Investigation of Octane Number Loss Based on Particle Swarm Optimization

Huawen Yang*
Wenzhou University, Wenzhou, Zhejiang Province, China
632297365@qq.com

Zihao Wang
Wenzhou University, Wenzhou, Zhejiang Province, China
wangzihao411318@foxmail.com

Liang Chen
Wenzhou University, Wenzhou, Zhejiang Province, China
l325467@qq.com

ABSTRACT

The Particle swarm optimization (PSO) is a swarm intelligence algorithm that simulates the predatory behavior of birds. It is inspired by the social behavior of bird flocking. It is widely used in many fields because of its easy implementation, high accuracy, and fast convergence. In this article, we propose a method to improve the performance of the PSO algorithm by combining it with a gradient boosting regression (GBR) model. We apply our algorithm for the optimization of octane number (express in RON) loss in the gasoline industry. RON is the most significant indicator that reflects the combustion petrol performance and it is the commercial brand name of petrol (e.g., 89#, 92#, 95#). Our simulation results demonstrate that RON average loss rate was greater than 30%, under the product's sulfur content was no greater than 5μg/g (Euro VI standard is no greater than 10μg/g).

CCS CONCEPTS

• **Theory of computation**; • **Theory and algorithms for application domains**; • **Computing methodologies**; • **Artificial intelligence**;

KEYWORDS

Particle swarm optimization algorithm, Octane number, Gradient boosting regression

ACM Reference Format:
Huawen Yang*, Zihao Wang, and Liang Chen. 2021. Investigation of Octane Number Loss Based on Particle Swarm Optimization. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3487075.3487086

1 INTRODUCTION

Soft computing techniques have shown great influences on engineering [1, 2], compared to the traditional studies. There are many optimization algorithms that come as a response to different problems. The particle swarm optimization algorithm, also known as the bird swarm foraging algorithm, is one of the intelligent optimization algorithms. It is a new evolutionary algorithm (EA), which originates from the study of natural creatures living in a group. Each individual possesses memory and experience. By interacting with each other or the surrounding environment, a group can carry out very complex tasks. It was proposed by J. Kennedy and R. Eberhart [3]. It uses rules that is simpler than those of genetic algorithm (GA) [4].

The PSO algorithms is a global random search algorithm based on swarm intelligence [5], which has a vast range of applications in many researches [6]. The superiority of the PSO algorithm in solving practical problems has been proved many times. For example, PSO optimization can give excellent results in the image processing of tidal current distribution learning [7]. The PSO algorithm in mathematics can help scholars improve the Euler method [8]. There have been many improvements over the PSO algorithm. In [9], a new two-subgroup particle swarm optimization algorithm was proposed, which had been successfully used in training artificial neural networks for the construction of a practical soft sensor in main engine Fractionator fluid catalytic cracking unit. Z Wu et al [10] further developed a new particle swarm optimization by partitioning the particle swarms into two equal-sized subgroups, of which the first adopts PSO model to evolve, and the second iterates by a cognitive model. [11] improves the convergence speed of the PSO algorithm by using a new searching equation which take into considerations of individual experience, social experience, and their integration. Multi-objective optimization PSO is also proposed, with its engineering applications being analyzed that can better determine the direction of future research [12]. To be more precisely, its excellent optimization effect makes the PSO algorithm very popular. However, the PSO algorithm is known to be easy in falling into the local optimum in some problems [13, 14], which leads to the decline of its accuracy. Some approaches have been proposed to overcome this drawback. In [15], the inertia weight was introduced in the PSO algorithm, which improved the algorithm's global search performance. A PSO model with Constriction Coefficient was constructed in [16], and a method is adopted to dynamically adapt the inertia weight of the PSO [17]. This paper represents a new attempt in resolving the problem. The optimization performance based on our proposed PSO algorithm is evaluated by data experiments.

2 PSO ALGORITHM THEORY

2.1 Standard Particle Swarm Algorithm

As above mentioned, the PSO algorithm is advocated to simulate the behaviour of migratory birds when seeking food. In other words, it is an abstract mathematical model that studies the behaviour of birds when searching for food. The model has following features:

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487086

1. Each bird is described as a point and called "particles". Each particle represents a solution to an optimization problem, and all particles search for the most optimal solution in a solution space.
2. All particles have a fitness value determined by an objective function, which is used to determine whether the current position is either good or bad.
3. The speed of each particle determines its distance and direction during flight for one step forward. Meanwhile, every particle has a memory function, to be used to remember the best position and make corresponding judgments.
Generally, with N particles in a D-dimensional solution space forming a community, a vector group is defined as:

$$\begin{cases} x_i = (x_{i1}, x_{i2}, \cdot s, x_{iD}) \\ v_i = (v_{i1}, v_{i2}, \cdot s v_{iD}) \\ Pbest_i = (p_{i1}, p_{i2}, \cdot s, p_{iD}) \\ Gbest = \left(p_{g1}, p_{g2}, \cdot s, p_{gD}\right) \end{cases} \tag{1}$$

Where, vector $x_i = (x_{i1}, x_{i2}, \cdot s, x_{iD})$represents the position of the i th particle, and vector$v_i = (v_{i1}, v_{i2}, \cdot s, v_{iD})$ represents the velocity of the i th particle. $Pbest_i$is the best position experienced by the individual, while Gbest is the best position reported by the group experience. To prevent surpassing the limit, the range of position change is limited to $(x_{min,d}, x_{max,d})$ in the d-dimensional ($1 \le d \le D$) space, and the speed change range is limited to $(-v_{max,d}, v_{max,d})$. Given the position and velocity of a particle, there is an updating formula for the position and velocity of the particle, also called iterative formula:

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_1 r_1 \left(Pbest_{id} - x_{id}^{k}\right) + c_2 r_2 \left(Gbest_g - x_{id}^{k}\right) \tag{2}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{3}$$

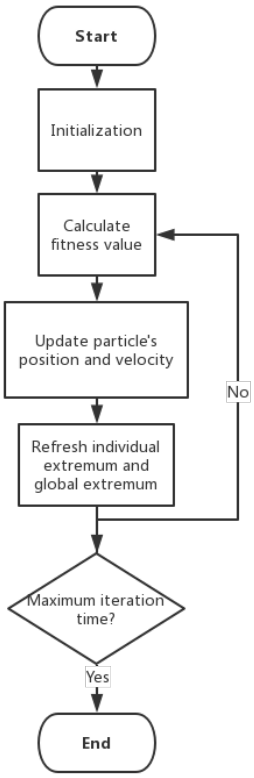
where, v_{id}^k and x_{id}^k are the d-dimensional components of the velocity and position of the particle i in k th iteration, ω is the inertia weight, c_1 and c_2 are learning factors. r_1 and r_2 are random numbers between 0 and 1. By continuously updating equations 2) and (3), the optimized convergence point can be obtained.

2.2 The Processing of PSO Algorithm

Algorithm 1 The PSO algorithm and flow chart are given as follows:

Start:
Step 1 Initialization:
1.The velocity vector of particles;
2.The position vector of particles;
Step 2 Calculating fitness value:
3.Calculating the fitness value of each particle;
Step 3 Updating:
4.Update the particle position and velocity based on fitness value;
5.Update Pbest and Gbest;
Step 4 Decision:
6.If satisfied with the stop condition, Continue to the next step;
Other else return the step 2 and repeat the above iteration;
Step 5 Output:
7. Output the corresponding results of the best particle;
End

PSO FLOW CHART



2.3 The Improved PSO Algorithm

The improved particle swarm algorithm formulas are as it follows: All particles are divided into 3 parts, according to a certain proportion. The ratio of each part of the particles is X, Y, Z (X+Y+Z=1). Since the particles are easy to fall into the local optima in the later stage, consequently, let X=1/4, Y=1/2, Z=1/4.
This part of the particles that account for 1/4 of the total is close to its individual extreme Pbest:

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_1 r_1 \left(Pbest_{id} - x_{id}^{k}\right) \tag{4}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \tag{5}$$

This part of the particles that account for 1/2 of the total approach global extreme Gbest:

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_2 r_2 \left(Gbest_{id} - x_{id}^{k}\right) \tag{6}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \tag{7}$$

This part of the particles that account for 1/4 of the total is close to its historical best Ibest in each iteration:

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_3 r_3 \left(Ibest_{id} - x_{id}^{k}\right) \tag{8}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \tag{9}$$

The inertial weight, in equation(2) given as equation(10).

$$\omega = \omega_{max} - \frac{f_i}{f_{max}} (\omega_{max} - \omega_{min}) \tag{10}$$

Where, ω_{max} and ω_{min} are the maximum and minimum number of weighting factors. f_i is the fitness value of i th particle, and f_{max} is maximum fitness value of it. In PSO theory, the inertia weight plays an important role in the convergence. We dynamically adjust its value according to the relation between the fitness value of the particles and Gbest in this algorithm. The flows of improved PSO algorithm are below shown.

Algorithm 2 The Improved PSO ALGORITHM

Improved PSO FLOW
1. Construct the initial population;
2. Utilize the trained model to evaluate the fitness function for each generation of individuals;
3. Update the position and speed of the individual according to the above formula;
4. Ascertain whether the maximum number of iterations is reached; otherwise, return to Step 2;
5. Output the global optimal solution;

3 EXPERIMENTS

3.1 Application of Gasoline Octane Number

In engineering practice, the focus of petrol cleaning is reducing the content of sulfur and paraffin in petrol while maintaining its octane number as much as possible. The loss value of octane number averages between 1.37 units and the minimum value of the similar devices is only 0.6 units [18]. Consequently, there is a lot of room for optimization consequence. We analyzed the data produced for four years from a petrochemical company, and used them in the experiment. The results are implemented on the Pycharm platform by using the Python software package.

On the one hand, the sulfur content in the raw materials is removed as much as possible, with the objective below 5 μg/g. On the other hand, the octane number of the product, representing the quality of gasoline, is guaranteed to be as high as possible. As a result, the RON loss should be decreased in the refining process.

We chose 29 possible representative and independent host variables from 367 possibilities (seven raw material properties, two spent adsorbent properties, two regenerated adsorbent properties, two product properties and another 354 operating variables) through the normalization of tabular data and reduction of the dimensionality. The 29 variables are shown in Table 1

3.2 The Construct of Predict Model

Based on the traditional machine learning methods [19], the boosting regression can be used for the prediction. Boosting sets up a strong learner by combining weak learners. AdaBoost and Boosting tree are the most famous algorithms. In order to improve the accuracy of the prediction model, three indicators were considered for three regression models (AdaBoosting regression, Gradient Boosting Regression and xgboost): The Mean Absolute Error (MAE), Mean Absolute Percentage Error(MAPE), Root Mean Square Error (RMSE). Ron loss and sulfur content are predicted by choosing the best one out of three regression models.

Table 1: The 29 Variables

Raw material RON	S-ZORB.AT_5201.PV
Raw material saturated hydrocarbon,v%	S-ZORB.SIS_TEX_3103B.PV
Raw material olefin,v%	S-ZORB.TE_7504.DACA
S-ZORB.FT_9401.PV	S-ZORB.TE_5001.DACA
S-ZORB.FT_1002.PV	S-ZORB.FT_1301.DACA
S-ZORB.FC_3103.PV	S-ZORB.AT_1001.DACA
S-ZORB.A-0010.DACA.PV	S-ZORB.TE_7106B.DACA
S-ZORB.A-0004.DACA.PV	S-ZORB.TE_5008.DACA
S-ZORB.FT_5104.PV	S-ZORB.TE_1105.PV
S-ZORB.TC_2801.PV	S-ZORB.PDI_1102.PV
S-ZORB.FC_2801.PV	S-ZORB.AT_1001.PV
S-ZORB.TE_2601.PV	S-ZORB.FT_9403.PV
S-ZORB.PC_2105.PV	S-ZORB.FC_3101.PV
S-ZORB.FT_1001.PV	S-ZORB.TE_5201.DACA
S-ZORB.TE_1001.PV	

The 29 main variables are divided into the training set and the test set with 8:2. The linear regression model is therefore trained. It is necessary to perform k-fold cross-validation on the model (where, k=10). AdaBoosting regression, Gradient Boosting Regression and xgboost were used to establish the regression model. The values of MAPE, MAE and RMSE are shown in Table 2.

The data in the table shows that Gradient Boosting Regression (GBR) is most effective as the average of its three smallest indices is smaller than those of the other two models. For this reason, we use GBR to predict product RON and sulfur content. The result (take the first 10) is shown in Table 3 below:

3.3 Optimizing RON Loss

We need to consider the factors of the main 26 variables (Remove three non-operating variables from 29 variables). The raw materials were taken into consideration during the optimization process and the properties of the spent adsorbent and regenerated adsorbent remained unchanged. The improved PSO algorithm and GBR model are used to optimize the Ron loss.

The next step is to calculate fitness values F_1 and F_2. They respectively correspond to the predictive value of the sulfur content and to the predictive value of the RON loss ratio. We apply the improved PSO algorithm in order to find the optimal solution, and then use the established sulfur content and Ron loss prediction model to anticipate them in the optimal solution. The total fitness value F = $aF_1 + bF_2$(a+b=1), and adjust a, b in table 4 and related parameters in table 5, the best parameter values and optimized results are as shown below:

The improved method optimized the sulfur content and Ron loss of 325 samples. The optimized sulfur content and Ron loss from 325 samples are shown in Figure 1 and Figure 2

As it can be seen in the Figure 2, the octane number loss of most samples can be reduced by more than 30%. Furthermore, it is ensured that under the constraint, the sulfur content is no more than

Table 2: The RON and Sulfur Content of Three Indicators

ModelIndicators	AdaBoosting Regression	Gradient Boosting Regression	Xgboost
RON_MAE(%)	0.25205590	0.20739302	0.23081048
RON_MAPE(%)	0.284972849	0.234871555	0.261357919
RON_RMSE(%)	0.31948939	0.26360741	0.29326059
S_ MAE	1.08675993	0.80330176	0.77734607
S_MAPE	30.47090266	21.03690858	20.56208544
S_ RMSE	1.397028651	1.237394766	1.244449685

Table 3: Sulfur and RON Predicted Value

RON value	Sulfur content value	Predicted RON value	Predicted Sulfur content value
87.02	3.2	86.69397693	3.403884026
89.2	6.6	89.20616846	5.664413107
90.12	3.2	90.12820039	3.383784317
88.02	3.2	87.58520853	2.93067094
88.12	3.2	88.13253059	3.767953509
89.78	3.5	89.70735273	4.45907903
88.36	3.2	88.50659545	3.632891122
87.52	3.2	87.70223066	9.323975227
87.72	3.2	87.40242234	4.06039073
88.5	7.6	88.50117914	6.702861094

Table 4: Various Parameters

Parameter	r	c_1	c_2	c_3	a	b	T	V_{max}	d	ω	number
Value	[0,1]	2	2	2	0.2	0.8	50	d/20	26	[0.1,0.9]	20

Table 5: Ron Optimization Results

a	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
b	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
RON average Loss Rate	0.34	0.40	0.32	0.31	0.30	0.36	0.33	0.33	0.31

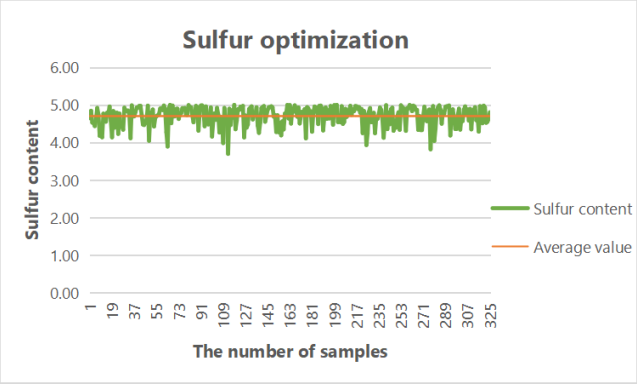


Figure 1: The Sulfur Content Optimization.

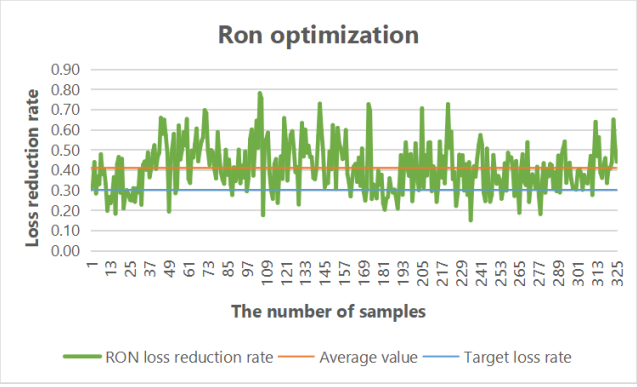


Figure 2: RON Loss Optimization.

Table 6: A Fraction of the Data

S-ZORB.TE_1001.PV	62	49	74	68	40	48	43	62	42	65
S-ZORB.AT_1001.PV	630	390	150	420	190	190	280	370	280	440
S-ZORB.PDI_1102.PV	0.15	0.15	0.15	0.05	0.05	0.2	0.05	0.1	0.05	0.2
S-ZORB.FC_2801.PV	700	700	650	850	700	750	850	800	700	700
S-ZORB.AT-0004.DACA.PV	0.6	0.5	0.7	0.7	0.5	0.5	0.7	0.7	0.7	0.7
S-ZORB.AT-0010.DACA.PV	0.6	1.6	1.5	1.2	1.6	1.1	0.9	1.6	0.7	1.6
optimized_S	4.54	4.72	4.46	4.06	4.61	4.65	4.92	4.94	4.52	4.52
optimized_RON	0.28	0.33	0.39	0.36	0.31	0.29	0.49	0.42	0.40	0.43

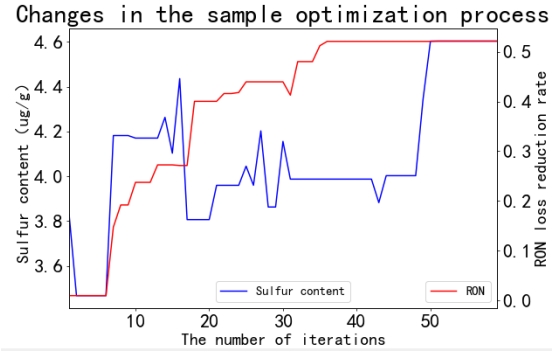


Figure 3: The RON Loss and Sulfur Content Trajectory of Sample No. 133.

5μg/g. We find out from Table 4, when a=0.2, b=0.8, that the optimization effect is the best. The average reduction ratio of the RON loss value is the highest and reaches 40%.

We select those whose RON loss reduction is greater than 30% from 325 samples and only show the partial optimized data in Table 6

As shown in Figure 3, only by taking the No. 133 optimized sample it is possible to better understand the change trajectory of the corresponding RON loss and sulfur content during the optimization and adjustment process of the main variables.

4 CONCLUSION

The classical bionics algorithm known as PSO algorithm is introduced in this paper. Moreover, a improvement based on PSO has been applied to optimize the RON loss and sulfur content during the refining process of gasoline. The particle swarms are partitioned into three parts of sizes 1/4, 1/2 and 1/4 respectively. The inertia weights are dynamically adjusted according to the fitness values of the particles. In the particle swarm, the individual fitness value is kept to be as close as possible to the maximum fitness value of this part of the particles. It prevents the PSO algorithm fall into local optima and enhances the particle's diversity. By combing with the GBR regression model for prediction, the Ron loss is reduced as much as possible and a better optimization effect is being achieved. The algorithm proposed in this paper can not only be applied to RON optimization problem, but also be to other practical engineering projects.

However, to a certain extent, this algorithm has some shortcomings. It cannot guarantee a more stable convergence effect in the update of inertial weights. In future research, it needs to be combined with the other algorithms to enhance the optimized efficiency of the model.

ACKNOWLEDGMENTS

This article was offered help and guidance by the tutors and students. The author would like to express thanks to all reviewers for constructive and valuable suggestions on this paper.

Contribution of authors. HY and ZW: design of the approach, implementation of the algorithm and the experiments, data analysis, writing and editing the manuscript; LC: supervision and guidance.

REFERENCES

[1] W Zhang *et al* (2020). State-of-the-art review of soft computing applications in underground excavations. Geoscience Frontiers, vol.11, pp. 1095-1106.
[2] S Isam and W Zhang (2021). Use of the Soft Computing techniques for TBM tunnelling optimization. Underground Space, 6(3): 233-239.
[3] J Kennedy and R Eberhart (1995). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, vol.4, pp. 1942-1948, DOI: 10.1109/ICNN.1995.488968.
[4] M Pluhacek *et al* (2013). On the behavior and performance of chaos driven PSO algorithm with inertia weight[J]. Computers & Mathematics with Applications, vol. 66, pp. 122-134.
[5] Z Gui, X Gao (2012). Theory and applications of swarm intelligence. Neural Computing and Applications, vol.21, pp. 205-206.
[6] Y D Zhang, S H Wang and G L Ji (2015). A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. Mathematical Problems in Engineering, vol.2015, pp.1-38.
[7] C Huang (2021). Particle swarm optimization in image processing of power flow learning distribution. Discover Internet of Things, vol. 1, pp. 1-12.
[8] X C Zhong, J Y Chen and Z Y Fan (2019). A Particle Swarm Optimization-Based Method for Numerically Solving Ordinary Differential Equations, Mathematical Problems in Engineering, vol. 2019, pp. 1-11, December, DOI: 10.1155/2019/9071236.
[9] G C Chen and J S Yu (2007). Two sub-swarms particle swarm optimization algorithm and its application.Control Theory & Applications, vol. 24, pp. 294-298.
[10] Z Yu, W Wu and L Wu (2012). An Improved Particle Swarm Optimization Algorithm Based on Two Sub-swarms. Advances in Computer Science and Information Engineering, vol. 2, pp. 443-448.
[11] C F Wang and Liu K (2018). An improved particle swarm optimization algorithm based on comparative judgment. Natural Computing, vol. 17, pp. 641-661.
[12] D Wang,D Tan and L Liu (2018). Particle swarm optimization algorithm: an overview. Soft Computing, vol. 22, pp. 387-408.
[13] W B Langdon and R Poli (2005). Evolving problems to learn about particle swarm and other optimizers. Evolutionary Computation, vol. 1, pp. 81-88.
[14] M Jiang, Y Luo and S Yang (2007). Stagnation Analysis in Particle Swarm Optimization. IEEE Swarm Intelligence Symposium, pp. 92-99, DOI: 10.1109/SIS.2007.368031.
[15] Y Shi and R Eberhart (1998). A modified particle swarm optimizer. IEEE International Conference on Evolutionary Computation Proceedings, pp. 69-73, DOI: 10.1109/ICEC.1998.699146.

[16] M Clerc and J Kennedy (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 6(1): 58-73.
[17] Y Shi and R Eberhart (2001). Fuzzy adaptive particle swarm optimization. Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, vol. 1, pp. 101-106.
[18] W Li *et al* (2021). Characteristic Selection and Prediction of Octane Number Loss in Gasoline Refinement Process. E3S Web of Conferences, 245(8):01040, DOI:10.1051/e3sconf/202124501040.
[19] T Mitchell *et al* (1990). Machine Learning. Annual Review of Computer Science, vol. 4, pp. 417-433.