Efficient evaluation of flatness error from Coordinate Measurement Data using Cuckoo Search optomisation algorithm

Ali M Abdulshahed[1]
Ibrahim Badi[2]
Ahmed Alturas[1]

1- Electrical and Electronic Engineering Department, Misurata University, Misurata, Libya
2- Mechanical Engineering Department, Misurata University, Misurata, Libya

Abstract

In this work, an attempt has been taken to display a brief idea about the applications of Nature-inspired optimisation algorithms in automated manufacturing systems. A new technigue based on the use of the Cuckoo Search optomization algorithm for flatness error estimation is proposed. The proposed technique has been validated and compared with will known optimisation methods, including deterministic and stochastic algorithms. Extensive simulation using Matlab environment in conjuction with measured data has been carried out to show and choose the most suitable and efficient algorithms for a given optimisation task. The analysis results for Cuckoo Search are compared with those obtained by Particle Swarm Optimisation , Convex hull, Improved Convex hull, and Least squares. The implementation proves that the nature-inspired optimisation algorithms outperform traditional algorithms with good global convergence capability and can act as an alternative optimisation method for automated manufacturing problems.

Keywords: CMM, AI, nature-inspired optimisation, Cuckoo Search.
Paper type: Research paper

Introduction

Many efforts have been done in order to perform how a workpiece will be manufactured with a good quality. Optimisation in manufacturing is an important issue because it may result in a shorter machining time, better surface quality and increase productivity. Recent growing interest in quality of manufacturing process has heightened the need for suitable optimisation techniques. Nature-inspired optimisation algorithms can be used for the optimisation of the machining process, also for prediction of the parameters of machining. For instance, researchers have adapted several (AI) methods for tool path optimisation such as, Artificial Neural Network (ANN), and Genetic Algorithms (GA). Although, these algorithms produce good solutions [1-3], they do not ensure that an optimal path will ever be found at the price of a prohibitive cost in computation. Brief details are given below of some common optimisation algorithms used in automated manufacturing systems. References are given either to the work that proposed them or to a more recent discussion of their use.

The classification of optimisation methods is not well established in literature, especially about the use of some terminologies. Generally, classification can be carried out in terms of the number of objectives, number of constraints, function forms, landscape of the objective functions, type of design variables, uncertainty in values, and computational effort. Classification of optimisation algorithms can also be carried out in a simple way; deterministic algorithms and stochastic algorithms.

Deterministic algorithms take the advantages of the analytical properties of the problem to generate a sequence of steps that finally converge to an optimum solution which might not be the global one. Deterministic algorithms will follow the same procedure or path whether the algorithm runs today or tomorrow. In the literature, several deterministic techniques exist such as Convex hull, and Least squares. On another hand, stochastic algorithms always have randomness in their procedure to find the optimal solution and classified as the most recent and powerful computational products of artificial intelligence techniques. Particle Swarm Optimisation (PSO) is a good example; the solution will be different each time, though the final solution may be no big difference, but the path of each particle is not exactly the same. Most of classical algorithms are deterministic. For instance, simplex method in linear programing is deterministic. Other deterministic algorithms used gradient (gradient based algorithms) information such as Newton-Raphson algorithm, steepest descent method, and conjugate gradient method. However, if there is some discontinuity in the objective function, they do not work well [4].

There are two types of stochastic algorithms; heuristic and meta-heuristic. Heuristic means “to find” or “to discover by trial and error”. Meta means “beyond” or “higher level”. Generally, meta-heuristic algorithms are usually considered as a higher level of heuristics, because meta-heuristic algorithms are not simple trial and error, meta-heuristics are designed to learn from past solutions, to be biased toward better moves, to select the best solutions, and to construct sophisticated search moves. Therefore, meta-heuristics can be much better than heuristic [4]. The efficiency of an algorithm can depend on many factors, such as the intrinsic structure of the algorithm, the way it generates new solutions, and the setting of its algorithm-dependent parameters.

Recentelly, Genetic algorithm (GA) and Particle Swarm Optimisation (PSO) are two popular intelligent computation techniques that have attracted much attention in tool path optimisation work [2, 3]. However, GA often suffers from premature convergence and degradation efficiency because of its highly epistatic objective functions, which makes the identified parameters highly correlated [5]. Even the crossover and mutation operations cannot ensure better fitness of offspring, due to the similar structures of chromosomes in the population.

The PSO algorithm was introduced by Kennedy and Eberhart in [6] as an alternative to other evolutionary techniques such as GA. The PSO algorithm is inspired by the behaviours of natural swarms, such as the formation of flocks of birds and schools of fish. The advantages of the PSO algorithm is that it does not require the objective function to be differentiable as in the gradient decent method, which makes few assumptions about the problem to be solved. Furthermore, it has a simple structure and its optimisation method illustrates a clear physical meaning. PSO consists of a population formed by individuals called particles, where each one represents a possible solution of the problem. Each particle *i* has a position *Xi* and tries to search the best position with time in D-dimensional space (solution space). PSO assumes all particles to fly with velocity V_i that is continuously adjused in light of its own experience and its companions' experience, including the current position, velocity and the best previous position experienced by itself and its companions. The motion of each particle can be determined by the following equations:

$$V_{in}^{k+1} = \omega V_{in}^{k} + c_1 r_1 (P_{in}^{k} - X_{in}^{k}) + c_2 r_2 (G_{jn}^{k} - X_{in}^{k}) \quad (1)$$

$$X_{in}^{k+1} = X_{in}^{k} + V_{in}^{k+1} \quad (2)$$

Where k is the iteration number, ω is the inertia weight, n=(1, 2, …. N), r_1 and r_2 are rondom numbers between 0 and 1 standing for the weight that particle gives to its own best position and that for its best neighbour's position, P_{in}=(p_{i2}, p_{i2}, …, p_{iN}) is the best previous position of the i^{th} particle (that gives the best fitness value), and G_{jn}=(g_{j2}, g_{j2}, …, g_{jN}) is the global best position of the best particle (J) in the swarm, c_1 and c_2 are accelerating coefficient that determine the maximum position step size of the particle in a single iteration [6, 7].

Therefore, instead of using the original algorithms, several versions of the PSO algorithm have been proposed in the literature in order to to optimise the automated manufacturing tasks and improve the performance of the original algorithm [8]. Many researches have employed PSO for generation of optimised tool path. Chih-Hsing et al., [2] Used PSO optimisation technique for finding the optimal tool path in 5-axis flank milling of ruled surfaces. A later work [9] used an improved tool path planning method based on PSO algorithms by offering smaller machining error and better planning flexibility. However, the right selection of PSO parameters plays an important role in balancing the global search and local search [10]. In the abovementioned studies, a better solution might be missed when these values are set fixed. Nowdays, an adaptive strategy for tuning PSO parameters also has been used in order to improve the performance of the tool path planning [11].

Cuckoo Search (CS) has been applied in many fields of optimisation and computational intelligence with promising efficiency. Yildiz in [12] has used CS to select optimal machine parameters in milling operation. The results were compared with those obtained using other well-known optimisation techniques such as, ant colony algorithm and genetic algorithms [12, 13]. The obtained results demonstrated that the CS is an effective approach for the optimisation of machining optimisation problems. More recently, Huang et al [14] used a new hybrid algorithm named teaching-learning-based cuckoo search (TLCS) for parameter optimisation problems in structure designing as well as machining processes. Optimisation of drill path can lead to a significant reduction in machining time which improves productivity of manufacturing systems. Lim et al. [15, 16] reported a combinatorial cuckoo search algorithm for solving drill path optimisation problem. The performance of CS algorithm was tested and verified with three different case studies from the literature. The simulation results conducted in this research indicates that the CS algorithm was capable of finding the optimal path for holes drilling process.

From the algorithm analysis point of view, a conceptual comparison of CS with Differential Evolution (DE), PSO, and artificial bee colony (ABC) in [17] suggested that CS and DE algorithms provide more robust results than PSO and ABC. Gandomi et al. [13] provided a more extensive comparison study for solving various sets of structural optimisation problems and concluded that CS in combination with Levy flights obtained better results than other algorithms such as PSO and GA. Generally, the choice of an algorithm for an optimisation task will largely depend on the type of the problem, the nature of the algorithm, the desired quality of solutions, the available computing resources, time limit, availability of the algorithm implementation, and the expertise of the decision makers.

In this paper, an attempt has been taken to display a brief idea about the optimisation algorithms, mostly the nature-inspired optimisation algorithms.

Extensive simulation using Matlab tests have been carried out to show and choose the most suitable and efficient algorithms for a given optimisation task. This work will enable the reader to open the mined to explore possible applications in the field of automated manufacturing systems.

Methodology

Cuckoo Search (CS)

Cuckoo Search (CS) is a meta-heuristic algorithm, introduced in 2009 by Xin-She Yang [18]. It has many advantages due to its simplicity and efficiency in solving highly non-linear optimisation problems with practical engineering applications [19]. CS satisfies the global convergence requirements and supports local and global search capabilities. In addition, CS uses Lévy flights based on the breeding strategy of some cuckoo species as a global search strategy [20]. CS is a stochastic algorithm, inspired by natural behaviour of a family of birds called Cuckoos. Some species of the cuckoo birds engage in an aggressive reproduction strategy; they lay their eggs in the nests of other host birds, which act as surrogate parents. The host bird may notice that the eggs are not their own so it either throws them away or abandons the nest and builds a new one elsewhere. Consequently, Cuckoo eggs have to be incredibly good mimics in order to be accepted into the nest.

In brief, the CS algorithm for global optimisation is based on three rules [4]: (i) each artificial cuckoo lays an egg in a randomly chosen nest in one generation; (ii) nests, which have the high-quality eggs (solutions) will be retained to the next generation; and (iii) the total number of nests is fixed, and a host species can discover an exotic egg with a probability $p_a \in [0, 1]$. Thus, the host bird can either throw the egg away or abandon the nest, and then randomly build a completely new nest in somewhere else. For simplicity in describing the CS algorithm, this last assumption can be estimated by the fraction of p_a of the n nests that are replaced by new nests with new random solutions at new locations. The fitness function of the solution is defined in a similar way as in meta-heuristics evolutionary methods. It is worth pointing out that in this simple algorithm, there is no distinction between a cuckoo, an egg, or a nest, since each nest has a single egg. The aim is to use the new and potentially better solutions to replace worse solutions that are in the nests. Based on these three rules, the basic steps of the CS are described in a pseudo code below:

Algorithm 1. Pseudo code of Cuckoo Search (CS) [18]

1:Objective function: $f(B)$, $B = (b_{i1}, b_{i2}, \dots, b_{iD})^T$;

2: Generate an initial population of n **host nests** b; $i = 1, 2, \dots, M$;

3: ***While*** $(t < MaxGeneration)$ **or (stop criterion)**

4: Get a Cuckoo randomly (say, i**)**

5: Generate a new solution by performing Levy flights;

6: Evaluate its fitness f_i

7: Choose a nest among n **(say,** j**) randomly;**

8: ***If*** $(f_i > f_j)$

9: Replace j **by new solution**

10: end if

11: A fraction (p_a) **of worse nests are abandoned and new ones are built;**

12: Keep the best solutions/nests;

13: Rank the solutions/nests and find the current best;

14: Pass the current best solutions to the next generation;

15: end while;

16: post process results;

17: end

Case study

The Coordinate Measuring Machines (CMMs) have proven to be reliable, flexible and very much suitable for determining the acceptability of manufactured parts. In recent years, CMMs have gained popularity in automated inspection for both the on-line and off-line inspe ction of manufactured components. The data for the evaluation of form errors obtained from CMM will be in Cartesian coordinates given with reference to a system of mutually orthogonal planes and the data combines form and size aspects. This data has to be further processed using appropriate techniques to evaluate the form error. CMM measurement uncertainty because of software has been problematic in the past, and has the ability to be a continued source of uncertainty, especially for minimum circumscribed, maximum inscribed and minimum zone data fits.

Few attempts have been made by previous researchers to develop methods for evaluating flatness error. The least-squares method (LSM) that minimizes the sum of the squared deviations of the measured points from a fitted feature has been suggested [21]. Although the least-squares techniques are based on sound mathematical principles, the error values obtained are not the minimum. The normal least-square fit has also been tried [22], but the values obtained are not the minimum. To obtain the minimum zone solution, numerical methods based on the Monte Carlo, Simplex and Spiral Search echniques [23] have also been suggested. Li et al., [24] has suggested a new simple approach called the convex-hull edge technique that gives the minimum value of form error.

Flatness error

Flatness is one of the most common features in precision coordinate metrology, and various criteria may be used for flatness error evaluation. Flatness error is defined as the distance between two parallel planes that contain the evaluated surface. Assuming $P_i(x_i, y_i, z_i)(i = 1,2, \ldots, n)$ is the measured points extracted by measuring a plane part. A flatness tolerance specifies a tolerance zone defined by two parallel planes within which the surface must lie. If all extracted data-points $P_i(x_i, y_i, z_i)$ are between two parallel planes, the minimum separation between these two parallel planes is called the minimum zone solution (MZS) of flatness error (see خطأ! لم يتم العثور على مصدر المرجع.). Assuming one of the two parallel plane equations of MZS is [25]:

$$z = ax + by + c \quad (3)$$

The distance d_i from datapoints $P_i(x_i, y_i, z_i)$ to the parallel plane is:

$$d_i = \frac{z_i - ax_i - by_i - c}{\sqrt{1 + a^2 + b^2}} \quad (4)$$

The minimum separation f between these two parallel planes is:

$$f = \min(\max(d_i) - \min(d_i))$$

$$= \min\left(max\left(\frac{z_i - ax_i - by_i - c}{\sqrt{1 + a^2 + b^2}}\right) - \min\left(\frac{z_i - ax_i - by_i - c}{\sqrt{1 + a^2 + b^2}}\right)\right)$$

$$= \min\left(max\left(\frac{z_i - ax_i - by_i}{\sqrt{1+a^2+b^2}}\right) - \min\left(\frac{z_i - ax_i - by_i}{\sqrt{1+a^2+b^2}}\right)\right)$$

(5)

Obviously, the minimum separation f is a function of(a, b). Consequently, evaluating the minimum zone flatness error is translated into searching the values of(a, b), so that the separation $f(a, b)$ is the minimum and this minimum value is just the flatness error. It is a non-linear optimisation problem.

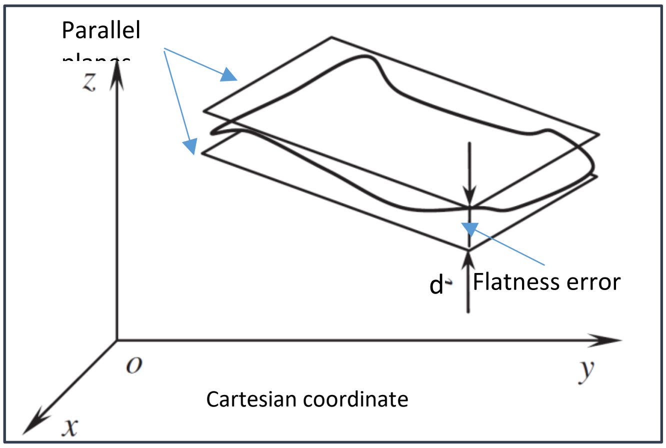

Figure 1: Flatness by minimum zone method.

Results and discussion

To validate the proposed scheme, a case study includes three differentr cases is provided, and the obtained results are compared with other methods presented in the literature. The PSO, convex hull, improved convex hull, and Least squares are developed and employed to estimate the flatness error. The measuremed data from the plane surface are given in Table 1 [26]. The procedures were programmed in the MATLAB environment and the results from different methods were demonstrated in Table 2. Deterministic methods, although effecient at a small scale, become impractical in large-scale problems. In such a case, nature-inspired optimisation algorithms are necessary. As shown in the table, the comparison shows that the global optimum solution of flatness evaluation problem using the CS and PSO can be give the exact solution. The CS and PSO algorithms take about 100 iterations to find the minimum zone solution of the flatness error, and the calculation results of flatness using the CS and PSO algorithm are 1.961161 µm and 1.960123 µm, respectifly.

Table 1: Comparison of the results calculated by different methods.

Case study	No. of points	Flatness tolerance				
		CS	PSO	Improved Convex hull	Convex hull	Least squares
Case study 01	15	1.960123	1.961161	1.972027	2.3774	2.3774
Case study 02	25	0.15321	0.15485	0.155150	0.1756	0.1856
Case study 03	25	0.002214	0.002227	0.002627	0.002817	0.00303

Conclusions

Robust optimisation tool for Coordinate Measuring Machines is becoming ever more important because of current industrial demands for higher productivity at increasing quality levels. In this work, a new intelligent technique based on the use of the Cuckoo Search optimisation algorithm for flatness error estimation is proposed. Extensive simulations using Matlab environment and measured data in conjunction with deterministic and nature-inspired optimisation algorithms have been carried out to verify the and show the effectiveness of the proposed scheme. The proposed algorithm has been validated and compared with Particle Swarm Optimisation , Convex hull, Improved Convex hull, and Least squares. Simulations and comparison show that the CS algorithm outperforms the PSO and other conventional algorithms, which can act as an alternative optimisation algorithm for CMM flatness error software that can be used for quality control. It can therefore be concluded that it is possible to optimise a flatness error using the Cuckoo Search algorithm, which can be used to determining the acceptability of manufactured parts. Future studies will concentrate on applications in other automated manufacturing systems under different operation environments.

References

[1] J. Ni, "CNC machine accuracy enhancement through real-time error compensation," *Journal of manufacturing science and engineering,* vol. 119, pp. 717-725, 1997.

[2] E. Abele, Y. Altintas, and C. Brecher, "Machine tool spindle units," *CIRP Annals - Manufacturing Technology,* vol. 59, pp. 781-802, 2010.

[3] W. Grzesik, "Experimental investigation of the cutting temperature when turning with coated indexable inserts," *International Journal of Machine Tools and Manufacture,* vol. 39, pp. 355-369, 1999.

[4] X.-S. Yang, *Nature-inspired metaheuristic algorithms*: Luniver press, 2010.

[5] M. Ye, X. Wang, and Y. Xu, "Parameter identification for proton exchange membrane fuel cell model using particle swarm optimization," *International Journal of Hydrogen Energy,* vol. 34, pp. 981-989, 1// 2009.

[6] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, 1995, pp. 39-43.

[7] A. Abdulshahed, A. P. Longstaff, and S. Fletcher, "A particle swarm optimisation-based Grey prediction model for thermal error compensation on CNC machine tools," in *Laser Metrology and Machine Performance XI, LAMDAMAP 2015*, Huddersfield, 2015, pp. 369-378.

[8] S. Liang, M. Rajora, X. Liu, C. Yue, P. Zou, and L. Wang, "Intelligent Manufacturing Systems: A Review," *International Journal of Mechanical Engineering and Robotics Research,* vol. 7, pp. 324-330, 2018.

[9] J. Freeman, A. White, and D. G. Ford, "Ball-screw thermal errors-a finite element simulation for on-line estimation," in *Laser Metrology and Machine Performance V*, Southampton, 2001, pp. 269-278.

[10] S. Fletcher and D. G. Ford, "Measuring and modelling heat transfer and thermal errors on a ballscrew feed drive system," in *Laser Metrology and Machine Performance VI*, 2003, pp. 349-360.

[11] P. B. Fernandes, R. C. L. De Oliveira, and J. V. F. Neto, "A Modified QPSO for Robotic Vehicle Path Planning," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018, pp. 1-7.

[12] A. R. Yildiz, "Cuckoo search algorithm for the selection of optimal machining parameters in milling operations," *The International Journal of Advanced Manufacturing Technology,* vol. 64, pp. 55-61, 2013.

[13] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with computers,* vol. 29, pp. 17-35, 2013.

[14] J. Huang, L. Gao, and X. Li, "An effective teaching-learning-based cuckoo search algorithm for parameter optimization problems in structure designing and machining processes," *Applied Soft Computing,* vol. 36, pp. 349-356, 2015.

[15] W. C. E. Lim, G. Kanagaraj, and S. Ponnambalam, "PCB drill path optimization by combinatorial cuckoo search algorithm," *The Scientific World Journal,* vol. 2014, 2014.

[16] W. C. E. Lim, G. Kanagaraj, and S. Ponnambalam, "Cuckoo search algorithm for optimization of sequence in pcb holes drilling process," in *Emerging trends in science, engineering and technology*, ed: Springer, 2012, pp. 207-216.

[17] P. Civicioglu and E. Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review,* vol. 39, pp. 315-346, 2013.

[18] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 210-214.

[19] X.-S. Yang, *Cuckoo search and firefly algorithm* vol. 37: Springer, 2014.

[20] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *arXiv preprint arXiv:1307.4186,* 2013.

[21] G. Samuel and M. Shunmugam, "Evaluation of circularity and sphericity from coordinate measurement data," *Journal of materials processing technology,* vol. 139, pp. 90-95, 2003.

[22] G. Samuel and M. Shunmugam, "Evaluation of straightness and flatness error using computational geometric techniques," *Computer-Aided Design,* vol. 31, pp. 829-843, 1999.

[23] G. Samuel and M. Shunmugam, "Evaluation of circularity from coordinate and form data using computational geometric techniques," *Precision engineering,* vol. 24, pp. 251-263, 2000.

[24] P. Li, X.-M. Ding, J.-B. Tan, and J.-W. Cui, "A hybrid method based on reduced constraint region and convex-hull edge for flatness error evaluation," *Precision engineering,* vol. 45, pp. 168-175, 2016.

[25] X.-L. Wen, X.-C. Zhu, Y.-B. Zhao, D.-X. Wang, and F.-L. Wang, "Flatness error evaluation and verification based on new generation geometrical product specification (GPS)," *Precision Engineering,* vol. 36, pp. 70-76, 2012.

[26] X. Zhu and H. Ding, "Flatness tolerance evaluation: an approximate minimum zone solution," *Computer-Aided Design,* vol. 34, pp. 655-664, 2002.

Appindeces

Table 2: The measurement data from the plane surface.

No	x (μm)	y (μm)	z (μm)
1	-2	1	5
2	-1	1	4
3	0	1	1
4	1	1	2
5	2	1	2
6	-2	0	4
7	-1	0	3
8	0	0	3
9	1	0	2
10	2	0	2
11	-2	-1	3
12	-1	-1	4
13	0	-1	2
14	1	-1	1
15	2	-1	2

Table 3: The measurement data from the plane surface.

No	x (µm)	y (µm)	z (µm)
1	0.2	0.2	-0.0664
2	0.2	0.4	-0.0644
3	0.2	0.6	0.0088
4	0.2	0.8	-0.0112
5	0.2	1	-0.0624
6	0.4	0.2	-0.0383
7	0.4	0.4	0.0655
8	0.4	0.6	0.0636
9	0.4	0.8	0.0285
10	0.4	1	-0.0061
11	0.6	0.2	-0.0952
12	0.6	0.4	-0.0115
13	0.6	0.6	-0.0241
14	0.6	0.8	0.0352
15	0.6	1	-0.02
16	0.8	0.2	0.0154
17	0.8	0.4	-0.0132
18	0.8	0.6	-0.0222
19	0.8	0.8	0.0771
20	0.8	1	-0.0004
21	1	0.2	0.0577
22	1	0.4	-0.0562
23	1	0.6	0.0921
24	1	0.8	0.0654
25	1	1	-0.0212

Table **5**: The measurement data from the plane surface.

No	x (μm)	y (μm)	z (μm)
1	0.2556	0.2994	0.0005
2	1.4992	0.3371	0.0013
3	2.6656	0.3726	0
4	3.5978	0.4009	0.0005
5	4.6241	0.4321	-0.0007
6	4.5989	1.264	0.0001
7	3.4451	1.2289	0.0008
8	2.7096	1.2066	0.0004
9	1.6726	1.2968	0.0014
10	0.5273	1.262	0.0009
11	0.1683	2.1413	-0.0002
12	0.9906	2.1663	0.001
13	2.5485	2.1801	0.0008
14	3.4605	2.1369	0.0011
15	4.8632	2.1795	-0.0017
16	4.8401	2.9417	-0.0014
17	3.6557	2.9058	0.0012
18	2.4224	2.8683	0.0012
19	1.3839	2.8368	0.0011
20	0.4966	2.8098	-0.0002
21	0.4672	3.7751	-0.0008
22	1.6709	3.8116	0.001
23	2.8864	3.8486	0.0006
24	3.7562	3.875	0.0008
25	4.6746	3.9029	-0.0003