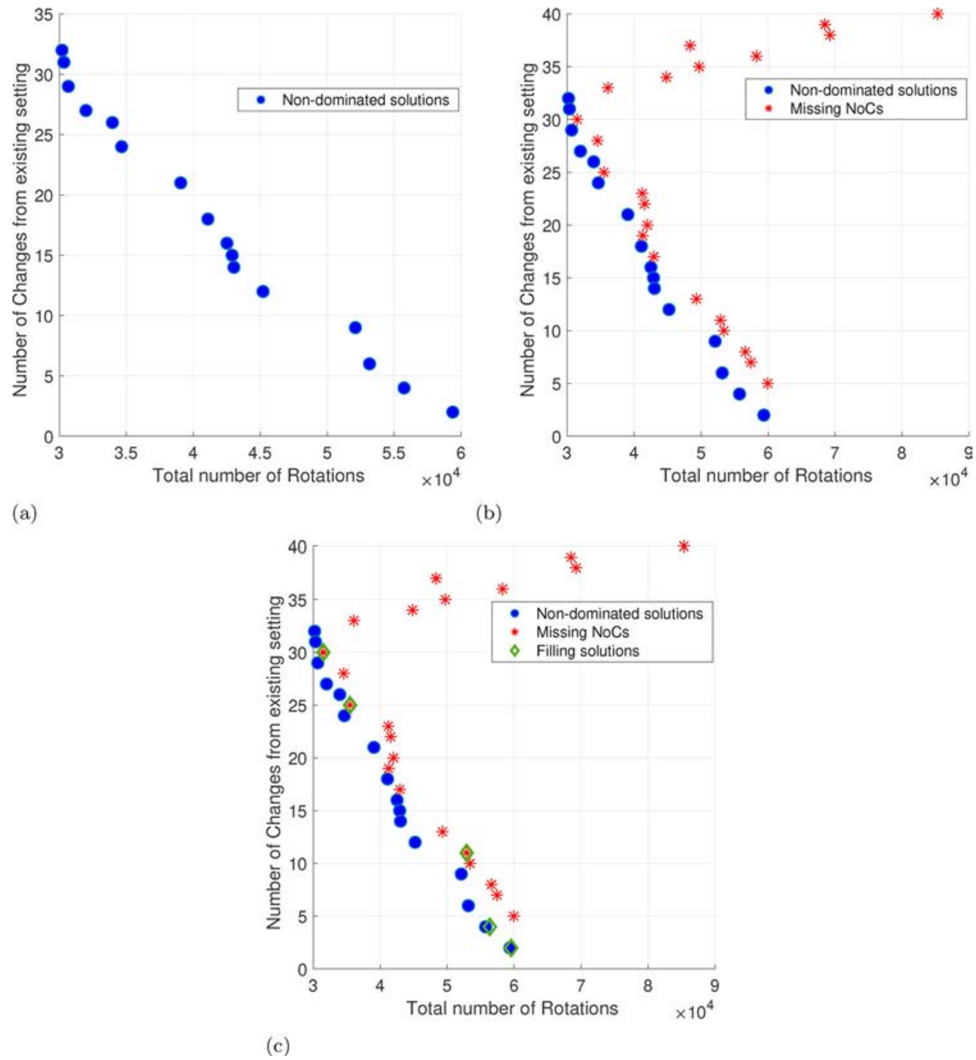


Figure 9. Illustration of environmental selection operator when the archive is too small (first scenario) for 45-index tool magazine with $\overline{N} = 50$, and $|\overline{P}_{t+1}| < 50$: (a) $|\overline{P}_{t+1}| = 16$; i.e. the 16 non-dominated solutions shown in this figure are first copied to $\overline{P}_{t+1}$; (b) missing NoC solutions and 22 solutions indicated by stars are added to $\overline{P}_{t+1}$, thereby increasing archive size to $|\overline{P}_{t+1}| = 16 + 22 = 38$; (c) remaining $50 - 38 = 12$ solutions are chosen from the best-dominated solutions in the multi-set $P_t + \overline{P}_{t+1}$ and added to the archive (indicated by diamonds); hence $|\overline{P}_{t+1}| = 50$.

Case 2: *The archive is too large* ($|\overline{P}_{t+1}| > \overline{N}$): Here, the best non-dominated NoR solution from each NoC is maintained; i.e. $\overline{P}_{t+1}$ includes at least one non-dominated solution from each NoC. Subsequently, the archive-truncation procedure, from the original SPEA2, is invoked, thereby resulting in $|\overline{P}_{t+1}| = \overline{N}$ via elimination of solutions from $\overline{P}_{t+1}$ through an iterative process. During each iteration, the solution i for which $i \leq_d j$ for all $j \in \overline{P}_{t+1}$ is eliminated in accordance with the below expression.

$$i \leq_d j \quad :\Leftrightarrow \forall\, 0 < k < |\overline{P}_{t+1}|: \sigma_j^k = \sigma_i^k \ \vee, \tag{10}$$

$$\exists\, 0 < k < |\overline{P}_{t+1}| \ : \ \left[\left(\forall\, 0 < l < k : \sigma_i^l = \sigma_j^l\right) \wedge \ \sigma_i^k < \sigma_j^k\right]. \tag{11}$$

Here, σ_i^k denotes the distance of i from a user-predefined (kth) nearest neighbour in $\overline{P}_{t+1}$. The truncation procedure ensures realization of a uniformly distributed solution selection.

5.2.4. Crossover

In general, when employing GAs, new solutions (offspring) are created from parent solutions via use of a crossover operator. In the case considered in this study with m-SPEA2, the recombination operator is embedded in the variation process (step 6), wherein a crossover operator has been designed to realise the above function. In addition, the constraint related to the fixed number of tool duplicates invokes a repair process on the offspring that does not satisfy this constraint.

The novel crossover operator that exchanges the strongest portion of the parents' string was designed in

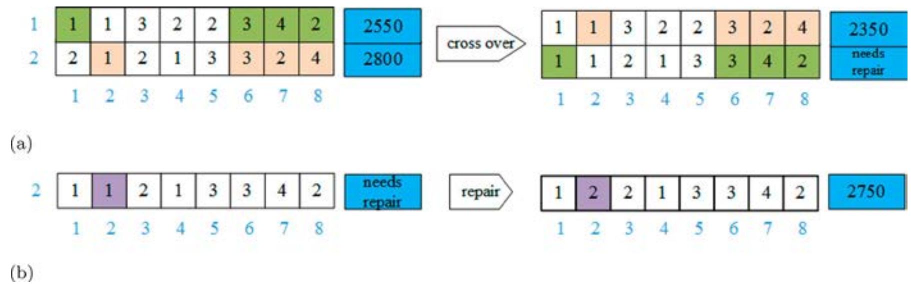


Figure 10. Customised crossover-repair operator used in the simple 8-index turret-magazine case described in Figure 1. (a) Best indexing route of the two parents is swapped. whilst maintaining other indexes constant. (b) Repair process for second offspring.

this study. Figure 10 describes the crossover operator used in the simple 8-index turret cases mentioned earlier.

The values of the first objective function for the two parents depicted on the left of Figure 10(a) equal 2550 and 2800 unit rotations. The crossover-repair procedure can be described as follows.

Step 1: Determine the shortest index-rotation path among the two parent solutions. The index positions corresponding to this path are highlighted for both parents in the figure. That is, P1 = {1,6,7,8} and P2 = {2,6,7,8}.

Step 2: Allocate the cutting tools placed at the best index locations in P2 (highlighted tools on P2) to corresponding index locations in P1 whilst retaining other tool locations. This implies that T1, T3, T2, and T4 are to be placed at index numbers 2, 6, 7, and 8, respectively, in P1 to create the offspring depicted on the right side of the figure. The second offspring can be similarly created.

Step 3: Check the constraints on the number of instances of all tools in the offspring to determine if repair is needed. Reference to the figure reveals that the second offspring comprises three T1 duplicates; however, in accordance with the problem input, only 2 instances of T1 are required. Hence, offspring2 must be repaired to obtain a feasible solution.

Step 4: (*Repair*) Figure 10(b) illustrates the repair process. As can be understood from the figure, one of the three T1 instances (positioned on indexes 1,2 or 4) in P1 must be replaced with T2 to obtain an optimum solution. Because T1 in index 1 was active during the crossover process (was exchanged from P1), the next instance of T1, which did not participate in the crossover, can be replaced by T2 assuming it to represent a strong gene of the string.

It is interesting to observe that the crossover operator considered in this study does create two strong offspring. As observed, the fitness of both offspring (2350 and 2750) is better compared to that of their parents.

5.2.5. *Mutation*

The customised mutation operator designed in this study locally alters two strings from a parent whilst maintaining the feasibility of the mutated solution. To this end, two random tool numbers are first selected. Subsequently, the index positions corresponding to the first instances of the selected tools are swapped. For example, considering the first offspring created via the crossover operation, Figure 11 illustrates the mutation operation that can be described as follows.

Step 1: Two random numbers between 1 and 4 are chosen $\{1, 4\}$. The first instances of T1 and T4 are corresponds index numbers 1 and 8, respectively.

Step 2: Next, the positions of T1 and T4 are swapped, thereby creating the mutated offspring with a better first-objective value of 2300 unit rotations.

It must be noted that because the mutation operator does not change the number of duplicates, the mutated offspring represent a feasible solution, and a separate repair process is not required. The mutation probability (ρ_μ) can be defined as a limit on the number of crossover offspring that can undergo the mutation process.

5.2.6. *Parameter setting*

The m-SPEA2 comprises three parameters: the archive set ($\overline{N}$), population set (N), and mutation probabil-

Figure 11. Mutation operation wherein index positions of the first instance of two random tools not participating in the crossover process are swapped to creating fitter offspring.

ity (ρ_μ). The values of these parameters must be set reasonably.

The values of the second objective function (NoC) are discrete and limited between the lower and upper bounds of 0 and 45, respectively. The archive set in m-SPEA2 is designed to contain non-dominated solutions, and therefore, increasing the archive size beyond 50 does not benefit the algorithm performance. Thus, in this study, we set $\overline{N} = 50$. With this setting, the second scenario of environment selection, wherein ($|\overline{P}_{t+1}| > \overline{N}$) can never be invoked.

The values of the population set and mutation probability were tuned to $N = 150$ and $\rho_\mu = 0.2$ in accordance with the Taguchi method described in Appendix.

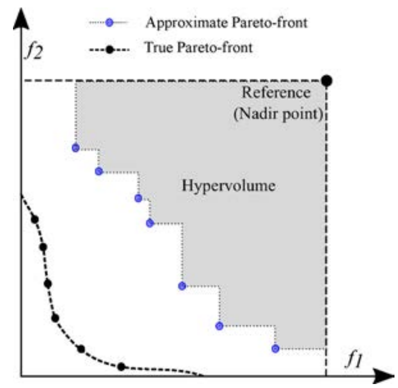


Figure 13. Hypervolume for minimisation of MOOP considered in this study with Nadir point as reference Amouzgar et al. (2019).

6. Results

The proposed algorithm for MOOTIP pertaining to the OP30 sequence of operations was scripted in MATLAB_R2019a and executed on a MacBook Pro laptop equipped with a 2.2-GHz Quad-Core Intel Core i7 processor and 16 GB RAM.

The initial population P_0 and archive (external) set $\overline{P}_0$ were generated to ensure the realization of well-distributed NoC solutions. Figure 12 depicts 200 random solutions ($|P_0| + |\overline{P}_0| = 200$) generated for one of the m-SPEA2 runs.

The algorithm was run 10 times with each iteration starting with a new random initial population. The criteria to stop the execution of the algorithm comprised two different metrics: the number of generations and convergence rate of the hypervolume indicator (Zitzler et al. 2003). The hypervolume indicator is commonly used to compare non-dominated solutions within each generation of an optimisation algorithm. The hypervolume between a reference point (Nadir point, in this study) and the obtained Pareto-optimal solutions can be represented in terms of the hypervolume indicator. Figure 13 depicts the hypervolume corresponding to the MOOP with objectives to minimised.

The m-SPEA2 algorithm was allowed to run until either the number of generations equalled its threshold value (1000) or no significant change was observed in the standard deviation of the hypervolume indicators during the last 100 generations; i.e. $STD(HV) < \epsilon$, wherein ϵ has a very small value.

All 10 instances of non-dominated solutions obtained in this study (from the 10 runs) were stored within a

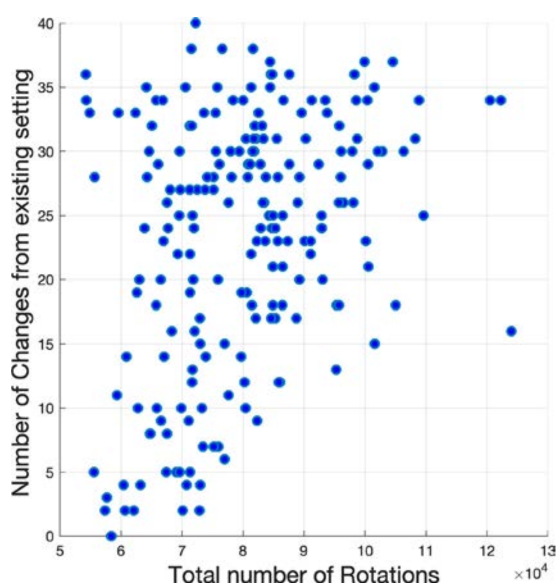


Figure 12. Two-hundred random initial solutions generated by executing an m-SPEA2 run.

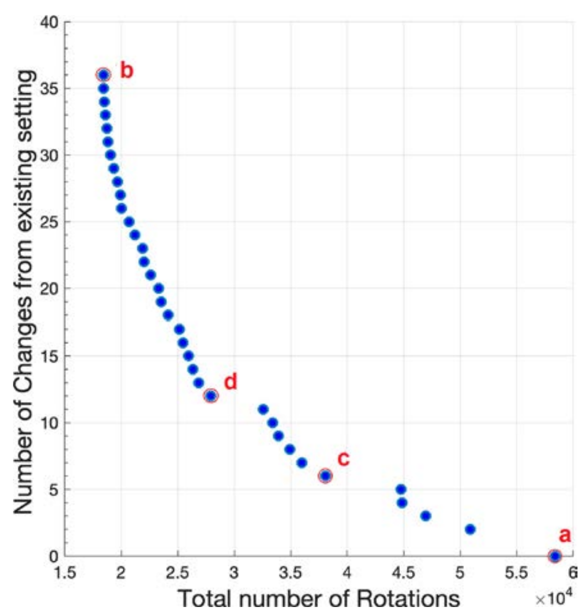


Figure 14. Non-dominated solutions in the objective space forming the Pareto front.

set, and the fitness values of all solutions in the set were calculated in accordance with the fitness-assignment procedure in Step 2 of the m-SPEA2.

In this study, 36 non-dominated solutions constituted the Pareto-front, as illustrated in Figure 14. Solutions *a* and *b*, marked with red circles in Figure 14 denote two extreme solutions to the problem. Solution *a* corresponds to the current tool-allocation in the factory, whereas solution *b* corresponds to the best solution in terms of NoR. Specifically, by changing the positions of 36 tools in the magazine, the total number of unit rotations can be reduced to 18,421 from their current value of 58,398. As a result, the average total number of rotations for machining a single crankshaft can be reduced from 45 to 14. In addition, by considering the machine index time (0.2 s), the non-machining time can be reduced from 9 s to 2.8 s per crankshaft. However, changing the positions of 36 tools in the turret magazine might not be a straightforward task; hence, the factory operator might be interested in other solutions that offer a substantial reduction in the non-machining time with comparatively lower NoC. Solutions *c* and *d* are two examples that represent this tradeoff. The results corresponding to solutions *a*–*d* are summarised in Table 9, whereas Figure 15 depicts corresponding tool-allocation layouts on the turret magazine. The positions of tools marked with a dark background remained unchanged with the current factory setting.

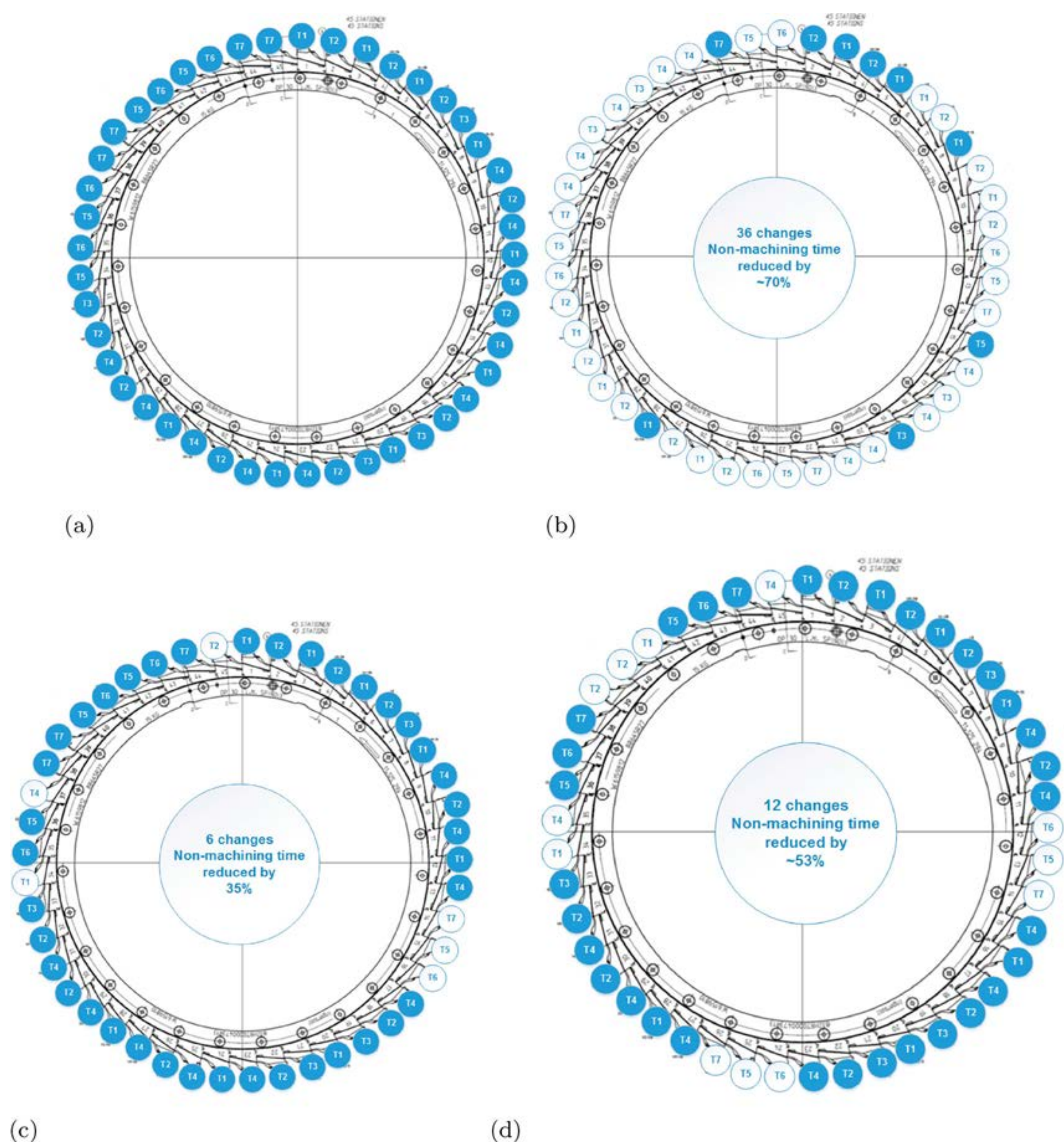


Figure 15. Tool allocation and improvement of non-machining time for 4 non-dominated solutions marked in Figure 14. Tool positions that remain unchanged compared to the current factory setting are marked by dark backgrounds. (a) solution *a*; (b) solution *b*; (c) solution *c*; (d) solution *d*.

Table 9. Results for four different non-dominated solutions marked in Figure 14.

Solution	NoC	Total no. of rotations	Rotation per part	Non-machining time per part (s)
a	0	58,398	~ 45	~ 9
b	36	18,421	~ 14	~ 2.8
c	6	38,085	~ 29	~ 5.8
d	12	27,909	~ 21	~ 4.2

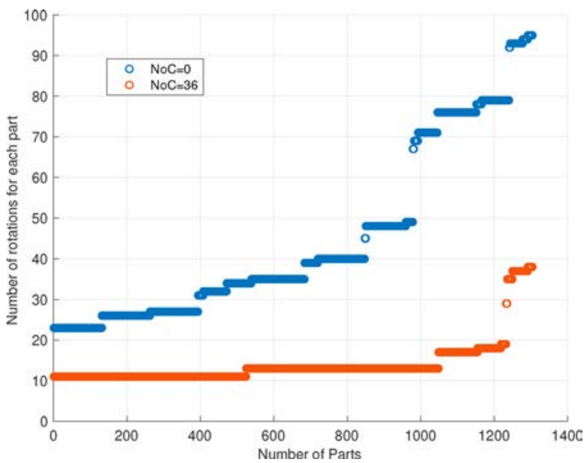


Figure 16. Number of unit rotations required to execute OP30 throughout the lifespan of 45-index turret magazine under two extreme solutions (marked by *a* and *b* in Figure 14).

Although the number of rotations per crankshaft is stated as a constant value in Table 9, Figure 16 depicts the number of unit rotations per part throughout the turret-magazine lifespan to not remain constant; instead, its value increases as different tools wear out.

The best NoR result obtained in this study (solution *b*) facilitates machining of the first two sets of 524 crankshafts each by performing 11 and 13 unit rotations, respectively, and subsequently, the NoR value increases until the last 12 crankshafts are machined by performing 38 unit rotations. In comparison, even if the current OP30 sequence is considered to use the shortest path, it requires 23 unit rotations to machine the first 131 crankshafts. This is more than twice the optimum number of rotations obtained in this study. Looking at the far end of the plot depicted in Figure 16 that corresponds to most tools being worn out, the application of solution *b* implies that the last crankshaft would be machined by performing 38 unit rotations, whereas use of the current tool-allocation strategy (solution *a*) requires 95 unit rotations.

Further in-depth analysis of the same figure reveals that the observed deviation in NoR along the *x*-axis (i.e. the lifespan of tools placed on the magazine) is lower for the results obtained in this study compared to the existing tool-allocation strategy. The lower the said deviation, the lower is the deviation in the tool-indexing time during the machining of 1304 crankshafts. This, in turn, results in better production planning.

While the non-dominated solutions in the objective space constitute the Pareto front (Figure 14), the solutions obtained in the decision space are depicted in Figure 17 as a heat map. Reference to the said figure reveals the possibility of extracting several tool-indexing sequences (indicated by black rectangles) repeated in several solutions. For instance, the solutions with lower NoC values ($NoC < 9$) (and consequently higher NoR) share the same tool-allocation sequence for indexes 1–11 and 19–27. On the other hand, solutions with NoR values ($29 < NoC < 34$) reveal identical tool-position sequences for indexes 1–8, 15–27, and 37–45.

The observed similarities in tool positioning for different solutions can be used to improve the performance of the m-SPEA2 algorithm to obtain optimum solutions. However, to explore the benefits of patterns hidden in the results, data-mining and knowledge-discovery methods must be employed. However, such efforts are beyond the scope of this study, but the same have been considered part of our future research endeavours.

Including the tool wear and finding the optimal tool allocation for the entire life span of the tools on the turret magazine in this study complements the results obtained from the most recent studies which assumed 'tools do not wear out' (Baykasoğlu and Ozsoydan 2016, 2017; Atta, Sinha Mahapatra, and Mukhopadhyay 2019; Palubeckis 2020). In addition, this study extends the TIP (as touched upon by previous studies) into a more realistic industrial problem by including more real-world assumptions and circumstances. Furthermore, defining a second objective to TIP, unlike previous studies (Amouzgar, Ng, and Ljustina 2020), which were inspired by industry and solving it by using a multi-objective optimisation algorithm, provides a tradeoff of solutions to decision-makers for better tool configuration.

Practical implications of this study include the realization of a significant (70%) reduction in the tool-indexing (from 9*s* to 2.8*s*) and cycle times, which in return improves the production-process efficiency. Further, the algorithm developed in this study can be used by production and tooling engineers as a decision-making tool to determine the optimum strategy for tool allocation on the turret magazine and other similar operations. Moreover, access to such a tool is beneficial in the event of changes in the operation settings due to design and development of new products or introduction of new tools with different lifespans.

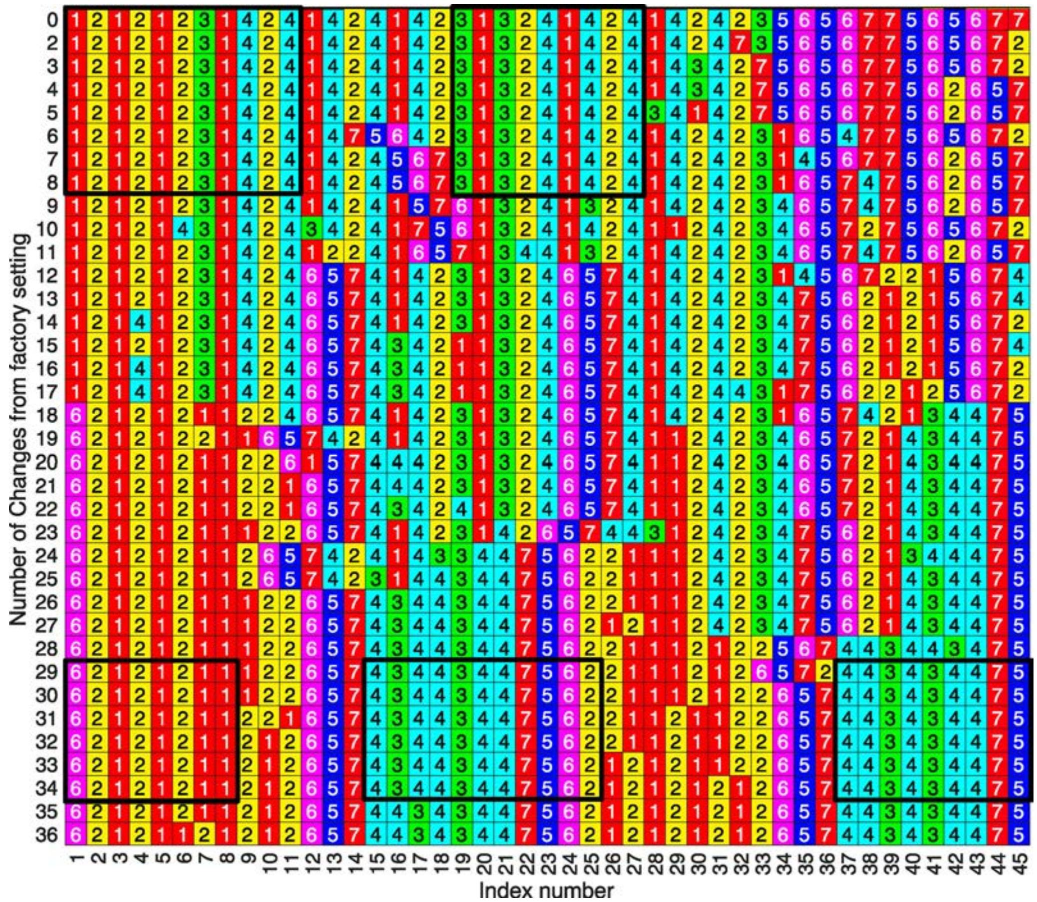


Figure 17. Non-dominated solutions in decision space illustrated as heat-map plot with each row representing a solution (optimum tool allocation). Small boxes represent magazine indexes, and tools placed at given index locations are marked by the tool number in the box. Triangles indicate similar tool-allocation sequences shared by several solutions.

7. Conclusions

This paper presents solution methodologies, the primary objective of which is to determine a set of tradeoff solutions to the multi-objective optimisation problem of indexing tools on a turret magazine with due consideration of individual tool lifespans and tool duplicates. To this end, a mathematical formulation considering tool duplicates and tool wear is proposed. Furthermore, a customised encoding–decoding scheme has been designed, and a modified GA (m-SPEA2) has been developed with the first-of-its-kind environment-selection mechanism as well as the crossover–repair and mutation operators.

As observed, compared to the current scheme of tool allocation used in the industry, the proposed optimisation algorithm provides a significantly improved tool-allocation strategy. In accordance with the optimised strategy, the total tool-indexing time can be reduced by 70% (from 9 to 2.8 s). Furthermore, the proposed m-SPEA2 generates a set of non-dominated solutions that constitute the Pareto-front in the objective space. The said front enhances the decision-making process, thereby improving the efficiency of the OP30 sequence of operations. It facilitates selection of the best-suited tool-allocation scheme based on the available resources as well as the production and tooling requirements of the factory. Moreover, optimisation results obtained in this study can be further improved by combining sequences of tool indexes observed in several non-dominated solutions. However, data-mining methods must be employed to investigate hidden patterns and extract knowledge from the obtained results in this regard.

In future research, findings of this study could be extended in several ways. From a problem perspective, the development of a solution strategy for analysing the collision detection of any solution obtained in this study before implementing in the factory can be considered. Moreover, another optimisation problem could be devised by considering other real-world factors, such as the number of tools to be optimised, as the decision variable.

Finally, other state-of-the-art algorithms with multi-objective characteristics and customised operators can be developed, and corresponding results obtained could be compared against those reported in this paper.

Acknowledgments

The authors thank their industrial partners, Volvo Car Corporation and AB Volvo, for their collaborative support during the project.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was funded by the Knowledge Foundation (KKS), Sweden, through the Profile project, Virtual Factories-Knowledge-Driven Optimisation (VF-KDO). The first author was partially funded by the Knowledge Foundation (KKS), Sweden, through the Prospket project, Predictive Modelling for Data Intensive industrial Processes and Systems (PreMoDIPS) (Stiftelsen för Kunskaps- och Kompetensutveckling) [grant number HSK2019/20].

Notes on contributors

Kaveh Amouzgar is an Assistant Professor in the Division of Industrial Engineering and Management at Uppsala University, Sweden. He is also partly affiliated with the Department of Production and Automation Engineering, University of Skövde, Sweden, where he obtained his Ph.D. degree in Informatics. His research interests are the application of Computational Intelligence, Simulation-Based Optimization and Industrial Analytics in manufacturing and production systems design and analysis. His email address is kaveh.amouzgar@angstrom.uu.se.

Amir Nourmohammadi is a postdoctoral researcher at the Department of Production and Automation Engineering, University of Skövde, Sweden, where he has been engaged in different national and international research projects in collaboration with industry. He holds a Ph.D. degree in Industrial Engineering from the Department of Industrial and Systems Engineering, Tarbiat Modares University, Tehran, Iran. His main research interests include the application of operations research and soft computing techniques in production and logistics, particularly in the era of Industry 4.0. His email address is amir.nourmohammadi@his.se.

Amos H. C. Ng is a professor in Production and Automation Engineering at the University of Skövde, Sweden. He is also a visiting professor in the Division of Industrial Engineering and Management at Uppsala University, Sweden, and the CEO of Evoma AB. He holds a Ph.D. degree in Computing Sciences and Engineering from De Montfort University, Leicester, UK. His main research interest lies in applying multi-disciplinary simulation, multi-objective optimisation, and AI-based prescriptive analytics for manufacturing/service/health-care systems design, analysis, and improvement. He can be reached by email addresses: amos.ng@his.se and amos.ng@evoma.se.

ORCID

Kaveh Amouzgar http://orcid.org/0000-0001-7534-0382
Amir Nourmohammadi http://orcid.org/0000-0001-6280-1848
Amos H.C. Ng http://orcid.org/0000-0003-0111-1776

References

Ahmadi, E., B. Goldengorin, G. A. Süer, and H. Mosadegh. 2018. “A Hybrid Method of 2-tsp and Novel Learning-based Ga for Job Sequencing and Tool Switching Problem.” *Applied Soft Computing*65: 214–229.

Amouzgar, K., S. Bandaru, T. Andersson, and A. H. C. Ng. 2018. “A Framework for Simulation-based Multi-objective Optimization and Knowledge Discovery of Machining Process.” *The International Journal of Advanced Manufacturing Technology* 98 (9–12): 2469–2486.

Amouzgar, K., S. Bandaru, T. Andersson, and A. H. C. Ng. 2019. “Metamodel-based Multi-objective Optimization of a Turning Process by Using Finite Element Simulation.” *Engineering Optimization* 52 (7): 1–18.

Amouzgar, K., A. H. Ng, and G. Ljustina. 2020. “Optimizing Index Positions on Cnc Tool Magazines Considering Cutting Tool Life and Duplicates.” *Procedia CIRP* 93: 1508–1513.

Amouzgar, K., A. Rashid, and N. Stromberg. 2013. “Multi-Objective Optimization of a Disc Brake System by Using Spea2 and RBFN.” In ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection. Portland, Oregan, USA.

Atta, S., P. R. Sinha Mahapatra, and A. Mukhopadhyay. 2019. “Solving Tool Indexing Problem Using Harmony Search Algorithm with Harmony Refinement.” *Soft Computing* 23 (16): 7407–7423.

Baykasoğlu, A., and T. Dereli. 2004. “Heuristic Optimization System for the Determination of Index Positions on CNC Magazines with the Consideration of Cutting Tool Duplications.” *International Journal of Production Research* 42 (7): 1281–1303.

Baykasoğlu, A., and F. B. Ozsoydan. 2016. “An Improved Approach for Determination of Index Positions on CNC Magazines with Cutting Tool Duplications by Integrating Shortest Path Algorithm.” *International Journal of Production Research* 54 (3): 742–760.

Baykasoğlu, A., and F. B. Ozsoydan. 2017. “Minimizing Tool Switching and Indexing Times with Tool Duplications in Automatic Machines.” *The International Journal of Advanced Manufacturing Technology* 89 (5-8): 1775–1789.

Baykasoğlu, A., and F. B. Ozsoydan. 2018. “Minimisation of Non-machining Times in Operating Automatic Tool Changers of Machine Tools Under Dynamic Operating Conditions.” *International Journal of Production Research* 56 (4): 1548–1564.

Crama, Y., L. S. Moonen, F. C. R. Spieksma, and E. Talloen. 2007. “The Tool Switching Problem Revisited.” *European Journal of Operational Research* 182 (2): 952–957.

Dang, Q. V., T. van Diessen, T. Martagan, and I. Adan. 2021. “A Matheuristic for Parallel Machine Scheduling with Tool Replacements.” *European Journal of Operational Research* 291 (2): 640–660.

da Silva, T., A. Chaves, and H. Yanasse. 2020. “A New Multicommodity Flow Model for the Job Sequencing and Tool Switching Problem.” *International Journal of Production Research* 1–16. https://doi.org/10.1080/00207543.2020.1748906.

Deb, K. 2001. *Multi-objective Optimization Using Evolutionary Algorithms*. Vol. Vol. 16. New York, NY, United States: John Wiley & Sons.

Dereli, T., A. Baykasoğlu, N. Gindy, and I. Filiz. 1998. “Determination of Optimal Turret Index Positions by Genetic Algorithms.” In *Proceedings of 2nd International Symposium on Intelligent Manufacturing Systems*, Vol. 2, 743–750.

Dereli, T., and I. H. Filiz. 2000. “Allocating Optimal Index Positions on Tool Magazines Using Genetic Algorithms.” *Robotics and Autonomous Systems* 33 (2): 155–167.

Dijkstra, E. W. 1959. “A Note on Two Problems in Connexion with Graphs.” *Numerische Mathematik*1 (1): 269–271.

Furrer, M., and T. Mütze. 2017. “An Algorithmic Framework for Tool Switching Problems with Multiple Objectives.” *European Journal of Operational Research* 259 (3): 1003–1016.

Gopala, A., and K. K. M. Rao. 2006. “Optimal Allocation of Index Positions on Tool Magazines Using An Ant Colony Algorithm.” *The International Journal of Advanced Manufacturing Technology* 30: 717–721.

Hertz, A., G. Laporte, M. Mittaz, and K. E. Stecke. 1998. “Heuristics for Minimizing Tool Switches when Scheduling Part Types on a Flexible Machine.” *IIE Transactions* 30 (8): 689–694.

Keung, K. W., W. H. Ip, and T. C. Lee. 2001. “The Solution of a Multi-objective Tool Selection Model Using the GA Approach.” *International Journal of Advanced Manufacturing Technology* 18 (11): 771–777.

Konak, A., S. Kulturel-Konak, and M. Azizoğlu. 2008. “Minimizing the Number of Tool Switching Instants in Flexible Manufacturing Systems.” *International Journal of Production Economics* 116 (2): 298–307.

Laporte, G., J. J. Salazar-González, and F. Semet. 2004. “Exact Algorithms for the Job Sequencing and Tool Switching Problem.” *IIE Transactions (Institute of Industrial Engineers)* 36 (1): 37–45.

Loiola, E. M., N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. 2007. “A Survey for the Quadratic Assignment Problem.” *European Journal of Operational Research* 176 (2): 657–690.

Mecler, J., A. Subramanian, and T. Vidal. 2021. “A Simple and Effective Hybrid Genetic Search for the Job Sequencing and Tool Switching Problem.” *Computers and Operations Research* 127: 105153. https://doi.org/10.1016/j.cor.2020.105153.

Ménard, M. A., C. G. Quimper, and J. Gaudreault. 2019. “Pre-optimizing Tools Positions for Turrets-based Cnc Machines when Facing Arbitrary Sequences of Production.” *IFAC-Papers OnLine* 52 (13): 1180–1185. 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019.

Ng, A. H. C., F. Siegmund, and K. Deb. 2018. “Reference Point Based Evolutionary Multi-objective Optimization with Dynamic Resampling for Production Systems Improvement.” *Journal of Systems and Information Technology* 20: 489–512.

Nourmohammadi, A., H. Eskandari, M. Fathi, and A. H. C. Ng. 2019. “Integrated Locating In-house Logistics Areas and Transport Vehicles Selection Problem in Assembly Lines.” *International Journal of Production Research* 59 (2): 598–616.

Paiva, G. S., and M. A. M. Carvalho. 2017. “Improved Heuristic Algorithms for the Job Sequencing and Tool Switching Problem.” *Computers & Operations Research* 88: 208–219.

Palubeckis, G. 2020. “An Approach Integrating Simulated Annealing and Variable Neighborhood Search for the Bidirectional Loop Layout Problem.” *Mathematics* 9 (1): 5.

Pehrsson, L., A. H. C. Ng, and J. Bernedixen. 2016. “Automatic Identification of Constraints and Improvement Actions in Production Systems Using Multi-objective Optimization and Post-optimality Analysis.” *Journal of Manufacturing Systems* 39: 24–37.

Raduly-Baka, C., and O. S. Nevalainen. 2015. “The Modular Tool Switching Problem.” *European Journal of Operational Research* 242 (1): 100–106.

Rao, Y., R. Meng, J. Zha, and X. Xu. 2019. “Bi-objective Mathematical Model and Improved Algorithm for Optimisation of Welding Shop Scheduling Problem.” *International Journal of Production Research*00: 1–17.

Reddy, B., and C. Rao. 2006. “A Hybrid Multi-objective Ga for Simultaneous Scheduling of Machines and Agvs in Fms.” *International Journal of Advanced Manufacturing Technology* 31 (5–6): 602–613. cited By 90.

Rezaei, J., and M. Davoodi. 2012. “A Joint Pricing, Lot-sizing, and Supplier Selection Model.” *International Journal of Production Research* 50 (16): 4524–4542.

Salonen, K., C. Raduly-Baka, and O. S. Nevalainen. 2006. “A Note on the Tool Switching Problem of a Flexible Machine.” *Computers and Industrial Engineering* 50 (4): 458–465.

Schwerdfeger, S., and N. Boysen. 2017. “Order Picking Along a Crane-supplied Pick Face: The Sku Switching Problem.” *European Journal of Operational Research* 260 (2): 534–545.

Shen, X. N., and X. Yao. 2015. “Mathematical Modeling and Multi-objective Evolutionary Algorithms Applied to Dynamic Flexible Job Shop Scheduling Problems.” *Information Sciences* 298: 198–224.

Soares, L. C. R., and M. A. M. Carvalho. 2020. “Biased Random-key Genetic Algorithm for Scheduling Identical Parallel Machines with Tooling Constraints.” *European Journal of Operational Research* 285 (3): 955–964.

Soares, L. C., J. A. Reinsma, L. H. Nascimento, and M. A. Carvalho. 2020. “Heuristic Methods to Consecutive Block Minimization.” *Computers & Operations Research* 120: 104948.

Soolaki, M. 2013. “A Novel 0-1 Linear Integer Programming Model and Nsga-ii for Dynamic Machine Tool Selection and Operation Allocation Problem in Fms.” *International Journal of Industrial and Systems Engineering* 14 (4): 463–483.

Taguchi, G., S. Chowdhury, and Y. Wu. 2005. *Taguchi's Quality Engineering Handbook*. Hoboken, NJ, USA: Wiley.

Tang, L., L. Zheng, H. Cao, and N. Huang. 2016. “An Improved Multi-objective Genetic Algorithm for Heterogeneous

Coverage Rfid Network Planning." *International Journal of Production Research* 54 (8): 2227–2240.
Tirkolaee, E. B., A. Goli, A. Faridnia, M. Soltani, and G. W. Weber. 2020. "Multi-objective Optimization for the Reliable Pollution-routing Problem with Cross-dock Selection Using Pareto-based Algorithms." *Journal of Cleaner Production* 276: 122927.
Tseng, H. E., M. H. Chen, C. C. Chang, and W. P. Wang. 2008. "Hybrid Evolutionary Multi-objective Algorithms for Integrating Assembly Sequence Planning and Assembly Line Balancing." *International Journal of Production Research* 46 (21): 5951–5977.
Wang, J., X. C. Xi, L. Qin, Y. O. Zhang, and W. S. Zhao. 2020. "Non-productive Time Optimization for 5-axis Edm Drilling Using Hvnts Algorithm." *International Journal of Production Research* 1–15. https://doi.org/10.1080/00207543.2020.1779961.
Wang, S., H. Zou, and S. Wang. 2020. "A Tabu-ga-based Parallel Machine Scheduling with Restrained Tool Resources." *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. https://doi.org/10.1177.
Zhang, C., X. Dong, X. Wang, X. Li, and Q. Liu. 2010. "Improved Nsga-ii for the Multi-objective Flexible Job-shop Scheduling Problem." *Journal of Mechanical Engineering* 46 (11): 156–164.
Zhang, L., Z. Li, G. Królczyk, D. Wu, and Q. Tang. 2019. "Mathematical Modeling and Multi-attribute Rule Mining for Energy Efficient Job-shop Scheduling." *Journal of Cleaner Production*241: 118289.
Zitzler, E., M. Laumanns, and L. Thiele. 2001. "Spea2: Improving the Strength Pareto Evolutionary Algorithm." TIK-report 103
Zitzler, E., L. Thiele, M. Laumanns, C. M. Fonseca, and V. D. F. Grunert. 2003. "Performance Assessment of Multiobjective Optimizers: An Analysis and Review." *IEEE Transactions on Evolutionary Computation* 7 (2): 117–132.

Appendix

The Taguchi design has been widely used in several applications ranging from industrial experiments to parameter setting for algorithms. The latter was considered in this study because appropriate parameter levels can significantly enhance the performance of optimisation algorithms (Tirkolaee et al. 2020). To determine the values of the two parameters (N and ρ_μ), the Taguchi method was applied to an $L_9(3^2)$ experimental design (Taguchi, Chowdhury, and Wu 2005) over the 3 parameter levels listed in Table A1.

Table A1. Levels of m-SPEA2 parameters used in Taguchi design.

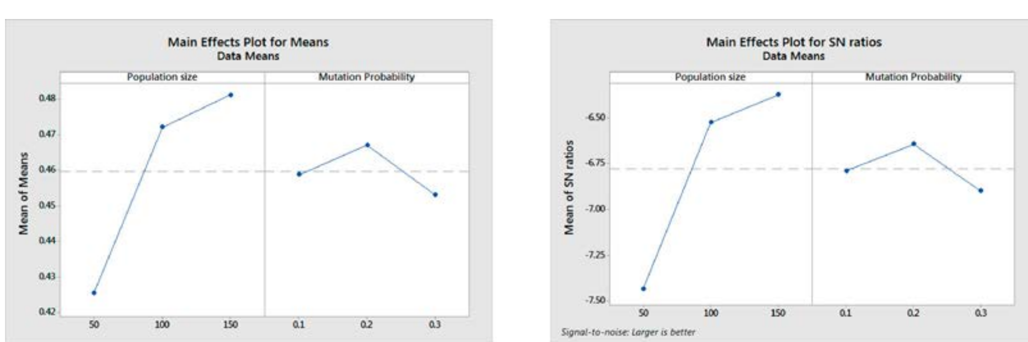
Parameter levels	N	ρ_μ
1	50	0.1
2	100	0.2
3	150	0.3

The Taguchi design employs the design of experiments approach to determine appropriate parameter levels. The algorithm performance at these parameter levels becomes insensitive to variations in uncontrollable factors without disregarding them (Nourmohammadi et al. 2019). This is realised by maximising the signal-to-noise ratio (SN) and considering it to be representative of the performance variation of the algorithm. The means of the objective function are maximised or minimised depending on the nature of the problem. In this study, because we aim to maximise the distance of non-dominated solutions from the Nadir point calculated by hypervolume indicators, Equation (A1) was employed to calculate and maximise the corresponding SN ratio (the larger the better).

$$SN_{(\text{larger is better})} = -10 \times \log_{10}\left(\frac{\sum_{e=1}^{NE} \frac{1}{HV_e^2}}{NE}\right) \tag{A1}$$

Here, NE denotes the number of experiments, and HV_e denotes the hypervolume indicator during the e-th experiment ($e = 1, \ldots, NE$). Figure A1 depicts the Means and SN-ratio plots corresponding to different combinations of parameter levels.

According to Figure A1, values of the optimum population size and mutation probability were set to $N = 150$ and $\rho_\mu = 0.2$ in this study.



(a) (b)

Figure A1. Plots obtained via Taguchi analysis for (a) Means; (b) SN ratio.