The International Journal of Advanced Manufacturing Technology (2018) 98:2469–2486
https://doi.org/10.1007/s00170-018-2360-8

ORIGINAL ARTICLE

A framework for simulation-based multi-objective optimization and knowledge discovery of machining process

Kaveh Amouzgar[1] · Sunith Bandaru[1] · Tobias Andersson[1] · Amos H. C. Ng[1]

Received: 30 November 2017 / Accepted: 15 June 2018 / Published online: 10 July 2018

Abstract

The current study presents an effective framework for automated multi-objective optimization (MOO) of machining processes by using finite element (FE) simulations. The framework is demonstrated by optimizing a metal cutting process in turning AISI-1045, using an uncoated K10 tungsten carbide tool. The aim of the MOO is to minimize tool-chip interface temperature and tool wear depth, that are extracted from FE simulations, while maximizing the material removal rate. The effect of tool geometry parameters, i.e., clearance angle, rake angle, and cutting edge radius, and process parameters, i.e., cutting speed and feed rate on the objective functions are explored. Strength Pareto Evolutionary Algorithm (SPEA2) is adopted for the study. The framework integrates and connects several modules to completely automate the entire MOO process. The capability of performing the MOO in parallel is also enabled by adopting the framework. Basically, automation and parallel computing, accounts for the practicality of MOO by using FE simulations. The trade-off solutions obtained by MOO are presented. A knowledge discovery study is carried out on the trade-off solutions. The non-dominated solutions are analyzed using a recently proposed data mining technique to gain a deeper understanding of the turning process.

Keywords Machining · Turning simulation · Multi-objective optimization · Cutting parameters · Tool geometry

1 Introduction

Optimization of manufacturing processes has been an attractive topic for research. A considerable number of attempts have been made to optimize different manufacturing processes [1–4]. Machining is one of the five main groups of manufacturing processes, along with casting, forming, powder metallurgy, and joining [5]. It involves removing parts of a metal workpiece to create a desired design. The removed parts are extracted in the form of chips. Turning is one of two major groups of machining cutting processes, i.e., the traditional machining processes. The other group which is modern machining processes includes electrical discharge machining (EDM) and abrasive water jet (AWJ). Since John Wilkinson introduced the cannon-borring machine in 1775, the machining process has undergone significant development. Optimization of metal cutting process is one of the major fields that attracted many researchers' attention to improve the quality of machining products [6]. In the review paper by Mukherjee and Ray [6], the focus is on optimization tools and techniques grouped into conventional and non-conventional techniques. Optimization of machining parameters by using several different optimization techniques has been studied in a literature review paper by Aggarwal and Singh [7]. In most of the studies, the optimization process is based on the actual process (experiment), mathematical material model, or some form of simulation (e.g., finite element method, FEM). The researchers applied various optimization techniques on an approximated mathematical model or on a cost and computational expensive experiment or computer simulation. However, the optimization studies including two or more objective, i.e., multi-objective optimization (MOO), are limited. In a state-of-the-art study by Tutum and Hattel [8], MOO of manufacturing process based on thermo-mechanical simulations is thoroughly reviewed.

The approaches towards solving a multi-objective optimization problem (MOOP) are classified into two groups [9]: (1) the preference-based approach or *a priori* technique and (2) the ideal approach or *posteriori* technique. In the first approach, by using some higher-level information a preference vector transforms the MOOP into a

✉ Kaveh Amouzgar
kaveh.amouzgar@his.se

[1] School of Engineering Science, University of Skövde, Skövde, Sweden

single-objective optimization. The preference-based approach is used in two research papers [10, 11], to study multi-objective optimization of cutting parameters. The ideal approach towards multi-objective optimization is to obtain a set of solutions in the form of Pareto-optimal solutions, where the desired solution is selected according to some higher-level information of the problem. In this approach, the decision maker will have a better understanding of the decision variables, objectives, and the relations between the two. In addition, it provides the freedom to analyze the results before selecting a preferred solution. Evolutionary algorithm (EA), due to their characteristic of using a population of solution that evolve in each generation, is well suited for the ideal approach in solving multi-objective optimization problems. However, it should be pointed out that in most practical applications, the solutions obtained by EAs may not be truly Pareto-optimal. Moreover, it is impossible to verify whether the solutions are Pareto-optimal when the objectives and constraints lack analytical forms. Therefore, it is customary in optimization literature to refer to solutions obtained using EAs a *trade-off solutions*.

The ideal approach was incorporated in [12], where they optimized the production rate and tool life in turning process by using an algorithm based on micro-GA. The objectives were generated by mathematical equations in terms of cutting depth, feed, and speed. Recent applications (2007–2011) of evolutionary optimization techniques in optimization of machining parameters have been reviewed by Yusup et al. [13]. In one study, Sultana and Dhar [14] minimized cutting temperature and cutting force in turning AISI-4320 steel by using a MOO algorithm based on GA subjected to keeping the surface roughness less than a constant value. Experimental study along with predictive models was implemented under response surface method (RSM). The cutting variables they considered were cutting speed, feed rate, pressure, and flow rate of high-pressure coolant.

Another study by Umer et al. [15] minimizes cutting force and tool-workpiece interface temperature by using three different surrogate models (RSM, radial basis functions (RBF), and neural networks) and a multi-objective optimization genetic algorithm (MOGA2) implemented in MODEFRONTIER. The objectives were obtained by modeling the oblique turning process in Abaqus FEM software. In another paper [16], cutting force and tool wear of a high-performance micro-milling process were generated by running 2D finite element simulations (in DEFORM-2D software). They were used as inputs for optimizing tool path and process parameters along with burr formation and surface roughness data, extracted from experiments. The multi-objective particle swarm optimization (MOPSO) techniques was employed for the optimization.

The evolving population of EAs during the optimization generations requires numerous objective function evaluations. In most of the studies, the objectives are extracted from the actual experiment, and consequently, performing several hundreds of turning operation experiments is practically impossible. Therefore, computer simulations predominantly FE ones shall be incorporated in MOO of turning operation. However, simulating a turning operation to represents the actual process to a certain extent requires computationally expensive FE models. The optimization algorithm can be integrated with metamodels of the FE simulation to reduce the optimization time. Nevertheless, there are two sources of approximation error in modeling the actual turning operation: one from simulating the operation with FE method and the other by modeling the FE simulation with metamodels. Therefore, being able to perform a MOO of a turning operation by using FE simulations efficiently and reducing the contiguity of adopting metamodels is an interesting area for research.

In this paper, we only deal with the second source of error because the purpose here is only to demonstrate, present, and analyze the developed framework. However, it is worth pointing out that without a proper analysis of modeling error, the trade-off solutions obtained in this study should not be considered as Pareto-optimal.

The above described multi-objective optimization of the turning process helps us find trade-off solutions with respect to the multiple objectives. Knowledge of these solutions is essential for informed decision making. However, often the decision-making process usually ends with the selection of a single trade-off solution for practical implementation. To gain a deeper understanding of the turning process and help identify or create high-performing solutions for similar applications in the future, it is essential to analyze the trade-off solutions using data mining and machine learning techniques. Such a process has been referred to as *innovization* in the past [17], which is a portmanteau of "innovation through optimization," implying that analysis of solutions from optimization can lead to innovations in constructing high-performing initial solutions for similar problems in the future.

Innovization was initially proposed as a manual process of visually identifying correlations and performing regression to obtain explicit relations between various combinations of variables and objectives. This procedure, though tedious and largely subjective, showed that innovations can indeed be made. For example, in [18], the authors show that all Pareto-optimal solutions of the brushless DC motor design use the same type of winding. With this knowledge, a future design can start with this winding type for all candidate solutions, thus intelligently reducing the search space. Following the proof-of-principle of innovization, data mining and machine learning techniques have been used to automate the process of post-optimal knowledge discovery. The techniques include classification trees [19], regression trees

[20, 21], rough sets [22, 23], association rule mining [24], self-organizing maps [25–27], and genetic programming [28] for automated innovization [29]. A survey of these and several visualization techniques can be found in [30].

In this paper, we developed a framework for performing an automated multi-objective optimization of machining processes by using FE simulations. This framework eliminates the time spent in the manual tasks during FE model setup and MOO and enables parallel computing. Both will expedite performing MOO by using FE simulations. The framework is presented by optimizing a simple metal cutting process in a turning operation by using DEFORM-2D software. The geometrical and process parameters, e.g., clearance and rake angel, cutting edge radius, and cutting speed and feed rate, respectively, are the variables and material removal rate, tool-chip interface temperature, and wear depth are the objectives. Genetic algorithm is used for the optimization and two of the objectives are computed from running the turning simulation in DEFORM-2D. The trade-off solutions are obtained and explicit knowledge is extracted from them using a recently proposed algorithm.

2 Framework

In this work, a framework is developed and presented for automated multi-objective optimization of metal cutting processes, specifically turning process. The framework is initiated by a parametrized CAD model. The model contains all the design requirements including geometrical bounds, constraints, shape, and topology. The parametric model is used to generate different designs of DoE as solid CAD model. This property will enable the implementation of shape or topological optimization with machining simulations. The framework is based on interactions between different pieces of a block, which we call modules. Each module consists of input and output parameters and a method which processes the inputs and generates the outputs. Here, the methods are different software, e.g., CAD software, meshing software, computing, and FE simulation software. The modules' interaction are automated and controlled by an automation system. The framework and the consisting modules, namely, geometry generator, mesh generator, process KEY file generator, FE file generator, FE simulation, result extractor, multi-objective optimizer, and knowledge extractor, are illustrated in Fig. 1. In the next subsections, the modules and the automated system are described in detail.

2.1 Geometry generator

The geometry generator creates one or several designs of different parts in a manufacturing process based on the design requirements, limits, and constraints in a CAD model format. In this study, FreeCAD is used to create the geometry of the cutting tool and workpiece. Shape, rake angle, clearance angle, and nose radius of cutting tool, and dimension and shape of the workpiece are examples

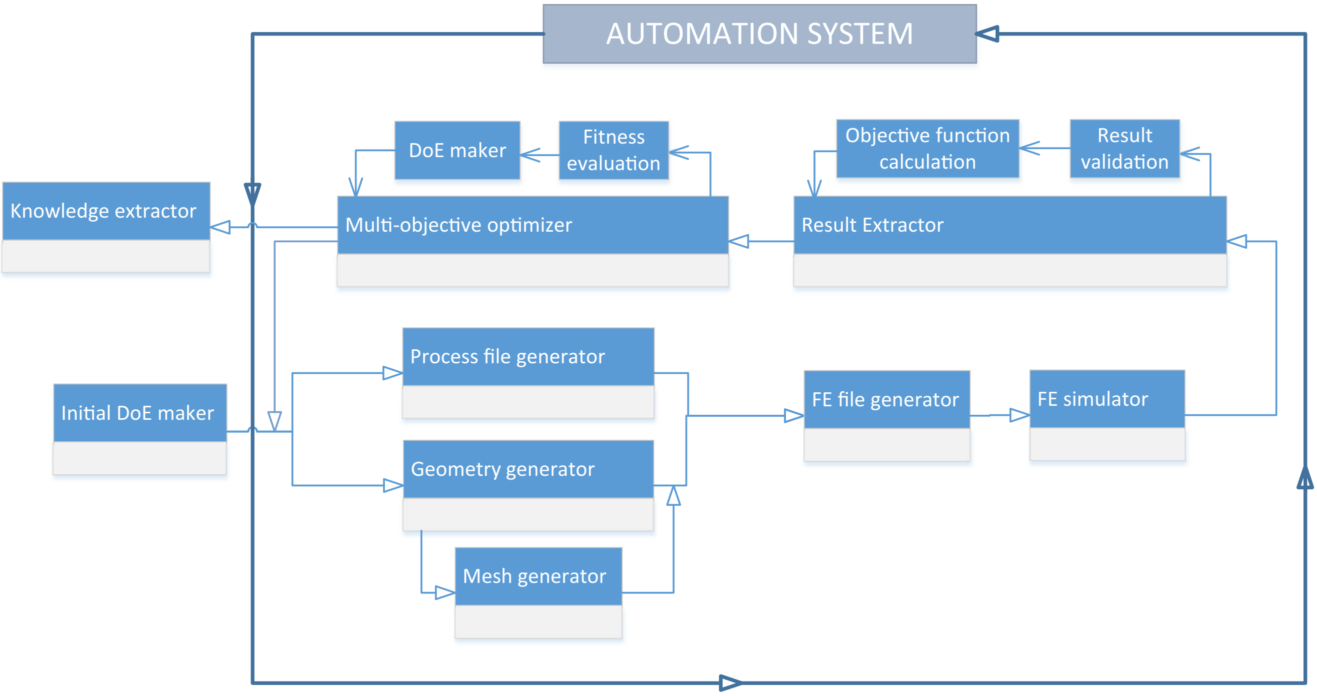


Fig. 1 Overview of the framework

of the parameters that can be set by geometry generator. Furthermore, the relative position of the tool and the workpiece and some of the process parameters such as depth of cut and feed rate can be defined in the CAD model. The geometry generator exports the design to a suitable file format, e.g., IGES or STL. Since DEFORM-2D software accepts a specific type of GEO file, the design is exported as IGES file, which is read by an IGES file reader in DEFORM-2D and exported as a file format (OUT) that is closely similar to the accepted GEO file. However, the OUT file must be slightly modified to be accepted by DEFORM-2D as a geometry.

2.2 Mesh generator

The exported geometry by the geometry generator is imported to the mesh generator. The mesh generator module meshes different parts based on predefined parameters such as type of element, number of elements, and mesh density. In this study, the final GEO file generated in the previous module is imported to mesh generator of DEFORM-2D, and the mesh file is created. Nevertheless, the mesh can be created by other software such as Gmsh, or for 3D parts, the geometry and the mesh can be imported as a single STL file.

2.3 KEY file generator

This module generates or modifies the already existing files required by the next module, i.e., the FE file generator, in the form of KEY file. The KEY file is a text-based file which contains different keywords and the corresponding values to that keyword. Material parameters including mechanical and thermal properties are set in this module. Process parameters, e.g., cutting speed, can also be defined here.

2.4 FE file generator

For running a FE simulation, boundary conditions, contact, FE step data, and type of FE simulation are required to be defined. Type of FE simulation and steps and stopping criteria can be defined by using the keywords in a KEY file. However, to set the boundary and contact conditions, it is necessary to identify and parametrize the generated mesh. Therefore, the mesh created in the mesh generator is imported in Matlab™ and triangulated to identify all aspects of the geometry, e.g., edges and corners. Now, boundary and contact conditions can be defined to the related elements and nodes of the mesh as a KEY file. Finally, all the KEY files are combined. The FE file that can be imported by DEFORM-2D in the form of DATABASE (DB) format is generated and exported.

2.5 FE simulation

DEFORM-2D/3D is the software used for simulating the turning process. All essential inputs to run a DEFORM-2D simulations are included in the database file generated by the previous module. After the simulation is run and reaches the specified stopping criteria, the final DB file is exported to result extractor. The 2D turning process implemented in this study is described with details in Section 3.

2.6 Result extractor

The final DB file is used to extract the results and calculate the objective functions. Result extractor will extract the required KEY file from any step of the DB file and calculates the objective function accordingly. In addition, this module validates the simulation result to be compliant to the constraints and experiment settings. For instance, in this study, the cutting length is the stopping criteria which must be checked. Therefore, the last step of the simulation is extracted as a KEY file. The corresponding mesh of the cut workpiece is imported into Matlab™ and the coordinates of the corner points, as depicted in Fig. 2, are compared to a pre-specified distance (stopping criteria) to validate the results. If difference is larger than a threshold, the objective function values related to that specific DoE are penalized.

2.7 Multi-objective optimizer

Multi-objective optimization (MOO), unlike single-objective optimization methods require several evaluations of each objective within the design space. There are several methods for solving multi-objective optimization problem (MOOP); however, evolutionary algorithms (EAs) are one of the most studied methods in this field. The EA methods start with a population that evolve through generations by employing evolutionary operators, e.g., crossover and mutation, mimicking the evolution present in nature. In this study, we employ an in-house implementation of the well-known strength Pareto EA (SPEA2) as the MOO solver with the following parameter values:

- Initial population size, 40
- External set population size, 40
- Cross over probability, 0.8
- SBX Crossover distribution index [31], 2

The method is proven to have some advantages compared to other existing techniques for problems with more than two objectives [32]. The SPEA2 has previously been used for optimizing real-world engineering problems [33, 34].

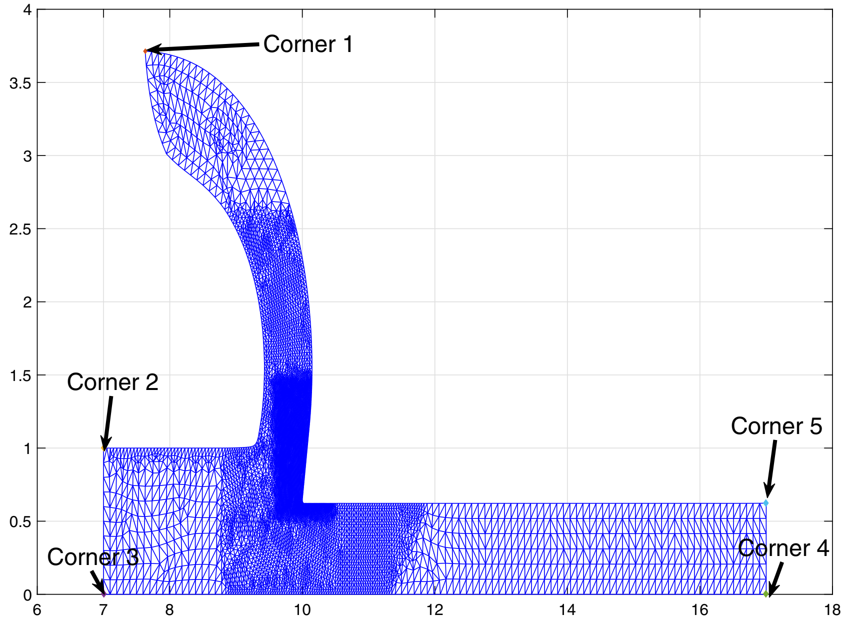


Fig. 2 Final shape of the workpiece (chip) after 7 mm of cut, and identification of the corner points

2.8 Knowledge extractor

In this module, we use the recently proposed flexible pattern mining (FPM) [35] approach to extract rules that distinguish the obtained trade-off solutions from the other feasible solutions obtained during the optimization process. FPM works on the same principle as *frequent itemset mining* that is used often on market basket data. Given a list of transactions containing multiple sets of items purchased by different customers at different instances, frequent itemset mining can identify sets of items that are often bought together. FPM treats the solutions of multi-objective optimization as “customers” and the variables as “transactions” and uses the *apriori algorithm* [36] to find clusters of solutions that take the *same values* for a subset of variables. For ordinal or continuous variables, FPM can also find clusters of solutions that take the *same range of values* for a subset of variables. These ranges of values for any variable x can be represented using rules such as $x > a$ or $x < b$.

2.9 Automation system

The automation system as shown in Fig. 3, is a cloud-based Windows script that controls and distributes the modules to computational devices. As stated in MOO section, objective evaluation of the population (DoE) in each generation of the optimizer is computationally expensive. This is due to the high computational time of FE simulations that will increase with more detailed models. Therefore, one approach of resolving this issue is to distribute the FE simulations (jobs) to several FE solvers across many computational workstations. Figure 3 shows the allocation of modules across parallel workstations (PWs) by the main workstation (MW). The PWs are basically FE solvers and result extractors. The results are stored in a shared cloud, when all the simulation results of the DoEs in an optimization generation are extracted the MW analyses the results and runs the multi-objective optimizer. The optimizer generates the DoE for the next iteration as a set of parameters. The geometry, meshm, and KEY file generators create the required files based on the new set of parameters and FE file generators create the DB files for all the population. Afterwards, the DoE distributor allocates the FE files to different PWs, based on their computational power. This loop continues until the optimizer reaches the specified stopping criteria.

The automation system’s executable files and the script files used to generate the FE are appended to the paper. The detailed procedure of the practical implementation and the description of the appended files are discussed in the Appendix.

3 FE simulation of turning process

To illustrate how the developed framework is applied to a MOO of a turning operation, a multi-objective optimization of a 2D turning operation by using FE simulation and DEFORM-2D software is carried out. The FE model is based on Lagrangian method. Figure 4 illustrates a schematic diagram of a 2D cutting operation. The variables considered in the study are the tool geometry parameters including clearance angle γ, rake angle α, and tool cutting edge radius r. Cutting speed v and feed rate f are the the process parameter that are indicated in Fig. 4. We

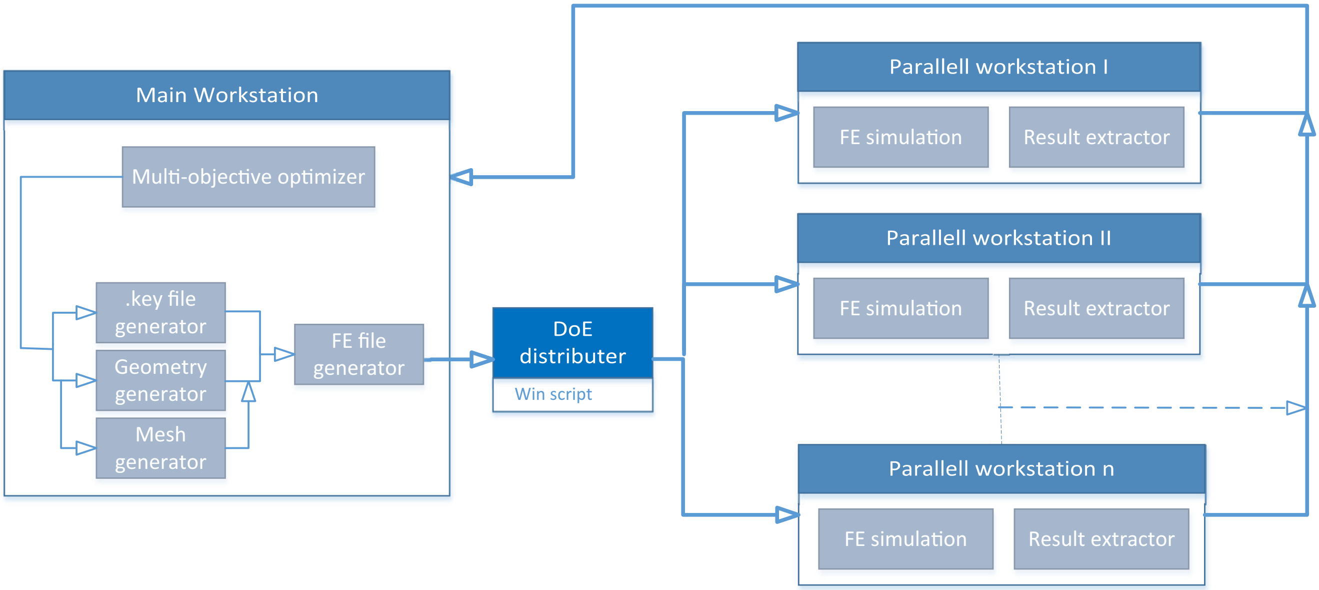


Fig. 3 Overview of the automation system

considered three objectives, which were extracted from the simulation after 7 mm of cut length, e.g., minimizing tool wear depth, minimizing the tool-chip interface maximum temperature and maximizing the material removal rate (MRR). MRR is calculated by

$$\mathrm{MRR} = vfd \tag{1}$$

where v (mm/sec) is the cutting speed, f (mm/rev) is the feed rate, and d (mm) is the depth of cut. The depth of cut is in the third dimension; thus, it is kept constant. A summary of the variables and objecitves are shown in Table 1. Table 2 shows the upper and lower limits of the variables.

3.1 Boundary conditions

The tool is fixed and the cutting is done by the movement of the workpiece towards the tool. The vertical speed of the workpiece is set to zero (fixed in y direction), and horizontal speed (in x direction) is assigned to the nodes on the bottom edge of the workpiece. The two edges of the workpiece in contact with the tool are in heat exchange with the environment and the temperature of the nodes of the two other edges are kept constant at 20 °C. In the same way for the tool, heat exchange is defined for the two edges in contact with the workpiece. The temperature of the nodes on the other two edges is set to room temperature.

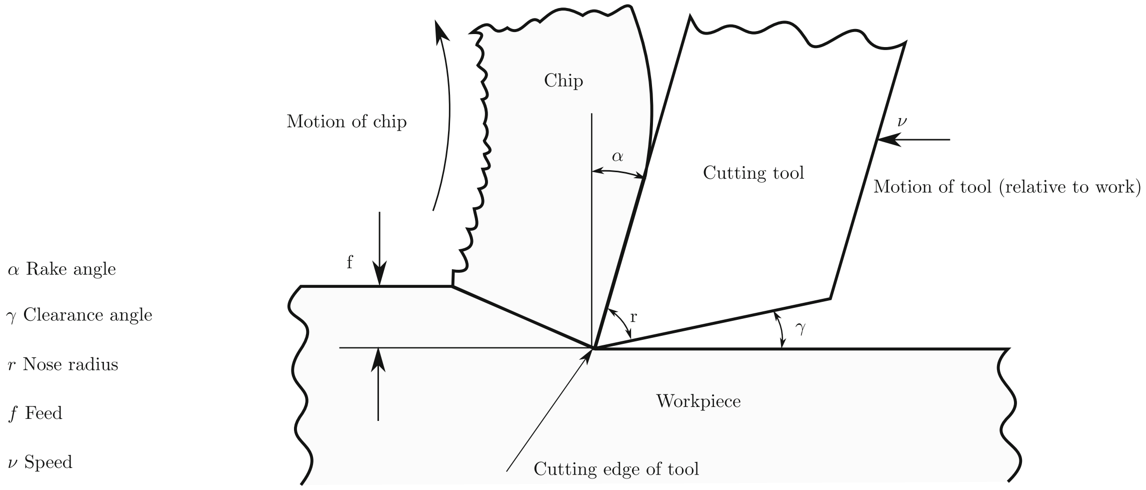


Fig. 4 Schematic diagram of turning operation

Table 1 Variables and objectives

Variables	Clearance angle γ (°)	Rake angle α (°)	Cutting edge radius r (mm)	Cutting speed v (mm/sec)	Feed rate f (mm/rev)
Objectives	Material removal rate (MRR) Maximize	Wear depth (ω) Minimize	Maximum interface temperature (T_{int}) Minimize		

3.2 Material properties

The turning process simulates the cutting of a plain carbon steel workpiece (AISI-1045) by an uncoated tungsten carbide (K10) cutting tool. The material properties for the workpiece and the tool are shown in Table 3. The Johnson-Cook (JC) constitutive model is used to simulate the workpiece material behavior within the range of strain rate, strain, and temperature during the process by:

$$\sigma = \left(A + B\epsilon^n\right)\left[1 + C\ln\left(\frac{\dot{\epsilon}}{\dot{\epsilon_0}}\right)\right]\left[1 - \left(\frac{T - T_0}{T_m - T_0}\right)^m\right] \tag{2}$$

were σ is the flow stress, ϵ is the true strain, $\dot{\epsilon}$ is the true strain rate, $\dot{\epsilon_0}$ is the reference true strain rate, T is the work piece temperature, T_0 is the ambient temperature, T_m is the workpiece material melting temperature, and A, B, C, n, and m are the model constants. The material model data for AISI-1045 are taken from [38], in which the calibration of Johnson-Cook constitutive model was carried out by using the SHPB high strain rate test. Table 4 shows the JC parameters for the workpiece incorporated in this study. The tool is assumed to be rigid.

3.3 Thermal properties

The thermal properties of AISI-1045 taken from [39] and K10 acquired from [40] are shown in Table 5. To reach a thermal steady-state condition, the simulation needs to be run much longer than specified 7-mm cut length used in this study. This will increase the computational time and storage capacity greatly, which makes the study impractical. Therefore, perfect contact condition in the interface of tool and workpiece is assumed in this study. This is accomplished by setting the heat transfer coefficient in DEFROM2D to a high value ($h_{\text{int}} = 100,000\ \text{KW/m}^2\ {}^\circ C$). An ideal contact condition has been utilized and experimentally verified in several studies [41–43]. Furthermore, Iqbal et al. [44] investigated the effect of interface heat transfer coefficient on high-speed machining. Figure 5 plots the variation of the interface temperature in terms of cutting time for four different DoEs. The plot shows that the interface temperature reaches a relatively thermal steady state at the end of the simulation.

3.4 Contact and friction model

The model incorporated for simulating the frictional condition at the tool-chip interface is vital for an accurate chip formation process. Malakizadi et al. [45] studied the influence of the friction model on FE simulation results. All friction models resulted to an identical range of minimum average error. In this study, two types of contact are utilized in the FE simulations: (i) sticking model and (ii) sliding model. The sticking friction model is defined by $\tau = mk$, where m is the shear friction coefficient (sticking coefficient) and k is the shear flow stress of the work material. The sliding friction model is defined by $\tau = \mu\sigma_n$ where μ is the Coulomb friction coefficient (sliding coefficient) and σ_n is the interface normal pressure.

The contact between the chip and workpiece is assumed to be governed by the sliding model with a constant sliding coefficient of $\mu = 0.8$. The tool and workpiece are assumed to undergo a hybrid contact consisting both friction models, namely sticking-sliding model defined by:

$$\tau = \begin{cases} \mu\sigma_n & \mu\sigma_n < mk \quad (L_{st} \le x \le L_{sl}) \\ mk & \mu\sigma_n \ge mk \quad (0 \le x \le L_{st}) \end{cases} \tag{3}$$

where L_{st} and L_{sl} are the sticking and sliding contact length, respectively. The constant sticking and sliding coefficients assigned for this study are $m = 1.0$ and $\mu = 0.6$, respectively.

3.5 Tool wear model

Usui's wear rate model [46] is adopted in this study. The model is defined by:

$$\dot{\omega} = D_1\sigma_n V_S e^{-(D_2/T)} \tag{4}$$

Table 2 Lower and upper bounds of the variables

Variable	γ (°)	α (°)	r (mm)	v (mm/sec)	f (mm/rev)
Range	2–15	0–15	0.01–0.1	1667–5000	0.05–0.4

Table 3 Material properties for the workpiece and the tool

Material properties	AISI-1045	K10 [37]
Density (g/cm^3)	7.85	14.95
Young's modulus (GPa)	f(Temp)	600
Poisson's ratio	0.3	0.25
Hardness (HRC)	7	81.4

where σ_n is the normal stress, T is the temperature, and V_s is the sliding velocity of the predicted nodal data of tool contact surface. The wear constants D_1 and D_2 are given in [47] for plain carbon steels and uncoated tungsten tools:

$$D_1 = 7.8 \times 10^{-9}, \quad D_2 = 5.302 \times 10^3. \tag{5}$$

Tool wear depth and maximum nodal temperature are extracted from simulations after more than half of the workpiece length is cut by the tool, i.e., 7 mm of cut length. The FE step length, number of steps, and total simulation time are defined accordingly for each simulation. The third objective, i.e, MRR, is calculated by using (1). The result extractor validates the result to conform with 7 mm traveled distance with the tolerance of $\epsilon = 1e^{-3}$. This is accomplished by result validation in the result extractor module (see Fig. 1), where the position of corner 2 of the workpiece (shown in Fig. 2) at the last step of the simulation is calculated and compared with the stopping criteria to check if the simulation has been completed successfully.

3.6 Mesh convergence

In FE modeling, the size of the mesh affects the accuracy considerably. A finer mesh generally results in a more accurate solution, at the same time increases the computational time. Therefore, a mesh convergence study determines a balanced mesh size and computational time. In this study, the effect of six different mesh sizes and consequently number of elements are investigated. They range from a coarse mesh and fast simulation to a very fine mesh with high number of elements and computationally expensive simulation. The mesh data including the minimum element size, number of elements, simulation time, and the extracted objectives are shown in Table 6. Mesh number 5 is chosen for this study, since a more refined mesh (mesh number 6) did not affect the objectives noticeably. The simulation time for this mesh size varies between 250 and 600 min depending on the set of variable combinations. Thus, by considering the capability of running 20 simulations at a same time, in the parallel workstations, and 40 DoEs at each generation (the population of the archive set in SPEA2 algorithm), each optimization generation takes around 15 to 20 h. The trade-off solutions both in objective and decision variable space seem to be converged after 17 generations. The results from the last generation of the optimization are extracted and presented in the next section.

It should be noted that obtaining the objective functions most accurately is not the purpose of the FE simulation in this study. As mentioned earlier, the FE simulation is adopted to demonstrate the framework and simulate the correct physical phenomenon of a basic turning operation.

4 Analysis of the framework and discussion

The potentials of the framework in investigating different aspects of a metal cutting process is discussed in this section. As mentioned before, the FE model used in this study only serves the purpose of demonstrating, presenting, and analyzing the developed framework and method for FE simulation-based MOO of a metal cutting operation. Hence, the following results and solutions should not be considered optimal before validation with physical experiments.

The final chip shape and temperature distribution on the tool and workpiece are shown in Fig. 6. The 40 trade-off solutions obtained using SPEA2 are shown in Fig. 7. The front reveals the conflicting nature of the MRR with other objectives. For instance, by accepting an increase in T_{int} and subsequently in ω, a more efficient process in terms of time can be achieved. Furthermore, the direct relationship of T_{int} and ω is realized from the trade-off solutions. This is the reason why the trade-off front is redundant, i.e., the front is a curve instead of a surface.

A helpful graphical tool for studying the relations between variables and objectives of the trade-off solutions is the correlation matrix plot in Fig. 8. Scatter plots of all pairs of variables-objectives appear in subplots along with the correlation coefficients related to each pair. The correlation coefficients measure the strength and direction of the linear relationship between each pair of variable

Table 4 The Johnson-Cook material parameters for AISI-1045 [38]

JC parameters	A (MPa)	B (MPa)	C	n	m	$\dot{\epsilon}_0$	T_0 (°C)	T_m (°C)
AISI-1045	553.1	600.8	0.0134	0.234	1	1	20	1500

Table 5 The thermal properties of the workpiece and the tool material

Thermal parameters	λ (W/m K)	$\rho \times C_p$ (J/cm^3 K)
AISI-1045 [39]	25 °C $< T <$ 600 °C: $3.91 \times 10^{-8}T^3 - 4.74 \times 10^{-5}T^2 - 0.0121T + 46.1$ $T >$ 600 °C: 26	25 °C $< T <$ 600 °C: $4.685 \times 10^{-6}T^2 + 1.527 \times 10^{-3}T + 3.664$ $T >$ 600 °C: 6.28
K10 [40]	80	5.7

and objective. The linear relation of MRR with speed and feed can be seen in the related subplots, confirmed by the high correlation coefficient of the two subplots. Speed and interface temperature are almost linearly correlated with a lower correlation coefficient, where the lower limit on the speed generates the trade-off solutions in an almost vertical line in the $\nu - T_{\text{int}}$ subplot. By observing the three subplots corresponding to γ, two clusters of solutions concentrated around clearance angles of 6 and 11° are noticeable. The subplots related to feed show that most of the trade-off solutions are spread out on the upper limit of the feed rate. Analyzing the results further by focusing on the correlation coefficient provides some useful information, about the trade-off solutions to this multi-objective optimization problem (MOOP). For instance, high correlation coefficient of cutting speed, 0.96, 0.94, and 0.88 with regard to ω, MRR, and T_{int} respectively, reveals the greater influence of ν on all objectives compared to other variables. Nevertheless, feed rate with correlation of 0.87, 0.80, and 0.71 with regard to T_{int}, MRR, and ω respectively has the second major effect on the objectives. The objectives have the least dependency to rake angle followed by the cutting edge radius. The clearance angle has a moderate effect on the objectives. The sign of the correlation coefficient determines the direction of the linear relationship. The positive correlation coefficients indicated a direct influence and a negative correlation shows the inverse influence. The clearance angle γ has an inverse effect on all objectives. In other words, by increasing γ, the values of all objectives will decrease.

Parallel coordinate plots are another useful tool for analyzing MOO results as shown in Figs. 9, 10, and 11. An application of coordinate plot is to study any trade-off solution across all objectives and variables. For example, solutions 13, 20, 23, and 39 are selected and highlighted in Fig. 9. Tooling engineers can choose any solution based on their interest on the value of any variable or objective and find the value of other corresponding variables and

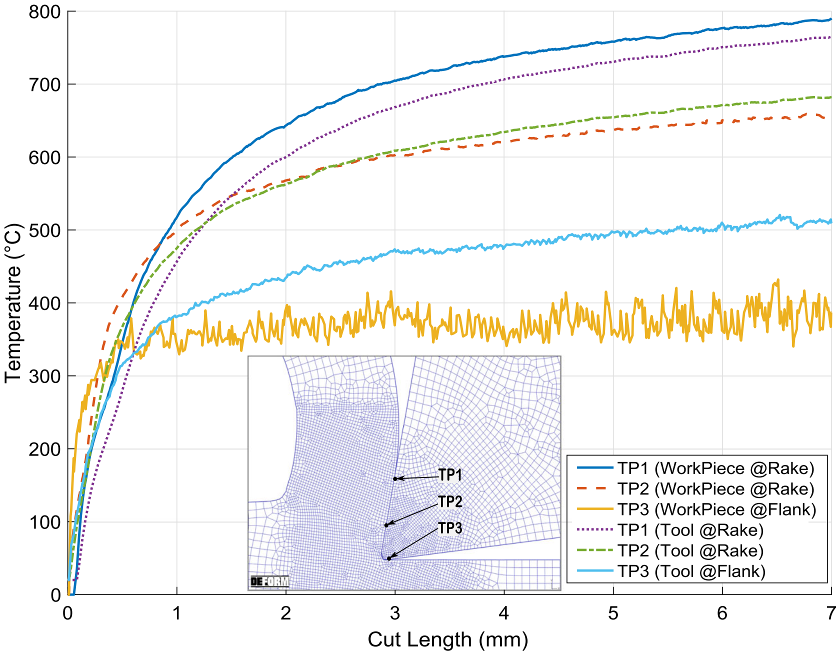


Fig. 5 Variation of tool and workpiece maximum temperature at the tracking points near the contact interface

Table 6 The mesh convergence study data

Mesh	Min. element size (mm)	No. of elements	Simulation time (min)	Interface temp. (°C)	Wear depth (mm)
Mesh1	0.04	1181	55	831.00	1.55E−07
Mesh2	0.025	1508	66	829.72	1.40E−07
Mesh3	0.01818	1917	165	819.10	1.32E−07
Mesh4	0.01414	2495	200	804.90	1.24E−07
Mesh5	0.01176	3203	335	795.31	1.18E−07
Mesh6	0.01	4042	560	795.51	1.17E−07

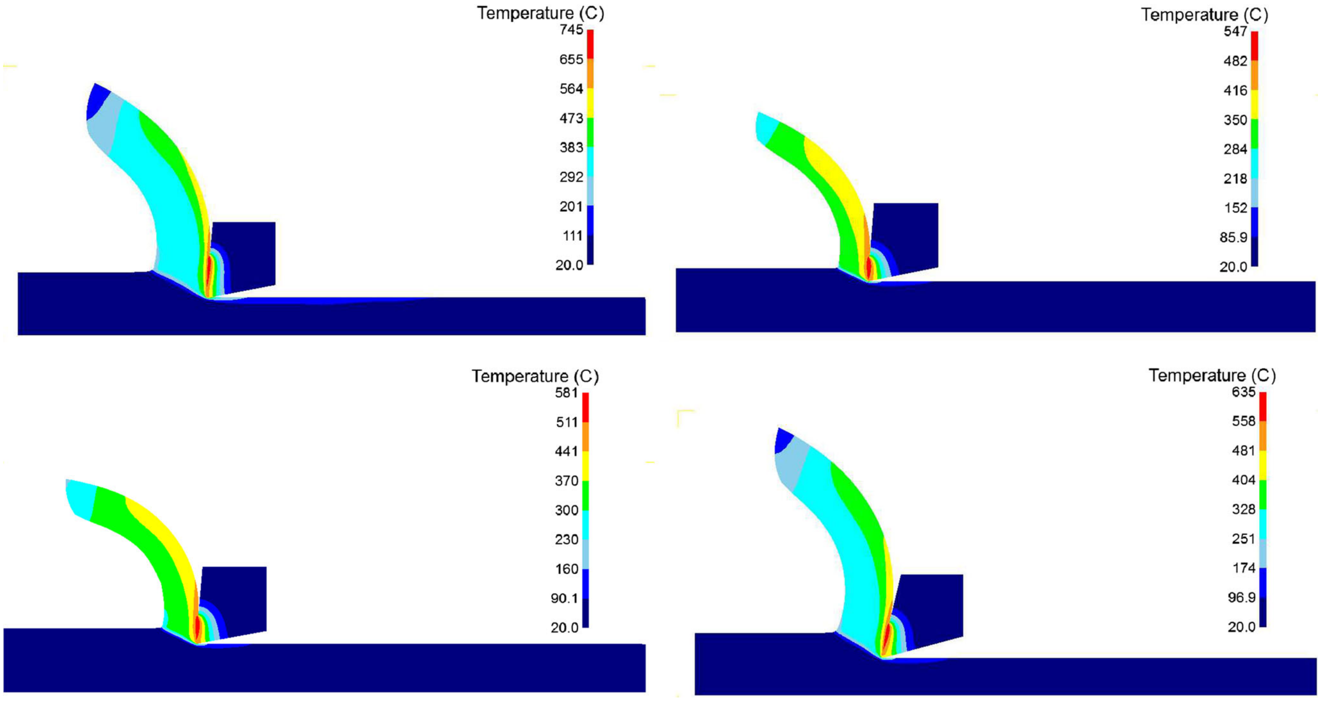
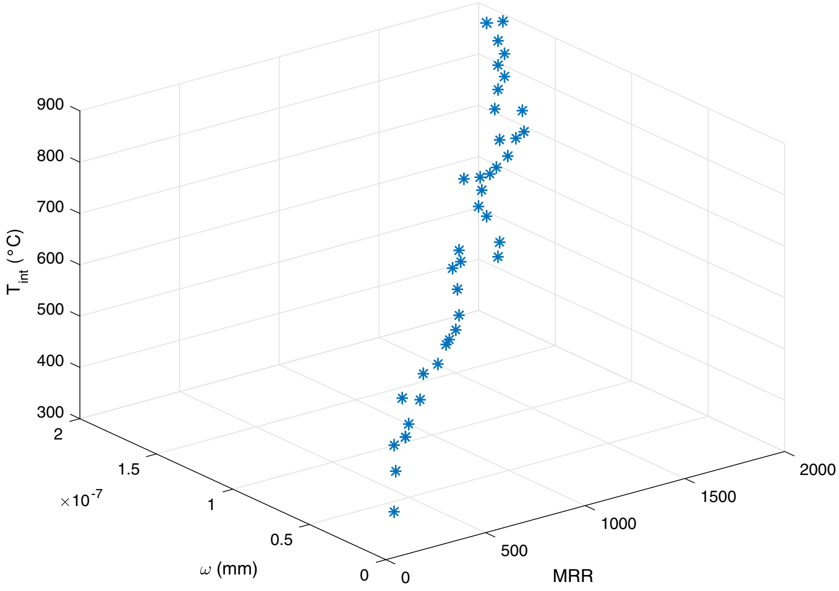


Fig. 6 Chip shape and temperature distribution of 4 different trade-off solutions

Fig. 7 Trade-off solutions in objective space



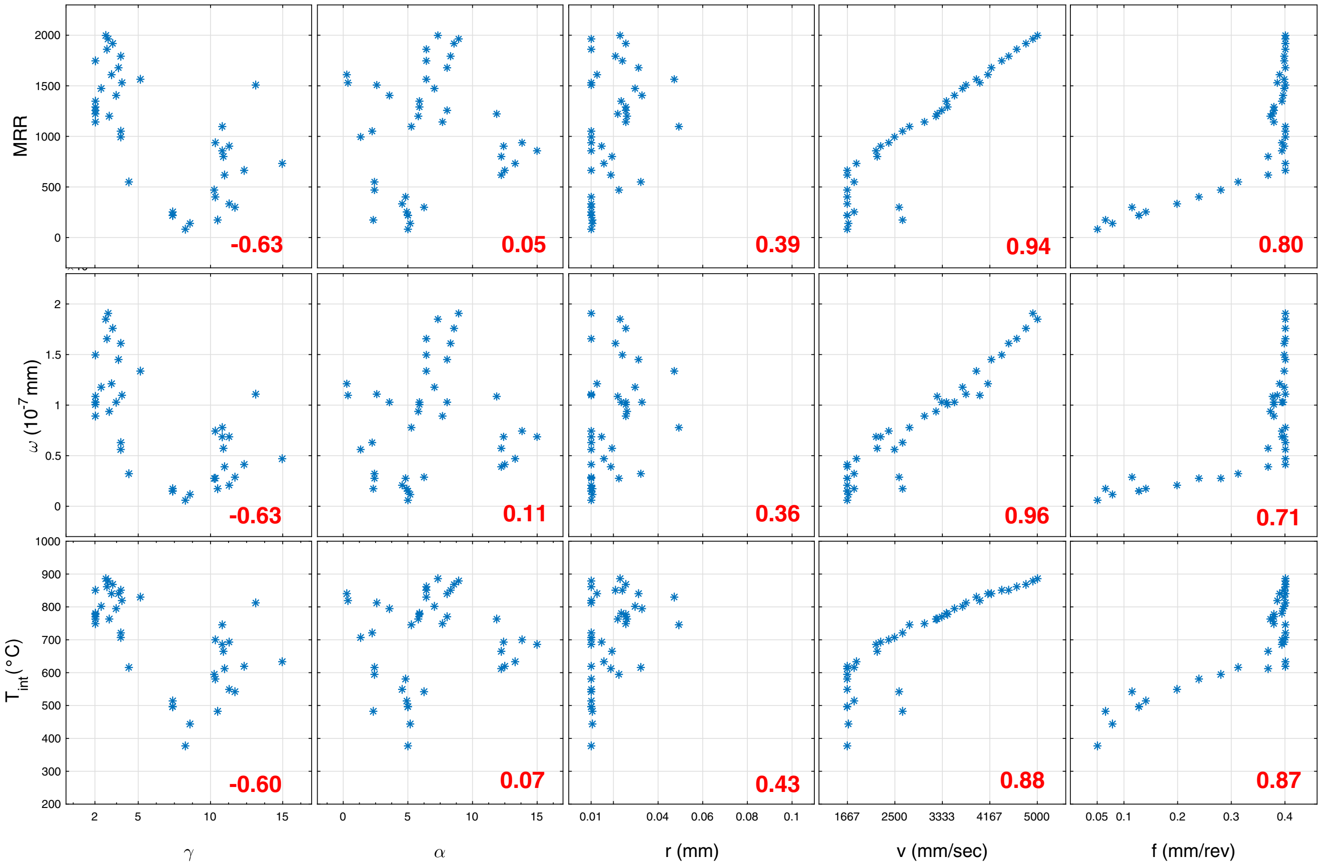


Fig. 8 Matrix plot of trade-off solutions between variables and objectives

objectives to that solution. For instance, solution 20 is generated with a high feed rate and cutting speed, while a tool with lowest possible cutting edge radius, a moderate rake angle, and low clearance angle is required. These parameters result in high T_{int}, ω, and MRR values. The parallel coordinate plot can assist a decision maker to limit the values of any parameter or combination of parameters to investigate the possible trade-off solutions within that range. Figure 10 shows the feasible trade-off solutions which deliver a tool-chip interface temperature of less than 600 °C. Shown in Fig. 10, 8 out of the total 40 trade-off solutions satisfy the limit on the interface temperature. Moderate clearance angle and relatively low rake angle, cutting edge radius, and cutting speed with a wide range of feed correspond to this setting.

Say that in a specific case, the available tools for tooling engineers are limited to a rake angle within $[5°, 7°]$ and cutting edge radius of [0.02, 0.04]. Figure 11 illustrates the feasible trade-off solutions for the tool limitation case, where only four trade-off solutions are feasible. The solutions yield to a relatively high MRR, moderate ω, and high T_{int}. Similarly, several other configuration of variables, objectives or combination of these two can be extracted by means of parallel coordinate plots.

4.1 Knowledge discovery

A cursory look at the matrix of scatter plots in Fig. 12 shows us that there is a lot more to be learned from the trade-off solution set as a whole. There are some visually apparent correlations between the objectives and the variables that show up in these plots. A data mining approach like FPM is capable of extracting rules with respect to the variables (i.e., decision space) that can then be mapped to the objective space. When extracted rules show up as well-defined clusters in the objective space, it indicates that these rules are specific to certain regions of the objective space and hence define its overall structure.

Table 7 shows all rules that can be found using FPM in a significant part (minSig $= 60\%$ in this case) of the trade-off solutions obtained above. The fourth column shows the percentage of other feasible solutions that also contain the corresponding rule. Rules are ordered such that the topmost rule best distinguishes the trade-off set from rest of the solutions. This distinguishing ability of the rules reduces as we move down the table. Rules with high significance in the trade-off set and low significance in the dominated solutions are characteristic features of the trade-off set and thus qualify to be knowledge that may be of interest to the user.

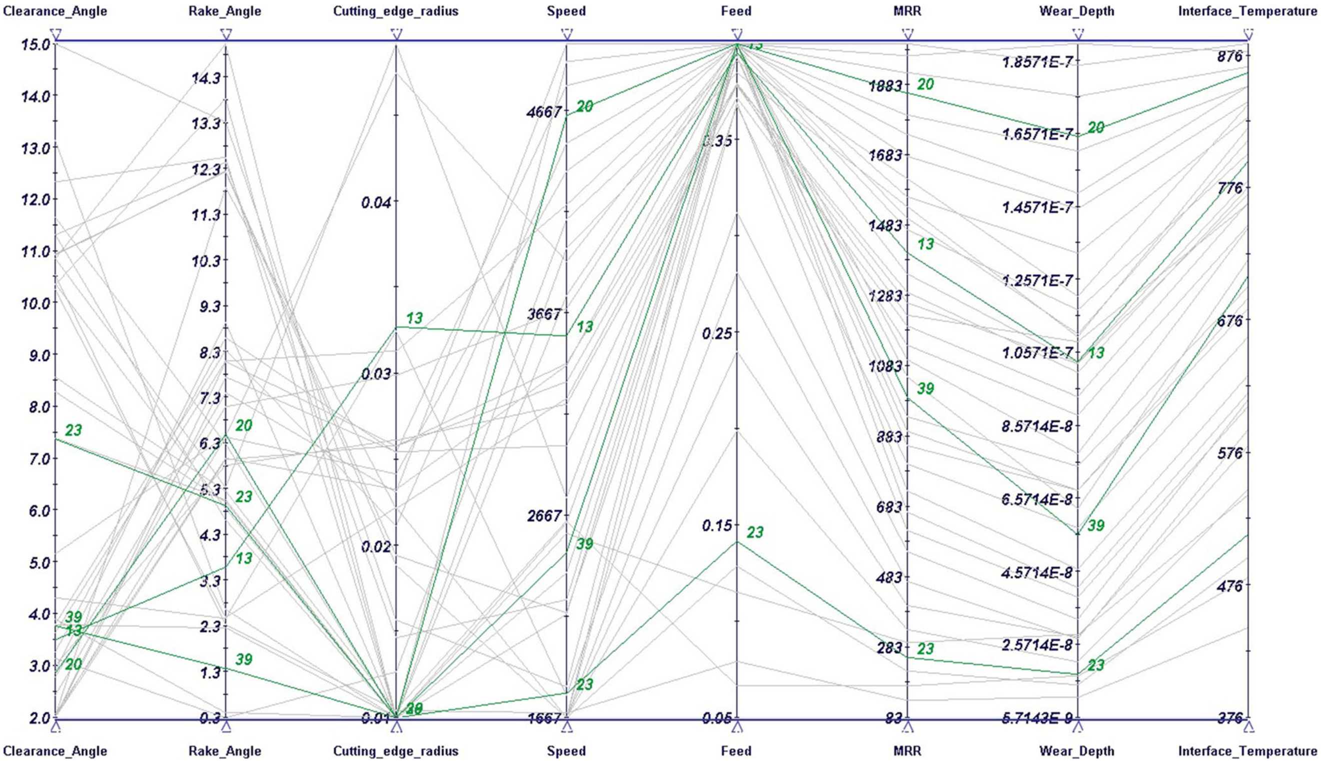


Fig. 9 Parallel coordinate plot, showing selected trade-off solutions

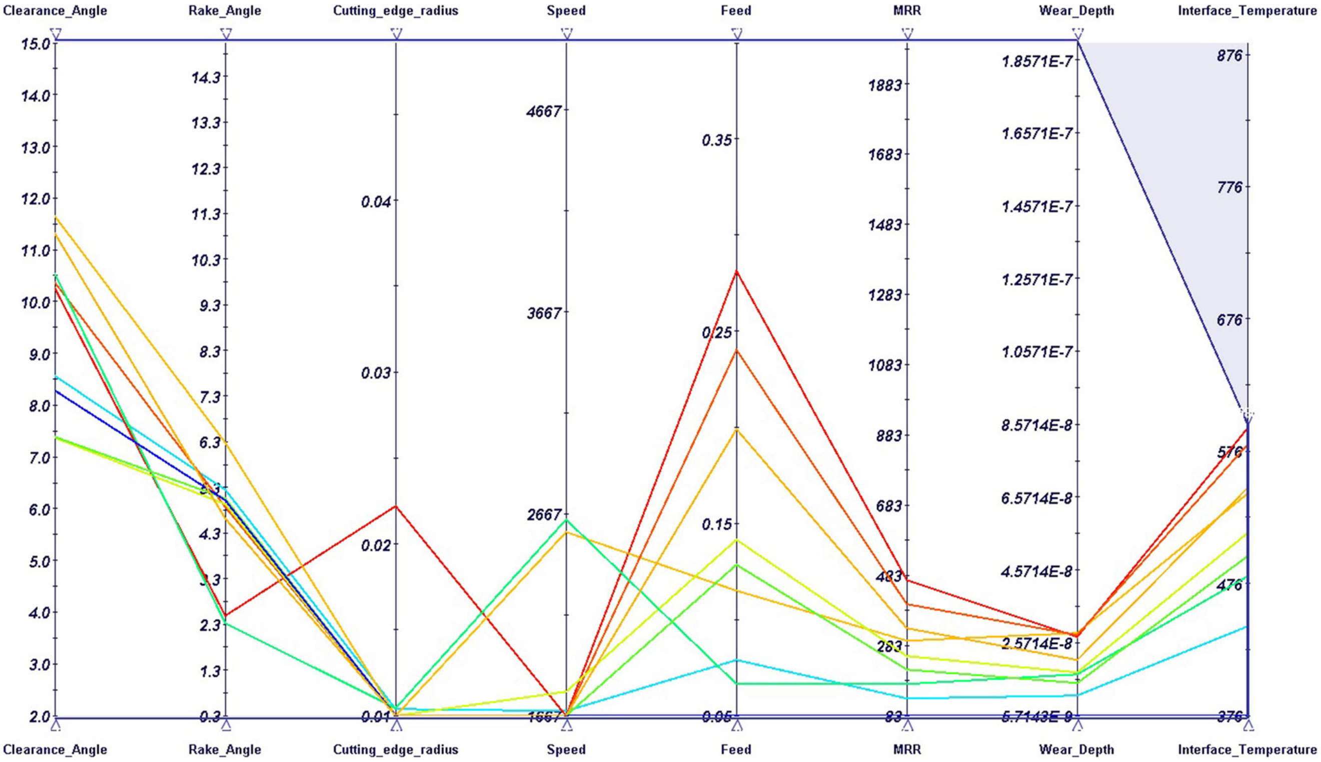


Fig. 10 Parallel coordinate plot, showing feasible trade-off solutions with a limit on interface temperature

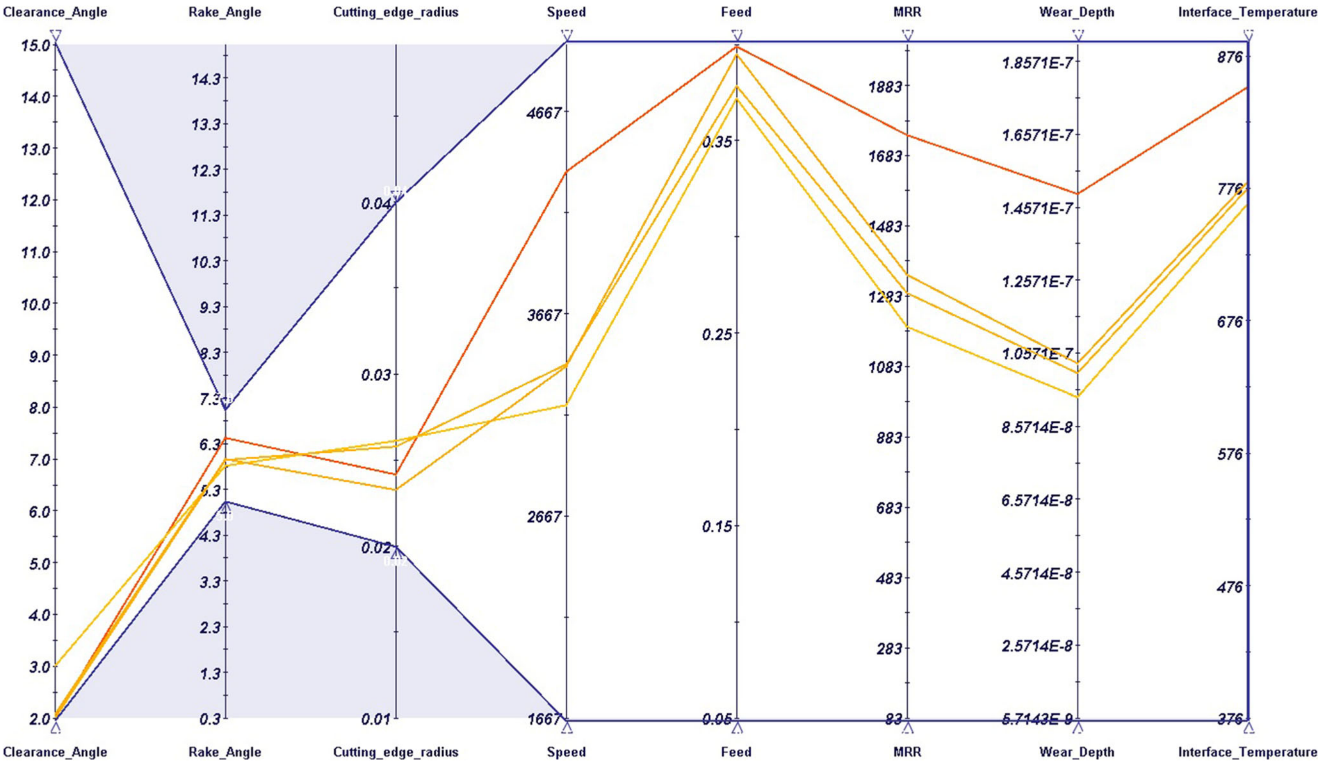


Fig. 11 Parallel coordinate plot, showing feasible trade-off solutions with limitations on geometry of the tool

Rule 1 indicates that most trade-off solutions have $f > 0.37894$. This can also be seen in the plots in the last column of Fig. 12. Since the bounds for the feed variable are $[0.05, 0.4]$, the rule represents a very narrow range of feed values close to its upper bound. This knowledge can be used as a thumb rule, for example, in future process optimizations. The second and the third rules indicate that a high proportion of trade-off solutions have $\gamma < 8.2892$ and $\alpha < 8.0606$. Such rules are not directly apparent from the first two columns of Fig. 12, but are important for refining the search space if more trade-off solutions are desired. Rule 4 can also be used as a thumb rule because it suggests a narrow range of values for the cutting edge radius, i.e., $r < 0.024177$ where the proportion of trade-off solutions is

Table 7 Rules with significance $\geq$ 60%

Rule No.	Rules (ordered by distinguishing ability)	Significance	
		Trade-off set	Elsewhere
1	Feed, $f > 0.37894$	61.76	23.03
2	Clearance angle, $\gamma < 8.2892$	60.16	40.13
3	Rake angle, $\alpha < 8.0606$	75.67	51.32
4	Cutting edge radius, $r < 0.024177$	66.84	50.33
5	Speed, $v > 2448.948$	69.52	58.22
6	Speed $v < 4284.1997$	89.04	82.89
7	Rake angle, $\alpha > 0.86467$	91.71	96.38
8	Cutting edge radius, $r > 0.018404$	62.03	66.78
9	Clearance angle, $\gamma > 2.0477$	88.77	97.37
10	Feed, $f < 0.39993$	77.54	94.74

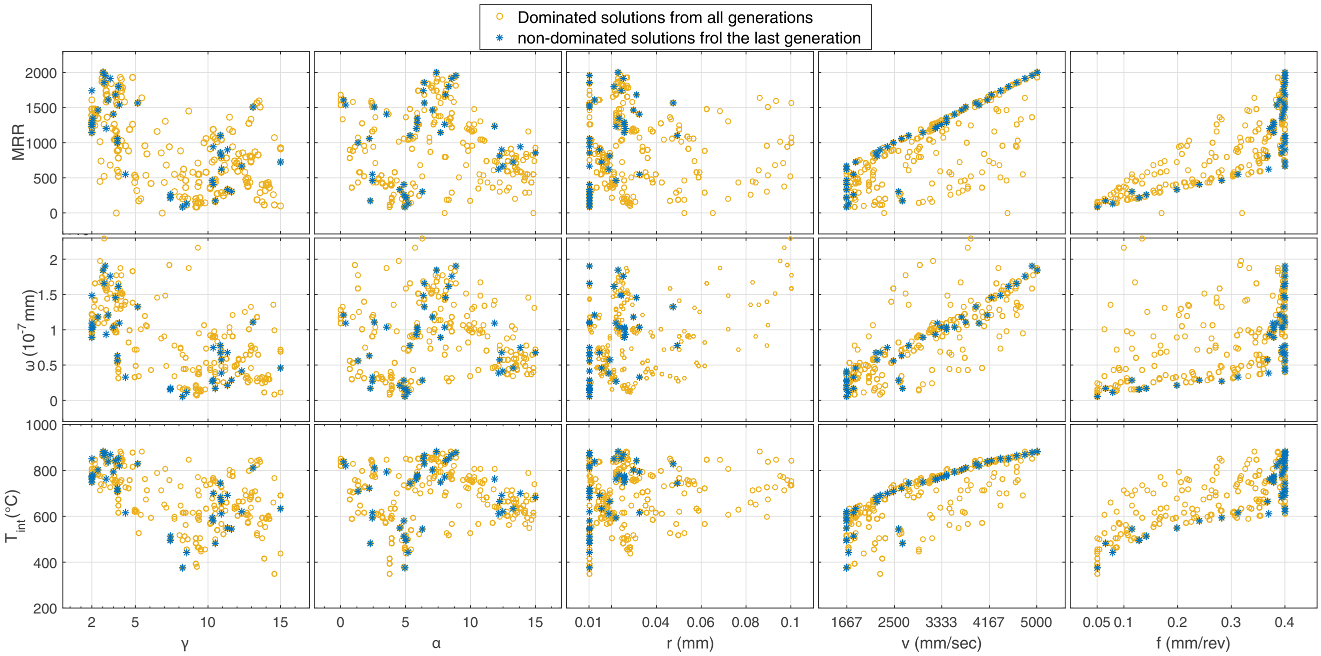


Fig. 12 Matrix plot of solutions in all 17 generations between variables and objectives

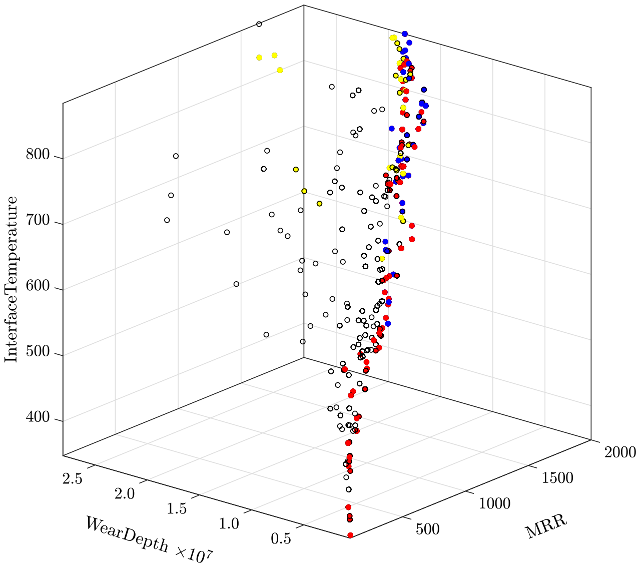


Fig. 13 Visualization of the rule $f > 0.37894$ in the objective space

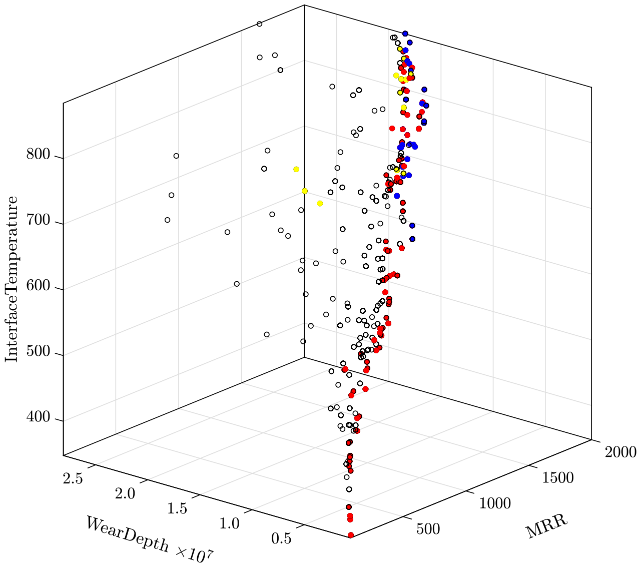


Fig. 14 Visualization of the compound rule $f > 0.37894 \wedge \gamma < 8.2892 \wedge r < 0.024177$ in the objective space

higher than that of dominated solutions. The third column of plots in Fig. 12 reveals that most trade-off solutions take r values close to its lower bound. Like the first, this rule can also be used as a thumb rule. Rules 5 and 6 together define a narrower range of values for the speed v than in the original problem, thus reducing the search space.

Note that as we go down the table of rules, the proportion of dominated solutions that follow the rule increases. From rule 7 onwards, this proportion is greater than that for trade-off solutions. Therefore, these rules should not be used for restricting the search space. However, they may still constitute as knowledge.

Each rule can be visualized in the objective space to reveal hidden structure. For example, Fig. 13 shows clusters of solutions for the most distinguishing rule, $f > 0.37894$. These are (i) cluster of trade-off solutions that contain this rule (blue, 61.76%), (ii) cluster of trade-off solutions that do not contain this rule (red, 38.24%), and (iii) cluster of other feasible solutions that also contain this rule (yellow, 23.03%).

It is interesting to see that this rule applies to solutions with high MRR and interface temperature.

FPM also allows us to combine different rules to form compound rules. In this case, the significance values have to be recalculated. For example, the first three rules in Table 7 can be combined to form $f > 0.37894 \wedge \gamma < 8.2892 \wedge r < 0.024177$. The rule is very specific and therefore, we expect it to be applicable to fewer solutions. Figure 14 reveals the clusters that this rule generates. Note that there is a clear transition in the objective space (from blue to red), where this compound rules ceases to apply. Knowledge like this can support the decision maker to understand how changes in the optimization problem formulation can affect Pareto-optimality of the solutions.

5 Concluding remarks

A framework for MOO of machining processes and knowledge discovery by utilizing FE simulations was developed. The framework was presented and applied to turning of AISI-1045 with an uncoated K10 WC tool. Two objectives, tool-chip interface temperature and tool wear depth, were extracted from FE simulations and one objective, material removal rate, was computed from a mathematical equation. Tool geometry parameters and turning process parameters were optimized. The trade-off solutions were presented by employing different graphical tools. Knowledge in the form of rules was extracted by means of flexible pattern mining. The results of MOO correspond with the available findings in literature. However, it is worth noting that most of the available literature does not treat the turning problem as multi-objective in the same way as described in this paper. Specifically, this paper obtains the parameters that optimize three objectives simultaneously and reveals the hidden structure between the objective space and the decision space. Overall, the framework proved to be effective in achieving the MOO of machining processes based on actual FE simulations. The framework can be employed in optimizing other machining processes as well as expanding the simulations to industrial 3D applications in future work.

Acknowledgements This work was partially financed by the Knowledge Foundation (KKS), Sweden, through the Synergy project, Knowledge-Driven Decision Support via Optimization (KDDS). The authors gratefully acknowledge their provision of research funding, and thank the industrial partners, Volvo Car Corporation and AB Volvo, for their collaborative supports during the project. We also we would like to thank Dr. Mirza Cenanovic for developing the initial architecture of the automation system.

Appendix: Implementation of the framework and description of the appended files

The text-based modules of DEFORM which can be used to set up and run simulations in automatic mode without going through the graphic user interface (GUI) is used. The text-based pre-processor $DEF\_PRE.EXE$ is employed to generate the database files.

The jobs (database files or FE simulation files) are submitted by calling the simulation control script

$DEF\_ARM\_CTL.COM$.

There are two different types of keywords that can be read by the pre-processor: input keywords and action keywords. Input keywords contain data that is directly used as data for a simulation. This can be a geometry definition, convection coefficient values, or other such data. Action keywords perform certain operations when the pre-processor is reading the data.

The pre-processor can be controlled by redirecting a text input control file (.inp). The .inp file contains the user inputs if the text-based system were run in interactive mode. The .key files contains a series of *Action Keywords* which trigger the pre-processor to perform a series of options.

First, the turning problem is modeled in GUI of DEFORM software with required material properties,

Table 8 Description of appended files, which is used to generate the FE file

File name	Input(s)	Output(s)	Decription
DoEMaker.m	–	.DB file	Receives the required value of the input variables, and outputs the database (.DB) file which is run by DEFORM to simulate the cutting process.
createStepKeyfile.m	Cutting length, Speed, and min element size	Step.key	Calculates the FE stopping controls such as number of FE steps, maximum FE running time, the maximum allowable time step size based on the cutting distance and minimum FE element size.
createFreeCADMacro.m	Tool's geometrical variables	Tool.FCMacro	Receives the tool geometrical parameter values and modifies a template FreeCAD script file to create an updated script with the new parameters.
The DoEMaker runs the script to create the workpiece and tool geometries in the form of a IGS file (Tool.IGS).			
The IGS file is run with Deform IGS reader (DEF_IGS.exe) to create an initial GEO file: DEF_IGS.OUT			
createGeofile.m	–	Tool.GEO	Modifies the GEO file (DEG_IGS.OUT) to a Deform compatible GEO format: Tool.GEO
createSpeedKeyfile.m	X and Y direction of speed, number of nodes in the workpiece that speed should be assigned to	Speed.KEY	Implements the speed in the workpiece coordinates and creates the KEY file
The DEFORM mesh generator is run on the tool.GEO file to create a meshed KEY file of the geometry: Tool.Key			
ExtractMeshData.m	–	BCCTMP: nodes with heat transfer ECCedge: elements with heat transfer	Receives the meshed geometry file from DEFORM mesh generator (Tool.Key), triangulates the mesh in MATLAB, and identifies the elements and nodes of the geometry which boundary conditions should be imposed on.
createBoundaryfiles.m	BCCTMP, BCCcode, ECCedge, ECCcode	BCCTMP.Key, ECCTMP.Key	Creates key files that sets the constraints codes to the related nodes and elements. DEFORM identifies the boundary conditions as codes in front of node or element numbers.
createContactfile.m	Object numbers (tool and workpiece)	Contact.Key	Creates a key file that sets the related self-contact and tool-workpiece contact codes.
createToolMaterialfile.m	THRCND: Thermal conductivity HEATCP: Heat capacity	MTLGRP.Key	Creates a key file that can modify the tool thermal conductivity and capacity
The default, template and all created Key files are assembled by using a .inp file in deform text-based pre-processor (DEF_PRE.EXE) and the final AssembledKey.KEY is created.			
The file name is changed to the related gene number of the population and the text-based pre-processor generates the database file (.DB) that is ready to be run by DEFORM.			
The DB files are distributed to different, parallel work stations (PW) and the simulation is run on the PWs.			
createObjandKeyfile.m	–	objectives_geneNr.mat	This file opens the text-based preprocessor, loads the last step plus the 10 last steps of the database and save the data as a keyword file. Then, it calculates the objectives and saves it as a .mat file.
ExtractData.m	–	objectives, StepN	Finds the objectives corresponding to the last step and the total number of steps.

thermal and friction properties, boundary conditions, FE controls, and geometries. Then, a template KEY file is exported with all the properties (def.KEY). The nodes of the workpiece, which the speed should be imposed on, are also extracted as a separate key file (defSpeed.Key). The procedure and description of different files used to generate the FE file, a file that is ready to be run by DEFORM, is shown in Table 8.

In a setting file (setting.ini), all the initial settings of the automation system including the number of jobs in each generation of optimization, available work stations and their names, number of cores, and clock speed are set.

In the cloud project folder, two folders are created: CurrentRun and NextRun. Each set of variables (from the DoE) generated by optimization algorithm is saved in NextRun folder as a text file named by *gene*. The number of *gene* files is equal to the number of DoE in each generation of the optimization study.

The automation system, denoted by UAS2 (Ultimate Automation System 2), is a cloud-based Windows script that controls and distributes the modules to computational devices. The overall algorithm of the USA2 is as follows:

MainPC:

```
1.  Wait while stop.txt exists
2.  Clear all files and folders in \CurrentRun\PCname\
    (for all PCnames)
3.  Assign jobs to PC folders
4.  Create go file, to signal the other PCs to run
5.  Clear all Temp folders on local PC
6.  RunWait AutoRun.exe on local PC
7.  Run W4ALJ.exe (Wait for all local jobs) on local PC
8.  Check if all PCs have started correctly
9.  RunWait WaitForAllPCs.exe (Wait for all PCs)
10. Copy results to result folders
11. Delete genes in NextRun
12. Start MATLAB script
13. Wait for new genes in NextRun
14. Send Email about generation completion
15. Delete genes in CurrentRun
16. Copy genes from NextRun to CurrentRun
17. Delete genes in NextRun
18. Increment generation and goto 1
```

Other PCs:

```
1. Clear all Temp folders.
2. Wait for stop.txt
3. Wait for go.txt in CurrentRun\PCname
4. Delete go.txt
5. RunWait AutoRun
6. Run W4ALJ (Wait for all local jobs)
7. Go to 1
```

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Wei L, Yuying Y (2008) Multi-objective optimization of sheet metal forming process using Pareto-based genetic algorithm. Journal of Materials Processing Technology 208(1):499–506. ISSN 0924-0136. https://doi.org/10.1016/j.jmatprotec.2008.01.014
2. Nor NHM, Muhamad N, Ibrahim MHI, Ruzi M, Jamaludin KR (2011) Optimization of injection molding parameter of Ti-6Al-4V powder mix with palm stearin and polyethylene for the highest green strength by using Taguchi method. International Journal of Mechanical and Materials Engineering 6(1):126–132. ISSN 1823-0334
3. Tutum CC, Hattel JH (2010) Optimisation of process parameters in friction stir welding based on residual stress analysis: a feasibility study. Sci Technol Weld Join 15(5):369–377. https://doi.org/10.1179/136217110X12707333260455
4. Olofsson J, Salomonsson K, Johansson J, Amouzgar K (2017) A methodology for microstructure-based structural optimization of cast and injection moulded parts using knowledge-based design automation. Adv Eng Softw 109(Supplement C):44–52. ISSN 0965-9978
5. Nagendra Parashar BS, Mittal RK (2002) Elements of manufacturing processes PHI Learning Pvt Ltd.
6. Mukherjee I, Ray PK (2006) A review of optimization techniques in metal cutting processes. Comput Ind Eng 50(1):15–34
7. Aggarwal A, Singh H (2005) Optimization of machining techniques—a retrospective and literature review. Sadhana 30(6):699–711
8. Tutum CC, Hattel J (2011) State-of-the-art multi-objective optimisation of manufacturing processes based on thermo-mechanical simulations. Springer, London, pp 71–133. ISBN 978-0-85729-652-8
9. Deb K (2001) Multi-objective optimization using evolutionary algorithms, vol 16. Wiley
10. Cus F, Balic J (2003) Optimization of cutting process by GA approach. Robot Comput Integr Manuf 19(1):113–121
11. Lee BY, Tarng YS (2000) Cutting-parameter selection for maximizing production rate or minimizing production cost in multistage turning operations. J Mater Process Technol 105(1):61–66
12. Sardiñas QR, Marcelino Rivas Santana, Brindis EA (2006) Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. Eng Appl Artif Intell 19(2):127–133
13. Yusup N, Zain AM, Hashim SZM (2012) Evolutionary techniques in optimizing machining parameters: review and recent applications (2007–2011). Expert Syst Appl 39(10):9909–9927
14. Sultana I, Dhar NR (2010) GA based multi objective optimization of the predicted models of cutting temperature, chip reduction co-efficient and surface roughness in turning AISI 4320 steel by uncoated carbide insert under HPC condition. pages 161–167, Sanya, 2010. ISBN 9780791859544. cited By 2; Conference of International Conference on Mechanical, Industrial, and Manufacturing Technologies, MIMT 2010 ; Conference Date: 22 January 2010 Through 24 January 2010; Conference Code: 80589
15. Umer U, Qudeiri JA, Hussein HAM, Khan AA, Al-Ahmari AR (2014) Multi-objective optimization of oblique turning operations using finite element model and genetic algorithm. Int J Adv Manuf Technol 71(1-4):593–603. https://doi.org/10.1007/s00170-013-5503-y. cited By 4

16. Thepsonthi T, Özel T (2014) An integrated toolpath and process parameter optimization for high-performance micro-milling process of Ti-6Al-4V titanium alloy. Int J Adv Manuf Technol 75(1-4):57–75. https://doi.org/10.1007/s00170-014-6102-2. cited By 3
17. Deb K, Srinivasan A (2006) Innovization innovating design principles through optimization. In: Proceedings of the 8th annual conference on genetic and evolutionary computation, GECCO 2006, ACM, pp 1629–1636
18. Deb K, Sindhya K (2008) Deciphering innovative principles for optimal electric brushless D.C. permanent magnet motor design. In: Proceedings of the World congress on computational intelligence (WCCI-2008). IEEE Press, Piscatway, pp 2283–2290
19. Ng AHC, Dudas C, Nießen J, Deb K (2011) Simulation-based innovization using data mining for production systems analysis. In: Multi-objective evolutionary optimisation for product design and manufacturing, Springer, pp 401–429
20. Dudas C, Frantzén M, Ng AHC (2011) A synergy of multi-objective optimization and data mining for the analysis of a flexible flow shop. Robot Comput Integr Manuf 27(4):687–695
21. Sugimura K, Obayashi S, Jeong S (2010) Multi-objective optimization and design rule mining for an aerodynamically efficient and stable centrifugal impeller with a vaned diffuser. Eng Optim 42(3):271–293
22. Greco S, Matarazzo B, Slowinski R (2008) Dominance-based rough set approach to interactive multiobjective optimization. In: Multiobjective optimization, Springer, pp 121–155
23. Sugimura K, Obayashi S, Jeong S (2007) Multi-objective design exploration of a centrifugal impeller accompanied with a vaned diffuser. In: Proceedings of the 5th Joint ASME/JSME fluid engineering conference, ASME, pp 939–946
24. Sugimura K, Jeong S, Obayashi S, Kimura T (2009) Kriging-model-based multi-objective robust optimization and trade-off-rule mining using association rule with aspiration vector. In: 2009 IEEE congress on evolutionary computation, CEC, IEEE, pp 522–529
25. Chiba K, Obayashi S, Nakahashi K, Morino H (2005) High-fidelity multidisciplinary design optimization of aerostructural wing shape for regional jet. In: Proceedings of the 23rd AIAA applied aerodynamics conference, AIAA, pp 621–635
26. Jeong S, Chiba K, Obayashi S (2005) Data mining for aerodynamic design space. J Aerosp Comput Inf Commun 2(11):452–469
27. Obayashi S, Jeong S, Chiba K (2005) Multi-objective design exploration for aerodynamic configurations. In: Proceedings of the 35th AIAA fluids dynamics conference and exhibit, AIAA, pp 2005–4666
28. Bandaru S, Deb K (2013) A dimensionally-aware genetic programming architecture for automated innovization. In: Proceedings of the 7th international conference on evolutionary multi-criterion optimization, EMO 2013, Springer, pp 513–527
29. Deb K, Bandaru S, Greiner D, Gaspar-Cunha A, Tutum CC (2014) An integrated approach to automated innovization for discovering useful design principles: case studies from engineering. Appl Soft Comput 15:42–56
30. Bandaru S, Ng AHC, Deb K (2017) Data mining methods for knowledge discovery in multi-objective optimization: part a—survey. Expert Systems with Applications 70:139–159
31. Deb K, Kumar A (1995) Real-coded genetic algorhithms with simulated binary crossover: studies on multimodel and multiobjective problems. Comput Syst 9(6):431–454
32. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-report, 103
33. Amouzgar K, Rashid A, Strömberg N Multi-objective optimization of a disc brake system by using SPEA2 and RBFN. In: Proceedings of the ASME 2013 international design engineering technical conferences, vol 3 B, Portland, August 4-7 2013. American Society of Mechanical Engineers. https://doi.org/10.1115/DETC2013-12809
34. Amouzgar K, Cenanovic M, Salomonsson K (2015) Multi-objective optimization of material model parameters of an adhesive layer by using SPEA2. In: 11Th world congress of structural and multidisciplinary optimization (WCSMO-11), pp 249–254
35. Bandaru S, Ng AHC, Deb K (2017) Data mining methods for knowledge discovery in multi-objective optimization: part B—new developments and applications. Expert Syst Appl 70:119–138
36. Agrawal R, Srikant R (1995) Mining sequential patterns. In: Proceedings of the 11th international conference on data engineering, IEEE, pp 3–14
37. López de Lacalle LN, Lamikiz A, Fernández de Larrinoa J, Azkona I (2011) Advanced cutting tools. In: Machining of hard materials, Springer, pp 33–86
38. Jaspers S (1999) Metal cutting mechanics and material behaviour Technische Universiteit Eindhoven Eindhoven
39. Halim T, Maria S (2008) Finite element modeling of the orthogonal metal cutting process: modeling the effects of coefficient of friction and tool holding structure on cutting forces and chip thickness PhD thesis
40. Klocke F (2011) Manufacturing processing i: cutting. RWTH ed., Springerlink
41. Arrazola PJ, Villar A, Ugarte D, Marya S (2007) Serrated chip prediction in finite element modeling of the chip formation process. Mach Sci Technol 11(3):367–390
42. Bordin A, Imbrogno S, Rotella G, Bruschi S, Ghiotti A, Umbrello D (2015) Finite element simulation of semi-finishing turning of electron beam melted Ti6Al4V under dry and cryogenic cooling. Procedia CIRP 31:551–556
43. Thepsonthi T, Özel T (2015) 3-d finite element process simulation of micro-end milling Ti-6Al-4V titanium alloy: experimental validations on chip flow and tool wear. J Mater Process Technol 221:128–145
44. Iqbal SA, Mativenga PT, Sheikh MA (2008) An Investigative study of the interface heat transfer coefficient for finite element modelling of high-speed machining. Proc Inst Mech Eng B J Eng Manuf 222(11):1405–1416
45. Malakizadi A, Hosseinkhani K, Mariano E, Ng E, Del Prete A, Nyborg L (2017) Influence of friction models on fe simulation results of orthogonal cutting process. Int J Adv Manuf Technol 88(9):3217–3232. https://doi.org/10.1007/s00170-016-9023-4
46. Usui E, Shirakashi T, Kitagawa T (1984) Analytical prediction of cutting tool wear. Wear 100(1):129–151. ISSN 0043-1648. https://doi.org/10.1016/0043-1648(84)90010-3
47. Kitagawa T, Maekawa K, Shirakashi T, Usui E (1989) Analytical prediction of flank wear of carbide tools in turning plain carbon steels. Part 2. Prediction of flank wear. Bull Jpn Soc Precis Eng 23(2):126–134