

# MODELLING OF A TURNING PROCESS USING THE GRAVITATIONAL SEARCH ALGORITHM

Hrelja, M.; Klancnik, S.; Balic, J. & Brezocnik, M.

University of Maribor, Faculty of Mechanical Engineering, Smetanova 17, 2000 Maribor, Slovenia

E-Mail: marko.hrelja@um.si, simon.klancnik@um.si, joze.balic@um.si, miran.brezocnik@um.si

## Abstract

This paper proposes the modelling of a turning process using a gravitational search algorithm (GSA). GSA is an optimization algorithm based on Newton's law of universal gravitation and mass interactions. In order to sufficiently describe the turning process, at least three independent variables are required: cutting speed, feed-rate, and cutting depth. Independent variables have impacts on dependent variables, which were in our case cutting force, surface roughness, and tool-life. The values of independent and dependent variables obtained by measurements serve as a knowledge database for feeding the GSA optimization process. During our research the GSA was used for optimizing the numerical coefficients of predefined polynomial models for describing the observed output variables. The accuracies of the obtained prediction models were proved by means of a testing data set that was excluded from the training data. The research showed that the obtained results were comparable with the other optimization algorithms such as particle swarm optimization (PSO). However, the optimization time required for GSA optimization was, in certain cases, significantly shorter.

(Received in December 2012, accepted in August 2013. This paper was with the authors 2 months for 2 revisions.)

**Key Words:** Machining, CNC Turning, Artificial Intelligence, Modelling, Gravitational Search Algorithm

## 1. INTRODUCTION

Following on from the expansion and improvement of modern machine tools newer and more complex products are being created. However, machining processes such as milling and turning are still of the utmost importance. Usually, the most important factor that defines the quality of a product is its surface roughness, which leads to assembling where certain tolerances are needed or, if the product is simple enough, a certain amount of roughness being allowed in order to spare the cutting tools and therefore extending a tool's life that is also affected by the cutting force.

The gravitational search algorithm is architecturally similar to the mainstream particle swarm optimization algorithm with the main differences being the optimization core where GSA uses Newtonian laws. The PSO method has been extensively researched by several researchers previously [1-3], therefore PSO serves as a good comparison regarding GSA research when determining its accuracy and run parameters, including optimization time. Genetic programming and genetic algorithm approaches are also popular for optimization methods [4, 5]. In order to obtain successful optimization of machining process it is of the essence that the algorithm is designed for multi-objective optimization, as there is more than one criterion that defines the machining [6]. According to multi-objective optimization, usage of the knowledge-based optimization procedure is essential as the algorithm predicts coefficients of the optimization polynomial regarding the selected machining model [7].

Many other different applications of knowledge-based intelligent systems based on the deterministic and/or non-deterministic modelling have been developed within mechanical engineering field (see for example [8-15]). Although PSO is commonly used, optimization can also be successfully done by applying neural networks [16-19]. In addition to neural

networks other approaches are also available such as the Taguchi method [20], genetic algorithms [21], and others.

Measurements have to be successfully carried out in order to acquire necessary data. In order to achieve successful turning machining modelling, tool-life has to be determined using tool condition monitoring, where the cutting force is also important and which directly causes surface roughness [22]. As modern materials are substantially tougher, it is of the essence that prediction models are designed for alloy-based materials [23]. As research usually turns towards the optimization of certain machining parameters such as lowering the cutting forces [24], minimizing power consumption [25], or maximizing tool-life [26, 27], it is also important to combine certain parameters in order to successfully predict production capabilities [28] in order to successfully increase the production rate.

As PSO, genetic programming, and neural networks are commonly known in mechanical engineering research [29], this paper proposes a relatively new approach called gravitational search algorithm. As basic GSA optimization works solely for determining the minimums and maximums of the mathematical functions [30, 31], for the purpose of this research it has had to be extensively modified. Such or similar modifications have already been tried out within other different research fields such as power consumption and electricity distribution [32].

In order to predict an optimal polynomial that would serve us for determining dependent variables (cutting force, surface roughness, and tool-life), three independent variables would have to be used, which would be cutting speed, feed-rate and cutting depth. According to the measured data, GSA would recalculate values to obtain model coefficients. As basic GSA optimization only calculates the minimums and maximums of mathematical functions, an additional optimization module would have to be designed that would simulate the multiple regression analysis of the imported training data from a knowledge-based table, where the optimization results would be polynomial coefficients.

The proposed system could serve as a tool for experienced machinists by simplifying the obtaining of machining parameters, whilst on the other hand it could also serve inexperienced workers as a proofing tool for verifying their predictions.

## **2. EXPERIMENTAL RESULTS**

Experimental data used in this research were based on the experimental work of Jurkovic as presented in [4, 33]. As these results have been used for various different optimization algorithms, they were also seen as excellent comparisons for different optimization approaches. Two sets of training data values were obtained, one for rough and one for finish machining.

The experiment was conducted using a tool holder and insert manufactured by the company Sandvik Coromant:

- Tool holder DDJNL 0-3225P15,
- Insert Sandvik Coromant DNMG 150608-PM4025.

The tool holder used for the experiment enabled exchangeable inserts of negative geometry, whilst the inserts were TiN covered.

Materials used for the experiment:

- Carbonized steel C45E (EN 10083/1996),
- A 6 m long hot-rolled cylinder with a diameter of  $\varnothing$  100 mm and a mass of 61.7 kg/m,
- The machining blanks were cut into cylinders with dimensions of  $\varnothing$  100 mm  $\times$  380 mm.

The cutting speed is a tangential component of the spindle speed and defines the values of the rotating spindle. The feed-rate defines the relative movement of the cutting tool according to the work-piece along the Z-axis, whilst the cutting depth defines the amount of material to be removed by a single cut of the tool, and is given in the X-axis direction. These input

parameters define the machining procedure and provide us with distinctive results where the cutting speed has to be higher for finishing whilst the feed-rate has to be of lower value in order to lower the surface roughness and maximize the tool life. The measured values are represented in Table I (rough turning), and Table II (finish turning). These data sets serve optimization purposes as trained data and test data sets. First 15 results are trained and the last 5 are test data set. A module within the GSA is necessary in order to obtain coefficients when using multiple regression analysis. According to the measured values, training data sets for both rough and finish turning can be realized.

Table I: Input and output values for rough turning.

Nr.	Input values			Output values		
	$V_c$ [m/min]	$f$ [mm/rev]	$a_p$ [mm]	$F_c$ [N]	$R_a$ [ $\mu$ m]	$T$ [min]
1	300	0.30	1.50	879.2240	4.300	17.6
2	400	0.30	1.50	894.3270	3.880	4.73
3	300	0.50	1.50	1436.299	11.11	6.68
4	400	0.50	1.50	1408.114	11.48	1.88
5	300	0.30	3.00	1754.215	4.210	13.8
6	400	0.30	3.00	1726.937	4.500	3.80
7	300	0.50	3.00	2896.122	14.29	4.10
8	400	0.50	3.00	2860.663	13.71	1.16
9	350	0.40	2.25	1677.149	8.100	5.38
10	350	0.40	2.25	1672.771	8.130	5.10
11	350	0.40	2.25	1679.359	8.120	5.44
12	350	0.40	2.25	1678.825	8.120	5.28
13	350	0.40	2.25	1675.829	8.110	5.50
14	350	0.40	2.25	1678.223	8.100	5.22
15	266	0.40	2.25	1697.504	7.820	12.9
16	434	0.40	2.25	1683.361	8.150	1.81
17	350	0.23	2.25	1002.763	2.460	10.5
18	350	0.57	2.25	2609.254	17.95	0.75
19	350	0.40	1.00	765.9210	6.360	6.65
20	350	0.40	3.50	2746.389	9.070	3.58

Table II: Input and output values for finish turning.

Nr.	Input values			Output values		
	$V_c$ [m/min]	$f$ [mm/rev]	$a_p$ [mm]	$F_c$ [N]	$R_a$ [ $\mu$ m]	$T$ [min]
1	400	0.10	0.40	128.893	0.77	32.66
2	500	0.10	0.40	130.755	0.80	11.15
3	400	0.20	0.40	201.899	1.70	25.89
4	500	0.20	0.40	202.200	1.67	7.450
5	400	0.10	1.20	337.859	1.11	28.43
6	500	0.10	1.20	330.745	1.19	9.230
7	400	0.20	1.20	492.945	2.14	20.74
8	500	0.20	1.20	550.848	1.77	5.610
9	450	0.15	0.80	299.005	1.26	14.44
10	450	0.15	0.80	301.647	1.30	14.38
11	450	0.15	0.80	304.772	1.29	14.39
12	450	0.15	0.80	299.519	1.28	14.48
13	450	0.15	0.80	299.875	1.27	14.43
14	450	0.15	0.80	303.832	1.28	14.46
15	366	0.15	0.80	313.225	1.37	34.46
16	534	0.15	0.80	307.622	1.31	6.120
17	450	0.06	0.80	174.024	1.21	20.25
18	450	0.23	0.80	406.719	2.32	10.93
19	450	0.15	0.13	61.2230	1.17	12.18
20	450	0.15	1.47	497.895	1.13	10.05

### 3. GRAVITATIONAL SEARCH ALGORITHM

The gravitational search algorithm is based on gravitational laws [30-32]. As gravitation is a natural phenomenon, it can be described as the pull of masses between two or more objects. In respect to optimization purposes the subjects are mass objects, where their properties are defined by their masses. As those objects with higher mass tend to move faster, this means that higher mass can trigger a local optimum so a compensation tool has to be introduced within the algorithm to compensate for this effect. Such a compensation tool slows down higher masses and so this error can be avoided. All objects within the search system are attracted to each other by gravitational forces that trigger global movements of the masses within the search system. Gravitational forces are described using Newtonian equations:

$$\vec{F} = m \cdot \vec{a} = G \cdot \frac{M_1 \cdot M_2}{R^2} \quad (1)$$

where the symbols stand for:

- $\vec{F}$  – Force,
- $m$  – Mass,
- $\vec{a}$  – Acceleration,
- $M_1$  – Mass / object 1,
- $M_2$  – Mass / object 2,
- $G$  – Gravitational constant,
- $R$  – Euclidean distance between objects 1 and 2.

In order to obtain successful operation of the algorithm, each mass object within the optimization system has to swap information with other objects where gravitational forces are the media for communication. As each object affects every other object directly, certain visibilities within the search space have to be created. By presuming that every object is made of the same material, objects with larger diameters should have larger gravitational fields and thus, in effect, objects with lower masses tend to follow objects with larger masses, defined as having better characteristics. Therefore an optimization coefficient should be introduced to slow down the movements of the larger objects, and speed up the objects with lower masses.

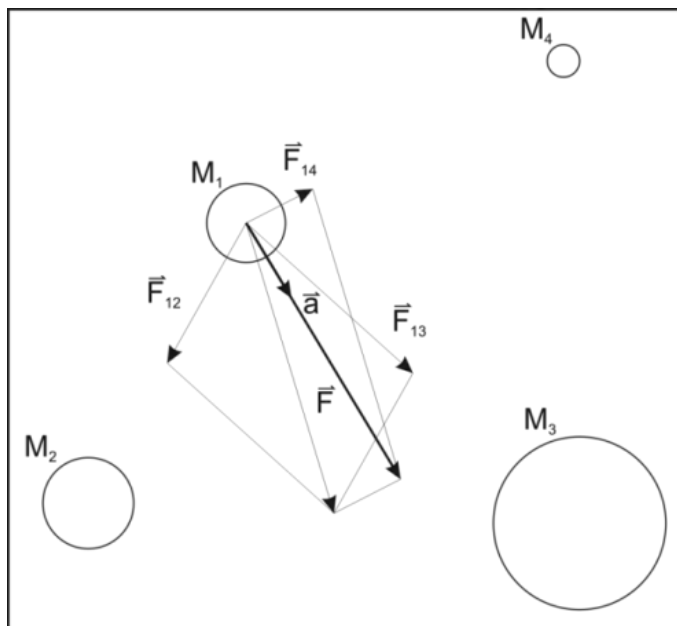


Figure 1: Object interactions within the system with different masses [22].

The attraction force increases if the masses of the objects increase, and this property also affects the lowering of Euclidean distances between them. As the forces and accelerations are vector quantities, the essence of the GSA can be described graphically, as seen in Fig. 1. The mass of an object having index 1 is marked as the central object and every other object affects its movement, where the vector marked  $\vec{F}$  and the acceleration vector marked  $\vec{a}$  are pointing towards the solution within one iteration.

Masses according to objects' types can be divided into three separate groups:

- Active gravitational mass affects within the system of two or more mass objects with gravitational fields that affect other objects the most.
- Passive gravitational mass represents objects with less mass, and affects the gravitational fields of larger objects passively, thus deviating from their movement by equivalently lower values.
- Inertia mass represents the mass value with which an object resists any change of its location or acceleration vector if the system affects its gravitational field. An object with higher inertia mass redirects its vector slower as the force has to be applied over a longer period of time in order to assign to the object's new position or course vector.

Therefore Newtonian equations can be rewritten as:

$$\vec{F} = G \cdot \frac{M_A \cdot M_P}{R^2} \quad (2)$$

$$\vec{a}_1 = \frac{\vec{F}}{M_I} \quad (3)$$

where the symbols stand for:

- $\vec{F}$  – Gravitational force,
- $G$  – Gravitational constant,
- $M_A$  – Active gravitational mass,
- $M_P$  – Passive gravitational mass,
- $M_I$  – Inertia gravitational mass,
- $R$  – Euclidean distance between masses  $M_A$  and  $M_P$ ,
- $\vec{a}_1$  – Acceleration value of object 1.

As the GSA has to work within optimization cycles, each iteration has to be capable of updating vectors and positional values to new positions, thus representing initial acceleration values within time-difference intervals  $\Delta t$  [30]:

$$\vec{a}_a^n = \frac{\Delta \vec{v}_a^n}{\Delta t} \quad (4)$$

Assuming the stochastic characteristics in given time  $t$  and acceleration of mass object  $a$  within  $n$ -dimensional space  $a_a^n(t)$ , recalculated values of object velocity and positions can be expressed as:

$$\vec{v}_a^n(t+1) = rand_a \cdot \vec{v}_a^n(t) + \vec{a}_a^n(t) \quad (5)$$

$$x_a^n(t+1) = x_a^n(t) + \vec{v}_a^n(t+1) \quad (6)$$

where the symbols stand for:

- $rand_a$  – Random factor, assures stochastic behaviour,
- $\vec{v}_a^n(t)$  – Object  $a$  velocity in time  $t$ ,
- $\vec{a}_a^n(t)$  – Object acceleration within the time  $t$ ,
- $x_a^n(t)$  – Object  $a$  position in time  $t$ ,
- $n$  –  $n$ -dimensional search space,

- $\vec{v}_a^n(t + 1)$  – Updated velocity of object  $a$ ,  
 $x_a^n(t + 1)$  – Updated position of object  $a$ .

Gravitational search algorithm optimization success can be defined as a fitness function. As the heavier objects tend to be better indicators for a solution, compensation with severely lighter and faster objects is introduced in order to obtain cooperation of the objects during convergence to an optimal solution. If the assumption of equality regarding gravitational and inertia masses can be proved, distinct equations can be written that serve to define fitness functions [30]:

$$m_a(t) = \frac{fit_a(t) - worst(t)}{best(t) - worst(t)} \quad (7)$$

where the symbols stand for:

- $fit_a(t)$  – Fitness value of the object  $a$  in time  $t$ ,  
 $best(t)$  – Best current value,  
 $worst(t)$  – Worst current value,  
 $m_a(t)$  – Mass object  $a$ .

Where  $fit_a(t)$  value has to be defined differently depending upon the search for either minimum or maximum values, whilst in the proposed case minimization is required:

$$best(t) = \min_{j \in \{1, \dots, N\}} fit_a(t) \quad (8)$$

$$worst(t) = \max_{j \in \{1, \dots, N\}} fit_a(t) \quad (9)$$

As with any other optimization procedure, local optimums are also obtainable. The simplest for eliminating such anomalies is by eliminating certain objects from the search system. This procedure has to be done with caution in order to prevent global optimum removal. As such, the research principle can be initiated by blurring of the search borders, so that the algorithm provides more optional solutions with each new iteration.

#### **4. PROPOSED GSA SYSTEM'S ARCHITECTURE**

In our case, the designed GSA optimization algorithm is constructed from five distinct units, with one additional module that expands the algorithm solver into a multiple regression analysis module, and one module that inserts the optimization function, (i.e. desired polynomial), thus enabling search for polynomial coefficients. The first unit generates search space boundaries with desired space dimensions. In the next step the mass objects are generated within random locations. The third unit establishes physical laws for GSA search space and enables custom parameterisations of certain values. The fourth unit is an evaluation module for object masses, defining the best and worst values within the iteration. The fifth unit updates the velocity vectors' and objects' locations. As the objects are created within the second unit, a multiple regression solver is introduced, which includes a training data set that the GSA can use in order to obtain coefficients for the optimization polynomial. This is located within an additional function optimization module. As such, the pseudo-code for the proposed optimization approach can be given, as shown in Fig. 2.

#### **5. PREDICTION MODEL**

The prediction model is integrated within the regression analysis module and it is manually adjustable. It consists of independent parameters ( $x_1, x_2, x_3$ ) and optimization coefficients ( $k_1, k_2, \dots, k_n$ ).

---

```

1:  START GSA
2:  Establish search space
3:  For each mass object
4:      initialize mass object
5:      run multiple regression analysis module / with measured experimental values
6:  END
7:  DO
8:      For each particle / import measured values
9:      calculate fitness function
10:     If fitness > best particle fitness best (t) → value becomes new best
11:     evaluate worst values as worst (t)
12:  END
13:     For each mass object
14:         calculate velocity, mass and acceleration values
15:         update object velocity and position
16:     END
17:     While → maximal number of iterations N are reached
18: END GSA

```

---

Figure 2: Proposed gravitational search algorithm in pseudo-code.

It is clear from Tables I and II that the values range from the  $10^3$  to  $10^{-2}$  grades, therefore it is extremely difficult to find one single optimization polynomial that would sufficiently describe all the dependent values  $f(x_1, x_2, x_3)$  which are in fact cutting force, surface roughness, and tool life. Several models have been tested, yet the best polynomial that proved to be as accurate as possible for the proposed paper turned out to be composed of ten coefficients that affected the influences of independent variables  $x_1$ ,  $x_2$ , and  $x_3$ :

$$f(x_1, x_2, x_3) = k_1 + k_2 \cdot x_1 + k_3 \cdot x_2 + k_4 \cdot x_3 + k_5 \cdot x_1 \cdot x_2 + k_6 \cdot x_1 \cdot x_3 + k_7 \cdot x_2 \cdot x_3 + k_8 \cdot x_1 \cdot x_1 \cdot x_1 + k_9 \cdot x_2 \cdot x_2 + k_{10} \cdot x_3 \cdot x_3 \quad (10)$$

where the symbols stand for:

$f(x_1, x_2, x_3)$  – output parameters  $F_c$  (cutting force),  $R_a$  (surface roughness), or  $T$  (tool life),  
 $x_1$  – cutting speed  $v_c$ ,  
 $x_2$  – feed rate  $f$ ,  
 $x_3$  – cutting depth  $a_p$ ,  
 $k_1, k_2, \dots, k_{10}$  – coefficients.

In order to obtain optimal results, proper run parameters for the GSA module also have to be chosen. The following run parameters were chosen: maximum number of iterations 1000, search space dimension 10, and the number of mass object 210. The iteration's parameter describes the desired number of iterations within one cycle algorithm run. The number of mass objects defines the number of objects interacting with each other whilst searching for the solution. The search space dimension defines the coefficient numbers with the objects' properties.

## 5.1 The best models

In order to successfully describe each dependent variable, separate polynomial models had to be produced, eqs. (11) to (16). Evaluation of the polynomials was proved on testing the whole measurement range and confirmed by dividing the measurements among 15 results into the learning base and the remaining 5 into the test phase. The results were consistent in both cases.

• Rough turning

$$F_c = 377.41 + 0.37 \cdot x_1 + 69.051 \cdot x_2 - 398.72 \cdot x_3 - 1.3 \cdot x_1 \cdot x_2 - 0.16 \cdot x_1 \cdot x_3 + 2007.785 \cdot x_2 \cdot x_3 + 47323.24 \cdot x_1 \cdot x_1 + 6.07 \cdot x_2 \cdot x_2 + 94.232 \cdot x_3 \cdot x_3 \quad (11)$$

$$R_a = -3.57 + 0.035 \cdot x_1 - 5.42 \cdot x_2 - 2.56 \cdot x_3 - 0.0017 \cdot x_1 \cdot x_2 - 0.0007 \cdot x_1 \cdot x_3 + 8.13 \cdot x_2 \cdot x_3 - 0.00004 \cdot x_1 \cdot x_1 + 37.337 \cdot x_2 \cdot x_2 + 0.127 \cdot x_3 \cdot x_3 \quad (12)$$

$$T = 155.95 - 0.484 \cdot x_1 - 183.89 \cdot x_2 - 7.787 \cdot x_3 + 0.37 \cdot x_1 \cdot x_2 + 0.0148 \cdot x_1 \cdot x_3 + 2.12 \cdot x_2 \cdot x_3 + 0.00033 \cdot x_1 \cdot x_1 + 22.38 \cdot x_2 \cdot x_2 + 0.102 \cdot x_3 \cdot x_3 \quad (13)$$

• Finish turning

$$F_c = 572.99 - 2.084 \cdot x_1 - 870.53 \cdot x_2 + 43.72 \cdot x_3 + 3.154 \cdot x_1 \cdot x_2 + 0.302 \cdot x_1 \cdot x_3 + 1441.71 \cdot x_2 \cdot x_3 + 0.0016 \cdot x_1 \cdot x_1 - 1.225 \cdot x_2 \cdot x_2 - 43.313 \cdot x_3 \cdot x_3 \quad (14)$$

$$R_a = 0.739 - 0.328 \cdot x_1 - 0.003 \cdot x_2 - 0.332 \cdot x_3 + 1.69 \cdot x_1 \cdot x_2 - 0.0242 \cdot x_1 \cdot x_3 - 0.0017 \cdot x_2 \cdot x_3 - 1.163 \cdot x_1 \cdot x_1 + 0.000008 \cdot x_2 \cdot x_2 + 66.361 \cdot x_3 \cdot x_3 \quad (15)$$

$$T = 341.52 - 1.133 \cdot x_1 - 3.031 \cdot x_2 - 9.81 \cdot x_3 + 0.346 \cdot x_1 \cdot x_2 + 0.034 \cdot x_1 \cdot x_3 - 5.499 \cdot x_2 \cdot x_3 + 0.00097 \cdot x_1 \cdot x_1 + 322.26 \cdot x_2 \cdot x_2 - 4.877 \cdot x_3 \cdot x_3 \quad (16)$$

**5.2 Prediction model testing phase and analysis**

The proposed model was proved by utilizing the testing method in order to confirm its accuracy. Utilizing an additional testing module within GSA was the most efficient way, as the optimization coefficients could be inserted directly. Due to the extensive number of test runs and their data information, only the average values and deviation errors are presented in Tables III and IV according to eqs. (11) to (16), the full results are commented on within the next sub-section. In both the following tables, which serve as demonstrational tools, ten predictions are implemented along with five recalculated measurements, as the test-base values.

Table III: Rough turning prediction values.

Cutting force			Surface roughness			Tool life		
Measured	Average	Deviation	Measured	Average	Deviation	Measured	Average	Deviation
$F_c$ [N]	[N]	[%]	$R_a$ [ $\mu$ m]	[ $\mu$ m]	[%]	$T$ [min]	[min]	[%]
1683.360	1743.11	3.55	8.150	8.420	3.34	1.810	1.910	5.47
1002.763	950.442	5.22	2.460	2.350	4.36	10.52	11.15	6.00
2609.254	2443.16	6.37	17.95	17.14	4.53	0.750	0.710	5.98
765.9210	734.081	4.16	6.360	6.820	7.29	6.650	6.320	4.89
2746.389	2659.53	3.16	9.070	9.880	8.97	3.580	3.830	6.90
Deviation average		4.49	Deviation average		5.70	Deviation average		5.85

Table IV: Finish turning prediction values.

Cutting force			Surface roughness			Tool life		
Measured	Average	Deviation	Measured	Average	Deviation	Measured	Average	Deviation
$F_c$ [N]	[N]	[%]	$R_a$ [ $\mu$ m]	[ $\mu$ m]	[%]	$T$ [min]	[min]	[%]
307.622	323.830	5.27	1.31	1.48	12.9	6.120	6.280	2.61
174.024	189.579	8.94	1.21	1.22	0.88	20.25	20.77	2.59
406.719	400.146	1.62	2.32	2.31	0.43	10.93	10.77	1.49
61.2230	63.2560	3.32	1.17	1.20	2.97	12.18	12.52	2.82
497.895	513.868	3.21	1.13	1.16	2.82	10.05	10.42	3.67
Deviation average		4.47	Deviation average		4.00	Deviation average		2.64



Please note that extensive testing of the model was done for all the dependable parameters, with ten separate prediction values and 20 test runs for all the measured values in order to ensure the most accurate average value along with the average deviation error.

The final results of the extensive tests are shown in Tables V and VI. Due to space limitations, in Tables III and IV only the final values for the test base are shown. Figs. 3 and 4 show four graphs, one for each of the regression analysis coefficients' calculations, which serve as bases for calculating the prediction values found in Tables III and IV. Due to space limitation, only four calculations are shown in the forms of graphs for surface roughness and finish turning. The graphs show that even if the input Iterations' values were set to 1000, a value of 600 iterations would be more than sufficient.

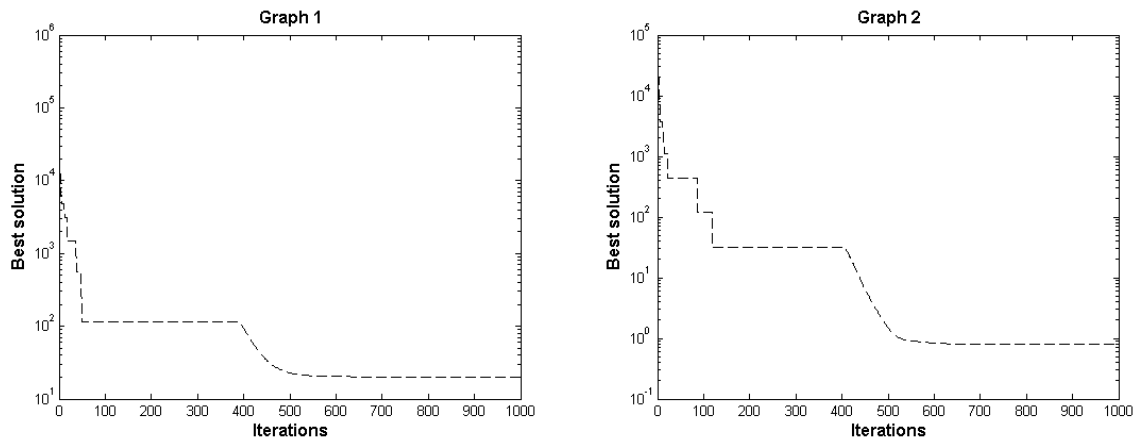


Figure 3: Regression analyses results: prediction 1 (left) and prediction 2 (right).

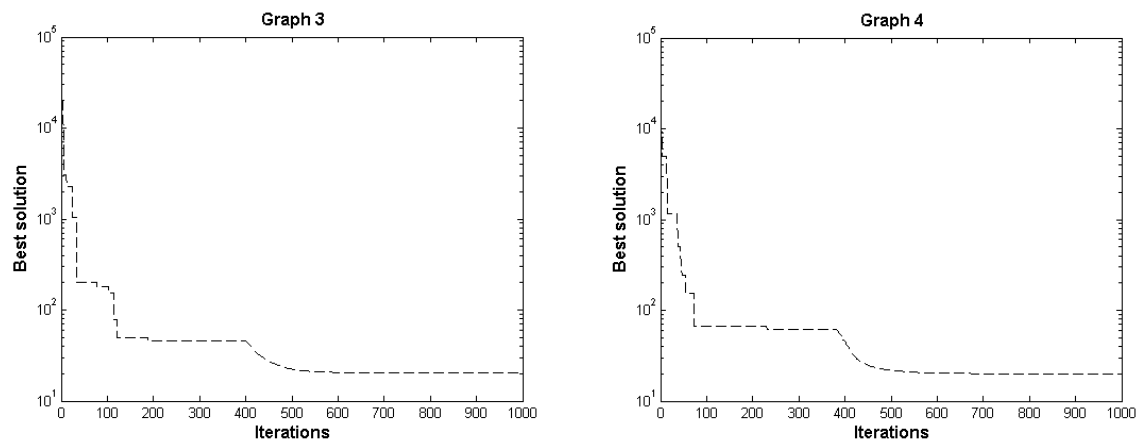


Figure 4: Regression analyses results: prediction 3 (left) and prediction 4 (right).

### 5.3 Results and discussion

The proposed solution utilizing the GSA optimization method is based on CNC-turning. Whilst the experimental results were substantially different for both rough and finish turning, two separate approaches were done, one for rough turning and one for finish turning. In this way more accurate models can be produced for each machining component, as separate polynomials had to be developed for cutting force, surface roughness, and tool-life. The final results are shown in concentrated form in Tables V and VI, showing minimal and maximal values along with the average deviation values of 20 test measurement recalculations. Note, the values in both tables are in percentages.

Table V: Final results for rough turning deviations.

	Cutting force – $F_c$ [%]	Surface roughness – $R_a$ [%]	Tool life – $T$ [%]
Min. deviation	1.04	1.02	0.67
Max. deviation	8.97	8.97	6.90
Average deviation	3.54	4.62	4.15

Table VI: Final results for finish turning deviations.

	Cutting force – $F_c$ [%]	Surface roughness – $R_a$ [%]	Tool life – $T$ [%]
Min. deviation	1.28	0.000	1.49
Max. deviation	8.94	13.30	6.97
Average deviation	3.57	4.020	4.50

In regard to deviation values for the best solutions, the percentage was well-below the expected maximum of 10 %, which is the number of approvable tolerances. For comparison, a mainstream particle swarm optimization algorithm was applied to the same experimental values used for this research, where the average deviations were calculated for cutting forces at 1.75 %, for surface roughness at 5.85 %, and tool-life was calculated with an average deviation of 4.31 %. As the values were below the desired 10 % deviation mark, it can be concluded that both the PSO and GSA algorithms are usable for machining parameters optimization. If more experimental results had been available for the learning base, the solutions would have been even more precise.

## **6. CONCLUSION**

This article proposed a new approach for predicting machining models using the gravitational search algorithm. The results' predictions were based upon Newtonian equations of natural interactions of objects in space, according to gravitational laws. The conclusions can be summarized as follows:

- Accuracy of the output parameters with final testing was well-below the maximal tolerated value of 10 %, with the average value of all deviations being 4.06 %.
- Results of both GSA and mainstream PSO were comparable, which confirms the usability of the newly-developed prediction model using GSA.
- The multiple regression analyses module proved to be an excellent tool for analysing the experimental values.
- Faster data analysis, utilizing the GSA algorithm, as it was modular based.
- Modularity enabled easy algorithm accommodation for different applications.

Further research into the GSA module would explore the possibilities of further reducing the deviation percentage in order to obtain more optimal or even better values. Analysis of GSA implementation for other machining processes such as milling, drilling, laser cutting, should also be researched, as certain output values and machining parameters are clearly similar to certain machining procedures.

## **REFERENCES**

- [1] Chan, F. T. S.; Tiwari, M. K. (eds.) (2007). *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*, I-Tech Education and Publishing, Vienna
- [2] Bharathi Raja, S.; Baskar, N. (2011). Particle swarm optimization technique for determining optimal machining parameters of different work piece materials in turning operation,

- International Journal of Advanced Manufacturing Technology*, Vol. 54, No. 5-8, 445-463, doi:10.1007/s00170-010-2958-y
- [3] El-Mounayri, H.; Dugla, Z.; Haiyan, D. (2013). Prediction of surface roughness in end milling using swarm intelligence, *Proceedings of the 2003 IEEE Swarm Intelligence Symposium SIS '03*, 220-227
- [4] Brezocnik, M.; Jurkovic, Z.; Sekulic, M. (2009). Modeliranje procesa obrade rezanjem primenom genetskih algoritama, *33<sup>rd</sup> Conference on Production Engineering of Serbia 2009*, 199-202
- [5] Cus, F.; Balic, J. (2003). Optimization of cutting process by GA approach, *Robotics and Computer-Integrated Manufacturing*, Vol. 19, No. 1-2, 113-121, doi:10.1016/S0736-5845(02)00068-6
- [6] Sardinas, R. Q.; Santana, M. R.; Brindis, E. A. (2006). Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes, *Engineering Applications of Artificial Intelligence*, Vol. 19, No. 2, 127-133, doi:10.1016/j.engappai.2005.06.007
- [7] Billatos, S. B.; Tseng, P.-C. (1991). Knowledge-based optimization for intelligent machining, *Journal of Manufacturing Systems*, Vol. 10, No. 6, 464-475, doi:10.1016/0278-6125(91)90004-L
- [8] Gusel, L.; Brezocnik, M. (2011). Application of genetic programming for modelling of material characteristics, *Expert Systems with Applications*, Vol. 38, No. 12, 15014-15019, doi:10.1016/j.eswa.2011.05.045
- [9] Gjelaj, A.; Balic, J.; Ficko, M. (2013). Intelligent optimal tool selections for CNC programming of machine tools, *Transactions of FAMENA*, Vol. 37, No. 3, 31-40
- [10] Kokotovic, B.; Glavonjic, M. (2013). Predicting of milling forces in a virtual manufacturing system, *Technical Gazette*, Vol. 20, No. 6, 1027-1035
- [11] Florjanic, B.; Govekar, E.; Kuzman, K. (2013). Neural network-based model for supporting the expert driven project estimation process in mold manufacturing, *Strojniski vestnik – Journal of Mechanical Engineering*, Vol. 59, No. 1, 3-13, doi:10.5545/sv-jme.2012.747
- [12] Kovacic, M.; Rozej, U.; Brezocnik, M. (2013). Genetic algorithm rolling mill layout optimization, *Materials and Manufacturing Processes*, Vol. 28, No. 7, 783-787, doi:10.1080/10426914.2012.718475
- [13] Barisic, B.; Pepelnjak, T.; Math, M. D. (2008). Predicting of the Lüder's bands in the processing of TH material in computer environment by means of stochastic modeling, *Journal of Materials Processing Technology*, Vol. 203, No. 1-3, 154-165, doi:10.1016/j.jmatprotec.2007.09.054
- [14] Giri, B. K.; Pettersson, F.; Saxen, H.; Chakraborti, N. (2013). Genetic programming evolved through bi-objective genetic algorithms applied to a blast furnace, *Materials and Manufacturing Processes*, Vol. 28, No. 7, 776-782, doi:10.1080/10426914.2013.763953
- [15] Kaljun, J.; Dolsak, B. (2012). Ergonomic design knowledge built in the intelligent decision support system, *International Journal of Industrial Ergonomics*, Vol. 42, No. 1, 162-171, doi:10.1016/j.ergon.2011.11.009
- [16] Senveter, J.; Klancnik, S.; Balic, J.; Cus, F. (2010). Prediction of surface roughness using a feed-forward neural network, *Management and Production Engineering Review*, Vol. 1, No. 2, 47-55
- [17] Zuperl, U.; Cus, F. (2003). Optimization of cutting conditions during cutting by using neural networks, *Robotics and Computer-Integrated Manufacturing*, Vol. 19, No. 1-2, 189-199, doi:10.1016/S0736-5845(02)00079-0
- [18] Senthilkumar, N.; Tamizharasan, T.; Anandakrishnan, V. (2013). An ANN approach for predicting the cutting inserts performances of different geometries in hard turning, *Advances in Production Engineering & Management*, Vol. 8, No. 4, 231-241, doi:10.14743/apem2013.4.170
- [19] Saric, T.; Simunovic, G.; Simunovic, K. (2013). Use of neural networks in prediction and simulation of steel surface roughness, *International Journal of Simulation Modelling*, Vol. 12, No. 4, 225-236, doi:10.2507/IJSIMM12(4)2.241
- [20] Cukor, G.; Jurkovic, Z.; Sekulic, M. (2011). Rotatable central composite design of experiments versus Taguchi method in the optimization of turning, *Metallurgija / Metallurgy*, Vol. 50, No. 1, 17-20
- [21] Jurkovic, Z.; Cukor, G.; Andrejcek, I. (2010). Improving the surface roughness at longitudinal turning using the different optimization methods, *Technical Gazette*, Vol. 17, No. 4, 397-402

- [22] Byrne, G.; Dornfeld, D.; Inasaki, I.; Ketteler, G.; König, W.; Teti, R. (1995). Tool condition monitoring (TCM) – The status of research and industrial application, *CIRP Annals – Manufacturing Technology*, Vol. 44, No. 2, 541-567, [doi:10.1016/S0007-8506\(07\)60503-4](https://doi.org/10.1016/S0007-8506(07)60503-4)
- [23] Simunovic, G.; Simunovic, K.; Saric, T. (2013). Modelling and simulation of surface roughness in face milling, *International Journal of Simulation Modelling*, Vol. 12, No. 3, 141-153, [doi:10.2507/IJSIMM12\(3\)1.219](https://doi.org/10.2507/IJSIMM12(3)1.219)
- [24] Ammar, A. A.; Bouaziz, Z.; Zghal, A. (2009). Modelling and simulation of the cutting forces for 2-5 D pockets machining, *Advances in Production Engineering & Management*, Vol. 4, No. 4, 163-176
- [25] Bhushan, R. K. (2013). Optimization of cutting parameters for minimizing power consumption and maximizing tool life during machining of Al alloy SiC particle composites, *Journal of Cleaner Production*, Vol. 39, 242-254, [doi:10.1016/j.jclepro.2012.08.008](https://doi.org/10.1016/j.jclepro.2012.08.008)
- [26] Choudhury, S. K.; Appa Rao, I. V. K. (1999). Optimization of cutting parameters for maximizing tool life, *International Journal of Machine Tools and Manufacture*, Vol. 39, No. 2, 343-353, [doi:10.1016/S0890-6955\(98\)00028-5](https://doi.org/10.1016/S0890-6955(98)00028-5)
- [27] Mgwatu, M. I. (2013). Integrated approach for optimising machining parameters, tool wear and surface quality in multi-pass turning operations, *Advances in Production Engineering & Management*, Vol. 8, No. 4, 209-218, [doi:10.14743/apem2013.4.168](https://doi.org/10.14743/apem2013.4.168)
- [28] Lee, B. Y.; Tarn, Y. S. (2000). Cutting-parameter selection for maximizing production rate or minimizing production cost in multistage turning operations, *Journal of Materials Processing Technology*, Vol. 105, No. 1-2, 61-66, [doi:10.1016/S0924-0136\(00\)00582-3](https://doi.org/10.1016/S0924-0136(00)00582-3)
- [29] Gusel, L.; Rudolf, R.; Romcevic, N.; Buchmeister, B. (2012). Genetic programming approach for modelling of tensile strength of cold drawn material, *Optoelectronics and Advanced Materials – Rapid Communications*, Vol. 6, No. 3-4, 446-450
- [30] Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. (2009). GSA: a gravitational search algorithm, *Information Sciences*, Vol. 179, No. 13, 2232-2248, [doi:10.1016/j.ins.2009.03.004](https://doi.org/10.1016/j.ins.2009.03.004)
- [31] Balachandar, S. R.; Kannan, K. (2010). A meta-heuristic algorithm for set covering problem based on gravity, *International Journal of Computational & Mathematical Sciences*, Vol. 4, No. 5, 223-228
- [32] Shaw, B.; Mukherjee, V.; Ghoshal, S. P. (2012). A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems, *International Journal of Electrical Power & Energy Systems*, Vol. 35, No. 1, 21-33, [doi:10.1016/j.ijepes.2011.08.012](https://doi.org/10.1016/j.ijepes.2011.08.012)
- [33] Cukor, G.; Jurkovic, Z. (2010). Optimization of turning using evolutionary algorithms, *Engineering Review*, Vol. 30, No. 2, 1-10