ДОДАТОК А

Перелік посилань на наукові дослідження університету

4. Корабльов М.М., Макогон А.Е. & Фомічов О.О. (2011). Аналіз збіжності імунних алгоритмів. Системи обробки інформації, 29-33. 14 лютого, 2011, Харків, Україна.

8. Korablev N.M. & Ivaschenko G.S. (2014). Parallel Immune Algorithm of Short-Term Forecasting Based on Model of Clonal Selection. Radio Electronics, Computer Science, Control, (2), 73-78.

MANUFACTURING OPTIMIZATION USING AN IMMUNOLOGICAL CLONAL SELECTION ALGORITHM

Maksym Torba
Graduate of the Faculty of Computer Science
Kharkiv National University of Radio Electronics

SCIENTIFIC ADVISER:

Grigorij Chetverykov
Professor of Software Engineering Department,
Chief Editor of Scientific Journal «Bionics of Intelligence»
Doctor of Engineering Sciences, Professor
Kharkiv National University of Radio Electronics

UKRAINE

Over the last twenty years, there has been a growing interest in the field of artificial immune systems (AIS). The purpose of the AIS is to use ideas derived from immunology to develop systems capable of performing a wide range of tasks in various fields of research [1].

The CLONALG algorithm, originally developed to solve the problem of pattern recognition, was adapted to solve the problem of optimizing multimodal functions [2][3].

The subject of the study is the use of an artificial immune system to solve the problem of production optimization. An abstract production model was created. Optimization will be to find the parameters of the production model that give the maximum profit. The CLONALG clonal selection algorithm was chosen to model the work of the artificial immune system and adapted to solve the problem. Models of production and artificial immune system were implemented in the software application, which was used in the study.

The task of optimization is to find the maximum profit and is formulated as follows:

$$\text{Income} = \sum_{i=1}^{n_p} \left(p_i * \left(v_i - m_i * m_p\right)\right) - (1 + b_i) * \\ * \sum_{i=1}^{n_w} (w_i * s_i * t_{wi}) - m_r * m_p) - \text{punish} \quad (1)$$

where n_p – number of item types (parameter), $p_i(n_m)$ – number of manufactured i-type items, $v_i(v_{bi}, t_{wi}, t_{bi}, w_q)$ – value of the i-type items, m_i – number of material needed to manufacture the i-type item (parameter), m_p – material price (parameter), n_w – number of types of employees (parameter), w_i – number of i-type employees (parameter), s_i – the salary of i-type employees, b – the bonus for employees (parameter), t_{wi} – real working time of the i-type machine/employee for one product, $m_r(p_i, n_m)$ – material remains, unused material, $p_{i_{min}}$ – minimum number of i-type items that must be manufactured to avoid punishment (parameter), $p_{i_{max}}$ – maximum number of items that can be manufactured, n_m – number of materials in the warehouse at the beginning of the day (parameter), $\text{punish}(p_{un}, p_{num}, v_i)$ – punishment for not produced required item number

A series of tests was carried out for various parameters of the algorithm. The default parameters were: population size = 100, number of iterations = 200, cell selection rate = 20%, cloning rate = 50% and watchdog = 50. Individual parameters are changed to determine their effect on the optimization result. The best solution is the maximum profit value obtained in 100 runs of the algorithm. Table 1 shows the influence of each parameter on the optimization process.

Table 1 – Influence of algorithm parameters on optimization process

Parameter value	Best value	Iteration with best value	Best solution
Influence of the number of iterations			
10	-167,0	10	1680,4
40	1216,45	37	
100	844,96	73	
500	1680,4	91	
Influence of the population size			
10	-795,31	74	1825,85
40	971,91	20	
200	1739,31	55	
500	1825,85	94	
Influence of the cloning coefficient			
0,25	819,73	26	1615,86
0,40	1547,17	113	
0,60	1615,86	172	
0,80	865,72	74	
Influence of the watchdog			
25	-101,07	19	1767,93
40	843,33	20	
60	948,33	59	
80	1767,93	151	
Influence of the selection rate coefficient			
0,05	1178,58	74	1255,00
0,10	558,54	24	
0,40	1255,00	57	
0,50	807,50	46	

Analysis of experimental results

1. Influence of population size

Based on the tests performed, it can be concluded that the population size strongly affects the modelling results. As the population grows, the best possible outcome improves significantly. The larger the population, the more solutions to the problem are compared simultaneously, which, in the absence of changes in other parameters, naturally increases the chances of finding the best solution. In addition, it can be seen that for small groups, the algorithm ended at 200th (default) iteration. This may be because the entire population is "stuck" at a local minimum and cannot get out of it (since it is known from other tests that better solutions exist). To avoid similar situations in the future, the mutation rate must be better matched.

2. Influence of the number of iterations

The number of iterations does not significantly affect the solution. In tests carried out after a certain number of iterations, it was seen that the best value remained unchanged. It is enough that the number of iterations is large enough for the algorithm to get to the point where significant improvement stops.

Thus, it would probably be better if the number of iterations depended on other parameters. The fact that the number of iterations does not directly affect the best value obtained by the algorithm can be confirmed by comparing the test for 40 and 100 iterations. At 40 iterations, the solution “fell” to the best local minimum, and the best result was about 1200, while at 100 iterations, the solution “fell” to the worst local minimum, and after reaching the limit of 800, despite a large number of iterations, they could not improve significantly.

3. Influence of the cloning coefficient

From the tests it follows that the cloning factor to a certain extent affects the quality of the solution. Naturally, the more clones, the more mutations, the greater the likelihood of better solutions than the previous ones. However, the computational complexity also increases significantly. So, the parameter should be chosen optimally in terms of improving the quality of the solution and the execution

time of the algorithm. In addition, as the ratio increases, the quality improvement appears to decrease, suggesting that a corresponding optimum exists.

The best value for a clone factor of 0.8, which is less than the best value for a factor of 0.6, seems to be an accident, not a decrease in the quality of the solution due to an increase in the factor. Such conclusions can be reached based on the analysis of the entire algorithm, where an increase in the coefficient cannot weaken the quality of the best solution.

4. Influence of the watchdog timer

Based on the tests, it can be determined that the effect of the watchdog parameter is small. This parameter affects the result of the algorithm only at small values, when the algorithm fails to correct the solution in several steps, and the watchdog timer is small enough to terminate the algorithm before reaching the maximum value. In many tests, we failed to see a situation in which the solution did not change best for more than 40 iterations, and then suddenly improved significantly. Thus, we can conclude that for this algorithm it makes no sense to set the watchdog parameter value over 40 and only increases the algorithm's running time.

5. Influence of the coefficient of selected cells

It can be seen that changing the parameter has no significant effect on the quality of the best solution. However, this can be also seen from the algorithm itself. Since the algorithm replaces the original cells only with the best solutions, the choice of additional, worse solutions for creating clones, which, apparently, will be omitted anyway, would be unjustified [4].

As a result, research goal was achieved. Tests have shown that, depending on various parameter settings, the algorithm is able to find the best result in a finite number of iterations. Although the algorithm does not always reach the global optimum in each specific run through its method of operation, the obtained values of the objective function are satisfactory.

The influence of such AIS parameters was investigated: population size, number of iterations, cloning coefficient, watchdog timer, coefficient of selected cells.

For a small number of iterations and a small population size, the algorithm is relatively efficient in terms of computational cost, but sometimes it turns out to be worse in the quality of the results obtained. On the other hand, for higher values of these parameters, it can be clearly investigated that the computational cost is higher, however, this leads to better results.

In further studies, it will be advisable to test the CLONALG algorithm with other target functions (for example, minimizing the time spent on work, minimizing the cost of material). It is also worth comparing the results obtained with those of other commonly used optimization methods.

To summarize, the immunological method using the principle of clonal selection proved to be able to solve the production optimization problem and can be successfully used for other optimization problems.

References:

1. Dasgupta, D. (2014). Immunological computation, theory and applications. Cleveland: CRC Press.
2. Li, Z., Zhang, Y. & Tan, H.-Z. (2011). IA-AIS: An Improved Adaptive Artificial Immune System Applied to Complex Optimization Problems. Applied Soft Computing, (11.8), 4692–4700.
3. Rokicki, L. (2019). The Application Of The CLONALG Algorithm In The Process Of Optimal Operation Control Of Hybrid AC/DC Low Voltage Microgrid. E3S Web of Conferences 8. Retrieved from https://www.e3s-conferences.org/articles/e3sconf/pdf/2019/10/e3sconf_pe2019_02011.pdf

clutch and brake pulley) m_2 – reduced mass of parts that move forward; m_3 – cargo weight; C_1 – coefficient of stiffness of the low-speed transmission shaft; C_2 – stiffness of another elastic connection

Move horizontally $x_3=l\varphi$, because $x_3=H/C2$.
The equations of motion of each mass will have the form [1]:

$$\begin{aligned} m_1\ddot{x}_1 &= P-F_1, \\ m_2\ddot{x}_2 &= F_1-F_2-W, \\ m_3\ddot{x}_3 &= F_2. \end{aligned} \tag{3}$$

Efficacy in elastic bonds:

$$\begin{aligned} F_1&=W+C_1(x_1-x_2), \\ F_2&=C_2(x_2-x_3). \end{aligned} \tag{4}$$

Solving the system (3) the S.A. Cossack received an equation for loads:

$$\begin{aligned} F_1&=A_1\cos w_1 t+A_2\cos w_2 t+D_1, \\ F_2&=A_3\cos w_1 t+A_4\cos w_2 t+D_2, \end{aligned} \tag{5}$$

where frequencies are determined by the formula:

$$\omega_{1,2}^2=\frac{1}{2}\left(\frac{C_1}{J_1}+\frac{C_1+C_2}{J_2}+\frac{C_3}{J_3}\right) \tag{6}$$

$$\mp\frac{1}{2}\sqrt{\left(\frac{C_1}{J_1}+\frac{C_1+C_2}{J_2}+\frac{C_3}{J_3}\right)^2-\frac{4C_1C_2}{J_1J_2}\frac{J}{J_3}},$$

where

$$J=J_1+J_2+J_3. \tag{7}$$

References:

[1] Ren, Z., Iwnicki, S. D., Xie, G. A. (2011). A new method for determining wheel–rail multi-point contact. Vehicle System Dynamics, (10). 1533–1551. https://doi.org/ 10.1080/00423114.2010. 539237.

[2] Haniszewski, T. (2017). Modeling the dynamics of cargo lifting process by overhead crane for dynamic overload factor estimation. Journal of Vibroengineering, 19 (1), 75–86. doi: https://doi.org/10.21595/jve.2016.17310

[3] Raksha, S. V., Anofriev, P. G., Bohomaz, V. M., Kuropiatnyk, O. S. (2019). Mathematical and s-models of cargo oscillations during movement of bridge crane. Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu, 2, 108–115. doi: https://doi.org/ 10.29202/nvngu/2019-2/16

[4] Фідровська, Н. М., Слепужніков, Є. Д. (2012). Визначення оптимальних параметрів ходових коліс мостових кранів. Науковий вісник будівництва, (69), 215–222. http://repositsc.nuczu.edu.ua/handle/123456789/7436

[5] Слепужніков, Є. Д. (2015). Визначення динамічних навантажень при пересуванні вантажного візка мостового крана. Машинобудування, (16), 34 – 37. http://repositsc.nuczu.edu.ua/handle/123456789/7458

[6] Fidrovska, N., Slepuzhnikov, E., Perevoznik, I. (2019). A contact problem solution with taking into account shear deformations. Science and Education a New Dimension. Natural and Technical Sciences. VII(23), (193), 80–81. doi.org/10.31174/SEND-NT2019-193VII23-20

[7] Казак, С. А. (1968). Динамика мостовых кранов. (с. 332-333). Москва: Машиностроение.

DOI 10.36074/30.10.2020.v1.25

USE OF ARTIFICIAL IMMUNE SYSTEMS TO SOLVE DATA PROCESSING PROBLEMS

Maksym Torba
Graduate of the Faculty of Computer Science
Kharkiv National University of Radio Electronics

SCIENTIFIC ADVISER:

Grigorij Chetverykov
Professor of Software Engineering Department,
Chief Editor of Scientific Journal «Bionics of Intelligence»
Doctor of Engineering Sciences, Professor
Kharkiv National University of Radio Electronics

UKRAINE

Appearing in the previous century, today artificial intelligence is one of the fastest growing branches of computer science. Many emerging technologies use artificial intelligence.

Artificial intelligence can be based on methods based on biological principles. Artificial neural networks - mathematical models, as well as their hardware or software implementation, which are based on the principles of the network of nerve cells of the body, have undergone significant development.

In addition to neural networks, the interest of computer scientists is attracted by genetic algorithms, evolutionary and molecular calculations, as well as artificial immune systems.

Artificial immune systems are a relatively new paradigm of computational intelligence that can be integrated with other methods. The immune biological system is considered as a source of ideas for solving problems in the field of data processing and analysis, mathematical modeling and information security.

Computational models of the artificial immune system are developed on the basis of the observed properties of the natural immune system and include the most promising in terms of calculations features. One such model is the immune network model [1]. The computational model of the immune network is based on two main propositions. When the antigen enters the body, its recognition by antibodies leads to the activation of the network and an increase in the number of antibodies - the immune network forms an immune response.

There is also a theory that the immune system is a regulated network of molecules and cells that recognize each other even in the absence of antigen. This theory is based on the assumption that different clones of antibodies are not isolated from each other, but maintain communication through interactions between their receptors.

One of the models of immune networks is aiNet - Artificial Immune Network [2]. The purpose of the aiNet algorithm is to build memory cells that recognize and represent the structural organization of data.

AiNet is a edge-weighted graph (Fig. 1), not necessarily connected, consisting of a set of vertices called antibodies and a set of edges connecting pairs of vertices, each of them corresponds to the weight or force of the connection, which is associated with each connected vertex.

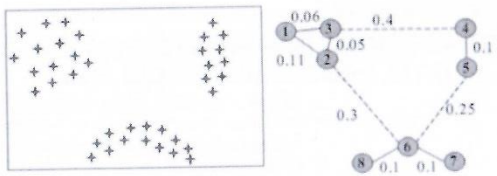


Fig 1. **The result of aiNET**

The method of k-nearest neighbors is an algorithm for automatic classification of objects or regression [3], which can show fairly accurate results.

The kNN algorithm easily decides to which class data must be assigned (Fig. 2), if all neighbors belong to the same class, otherwise the class is set using two methods:

1. Take a simple majority for the right decision. To which class do most of the neighbors belong, to that class will the point belong; That is a pretty simple solution, which shows quite good precision;

2. Do the same, but give the nearest neighbors more weight. This approach implies that the points have meaningful weights according to the subject area.

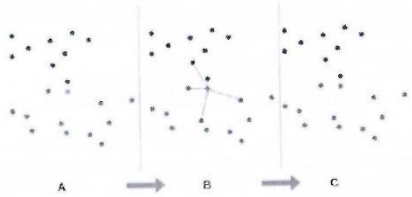


Fig 2. **kNN in action**

However, it is worth considering the case when several points lie at the same minimum distance next to the classified point. For example, the nearest are two points at once. One way to choose a class is to choose a class at random.

As already mentioned, models of the artificial immune system are successfully integrated with other computational models. At this stage, the kNN model and the aiNET artificial immune network model can be combined. To solve the classification problem, we obtain a hybrid model that has taken in the advantages of the two models. Thus, a hybrid model that learns will help to avoid random class setting.

KNN's shortcomings are known to occur when dealing with a large amount of data. Finding k nearby neighbors becomes a time-consuming task. One can work around this issue by using distributed calculations.

One of the trends of recent years is the use of microservices [4]. This approach will help speed up data processing by about as many times as microservices will be running (keep in mind the indirect costs of working with microservices, such as 300ms of network time).

The work of the program is proposed to be organized as follows. The source data is a figure consisting of a set of points. The server part of the application will

accept the request for image processing and decide how to divide it into parts and perform distributed. In parallel with the server, microservices are launched, which listen to the server and wait for a request for clustering. After dividing the image into parts, the server will make a request to each microservice (Figure 3), and will send them their parts of the image for processing. Microservices use a hybrid model kNN + aiNET.

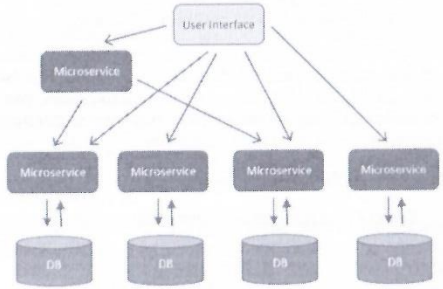


Fig 3. **Organization of an application with microservices**

When all microservices process their parts of the image, the server will collect the result of each microservice and send it to the client.

Conclusions. Thus, an improved approach to data classification using kNN is considered. As a training model, it is proposed to use the principles of artificial immune network – aiNET model, as well as microservices for distributed data processing, which will significantly reduce the time of work.

References:

[1] Dasgupta, D. (2014). Immunological computation, theory and applications. Cleveland: CRC Press.

[2] Watkins, A. (2012). New Classifier Based on Resource Limited Artificial Immune Systems (Vol. 2. pp. 146–151). CEC '12. August 07, 2012. Honolulu, USA.

[3] Muwang, C. & Kuo, C. (2008). Application of Artificial Immune System Approach in MRI Classification. EURASIP Journal on Advances in Signal Processing, (1), 208-212.

[4] Villamizar, M., Garces, O. & Castro, S. (2015). Evaluating the monolithic and the microservice architecture pattern to deploy webapplications in the cloud. September 21-25 2015, 10th Computing Colombian Conference (10CCC). Bogota, USA.

CERTIFICAT DE PARTICIPANT ΛΌΓΟΣ

Certificate provides at least a 0.2 ECTS credits to awarded participants for being involved.

NVGG № 301020-083
du 30.10.2020

Maksym Torba

a pris une participation à la conférence scientifique et pratique internationale

TENDANCES SCIENTIFIQUES DE LA RECHERCHE FONDAMENTALE ET APPLIQUÉE

30 OCTOBRE 2020 • STRASBOURG, RÉPUBLIQUE FRANÇAISE

Euro Science Certificate № 22182 du 04.10.2020 UKRISTEI (Ukraine) Certificate № 450 du 05.10.2020

ORCID Crossref OpenAIRE

Les matériaux du participant à la conférence ont été acceptés et publiés dans la collection de papiers scientifiques ΛΌΓΟΣ.

URL: *https://ojs.ukrlogos.in.ua/index.php/logos/issue/view/30.10.2020*

EUROPEAN SCIENTIFIC PLATFORM

Chef de l'organisation publique
Plateforme scientifique européenne
HOLDENBLAT MARIIA

EUROPEAN SCIENTIFIC PLATFORM
INTERNATIONAL NON-GOVERNMENTAL ORGANIZATION

ДОДАТОК В

Слайди презентації

Дослідження моделі штучної імунної системи для вирішення задачі оптимізації виробництва

Виконав:
ст.гр. ПЗСм-19-1
Торба Максим Олегович

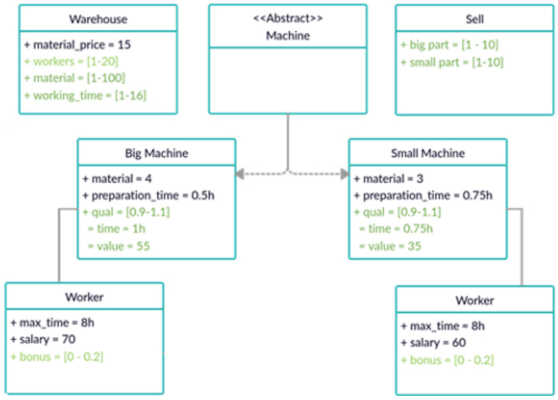
Керівник:
проф. каф.ПІ, д.т.н.,
проф. Четвериков Григорій Григорович

Постановка завдання

- проаналізувати предметну область;
- побудувати математичну модель виробництва і поставити задачу оптимізації;
- побудувати математичну модель штучної імунної системи, в основі якої лежатимуть принципи клонального відбору; адаптувати модель для вирішення задачі оптимізації змодельованого виробництва;
- розробити програмний додаток, який допоможе дослідити вплив параметрів побудованої ШІС на оптимізацію виробництва;
- проаналізувати отримані результати.

3

Модель виробництва



- Виробництво виготовляє деякі деталі
- Зміна – 8 годин, максимум працює 16 годин на день
- Два види верстатів: великі і малі
- З кожним верстатом пов’язаний один оператор
- Є замовлення з необхідною кількістю деталей
- Два види продукції: великі і дрібні деталі
- Бонуси, штрафи, прискорення
- Прибуток – сумарна вартість усіх виготовлених деталей за вирахуванням зарплати операторам і, при наявності, штрафу

4

Функція цілі

Задача оптимізації полягає в пошуку максимального прибутку і формулюється наступним чином:

$$\text{Income} = \sum_{i=1}^{n_p} \left(p_i * \left(v_i - m_i * m_p\right)\right) - (1 + b_i) * * \sum_{i=1}^{n_w} (w_i * s_i * t_{wi}) - m_r * m_p) - \text{punish}$$

5

Штраф за невиконання необхідної кількості деталей описується формулою:

$$punish = p_{un} * \sum_{i=1}^{n_p}(p_{num_i}) * v_i, \quad (2)$$

$$p_{num_i} = \begin{cases} 0 & \text{if } p_{i_{min}} - p_i \leq 0 \\ p_{i_{min}} - p_i & \text{if } p_{i_{min}} - p_i > 0 \end{cases} \quad (3)$$

де p_{un} – штрафний коефіцієнт, $p_{num_i}(p_{i_{min}}, p_i)$ – кількість елементів і-го типу, для яких буде нарахований штраф, p_i – вироблена кількість деталей і-го типу

Вартість однієї деталі обчислюється за формулою:

$$v_i = v_{bi} * (\frac{t_{wi}}{t_{bi}} * w_q * 100\%) \quad (8)$$

де t_{bi} – базовий час роботи машини і-го типу над одним продуктом (параметр), v_{bi} – базова вартість продукту, виробленого машиною і-го типу (параметр), w_q – якість працівника (залежить від бонусу)

6

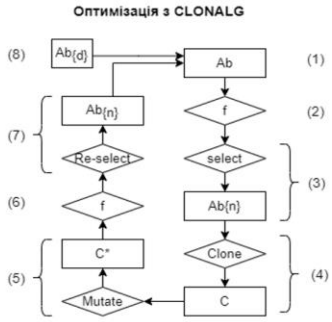
Значення параметрів моделі виробництва, використані для проведення експерименту

Параметр	Позначається як	Значення
Кількість матеріалу	n_m	[x − 100x]
Вартість матеріалу	m_p	4
Час праці	t_f	[1 − 16]
Мінімальна кількість великих деталей	$p_{0_{min}}$	[0 − 10]
Мінімальна кількість дрібних деталей	$p_{i_{min}}$	[0 − 10]
Зарплата оператора великого верстату	s_0	19
Кількість сировини на велику деталь	m_0	6
Час на підготовку великого верстату	t_{p_o}	1h 45 min
Вартість великої деталі	v_{b_o}	70
Час для виготовлення верстатом великої деталі	t_{b_o}	1h
Кількість великих верстатів	c_0	[0 − 30]

Зарплата оператора малого верстата	s_1	17
Кількість сировини на малу деталь	m_1	4
Час на підготовку малого верстату	t_{p1}	1h 25 min
Вартість малої деталі	v_{b1}	50
Час для виготовлення малої деталі	t_{b1}	1h 25 min
Кількість малих верстатів	c_1	[0 − 30]
Максимальний час праці працівника	t_w	8h
Премія працівника	b	[0.0 − 0.5]
Коефіцієнт штрафу	p_{un}	[0 − 1]

7

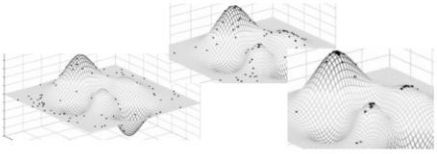
Алгоритм CLONALG



Цикл для кожного покоління:

1. **Представлення цільової функції популяції** антитіл
2. **Обчислення афінності** антитіл
3. **Відбір** антитіл
4. **Клонування** анитіл
5. **Мутація** клонів
7. **Обчислення афінності** клонів
8. **Генерація d випадкових антитіл** для підтримки різноманітності в популяції **Ab**

Переміщення антитіл по поверхні функції з кожними наступними ітераціями

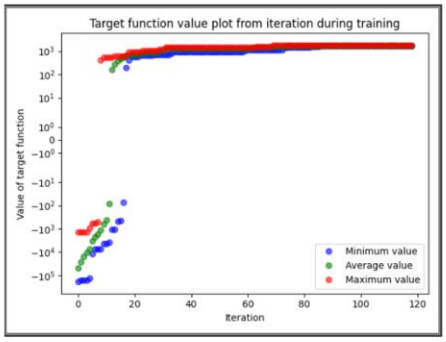


8

Програмна реалізація

Інструменти розробки:
- Python
- Matplotlob.pyplot
- Середовище PyCharm

Реалізовано:
- Алгоритм Clonalg
- Алгоритм роботи виробництва
- Тестування моделі
- Візуалізація даних




```
Iteration: 174  Best value: 1756.87 Worst value: 1703.69    Average value: 1730.28
Iteration: 175  Best value: 1756.87 Worst value: 1703.69    Average value: 1730.28
Iteration: 176  Best value: 1756.87 Worst value: 1703.69    Average value: 1730.28
Iteration: 177  Best value: 1756.87 Worst value: 1703.69    Average value: 1730.28
Iteration: 178  Best value: 1756.87 Worst value: 1703.69    Average value: 1730.28
Iteration: 179  Best value: 1756.87 Worst value: 1703.69    Average value: 1730.28
Max value 1756.87. No progress since 50 iterations
{'object': <Models.factory.Factory object at 0x0000022B58AEE130>,
 'specifications': {'big_machines': 12,
                    'bonus': 0.2799999999999999,
                    'haste': 0,
                    'material': 943,
                    'small_machines': 9,
                    'time': 16},
 'value': 1756.87}

Process finished with exit code 0
```

9

Опис експерименту

Знайти максимальне значення прибутку при заданих параметрах, отримане за 100 запусків алгоритму. Окремі параметри змінюються для визначення їх впливу на результат оптимізації.

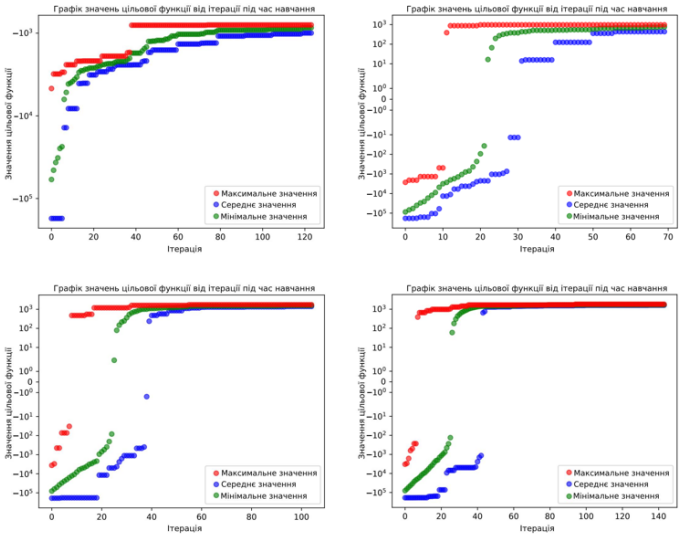
Параметрами за замовчуванням були:

- розмір популяції = 100,
- кількість ітерацій = 200,
- коефіцієнт відібраних клітин = 20%,
- коефіцієнт клонування = 50%
- watchdog = 50.

10

Вплив розміру популяції

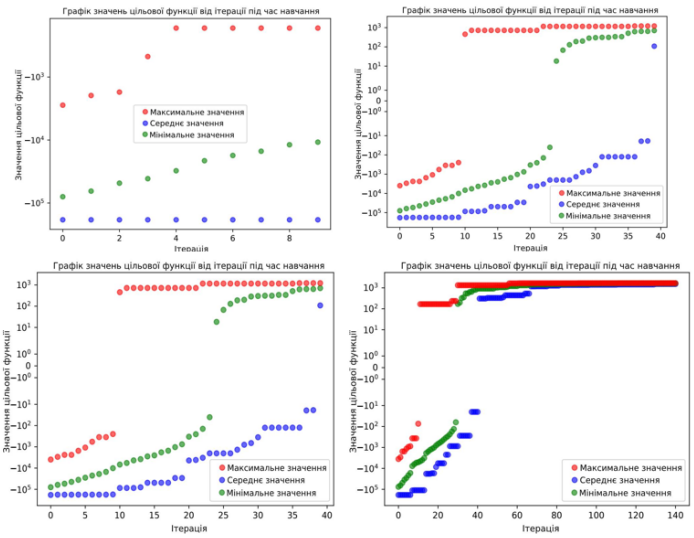
Розмір популяції	Найкраще значення	Ітерація з найкращим значенням	Найкращий розв'язок
10	1748.8	74	1931.39
40	1875.88	20	
200	1917.62	55	
500	1931.39	94	



11

Вплив кількості ітерацій

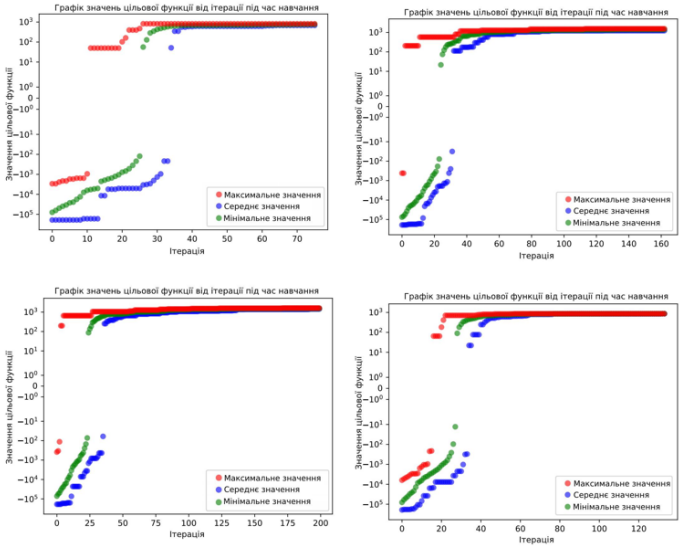
Кількість ітерацій	Найкраще значення	Ітерація з найкращим значенням	Найкращий розв'язок
10	1632.2	10	1907.22
40	1871.88	37	
100	1904.13	73	
500	1907.22	91	



12

Вплив коефіцієнту клонування

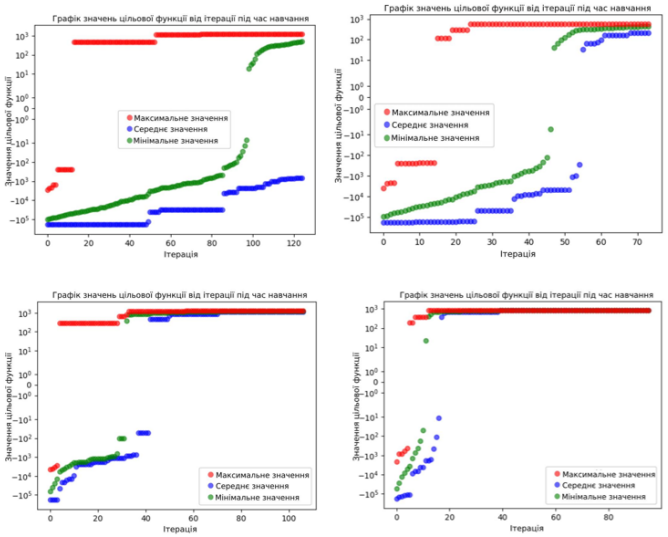
Коефіцієнт клонування	Графік	Найкраще значення	Ітерація з найкращим значенням	Найкращий розв'язок
0,25	Графік 10	1819,73	26	1915,86
0,40	Графік 11	1847,17	113	
0,60	Графік 12	1915,86	172	
0,80	Графік 13	1904,72	74	



Вплив коефіцієнту вибраних клітин

13

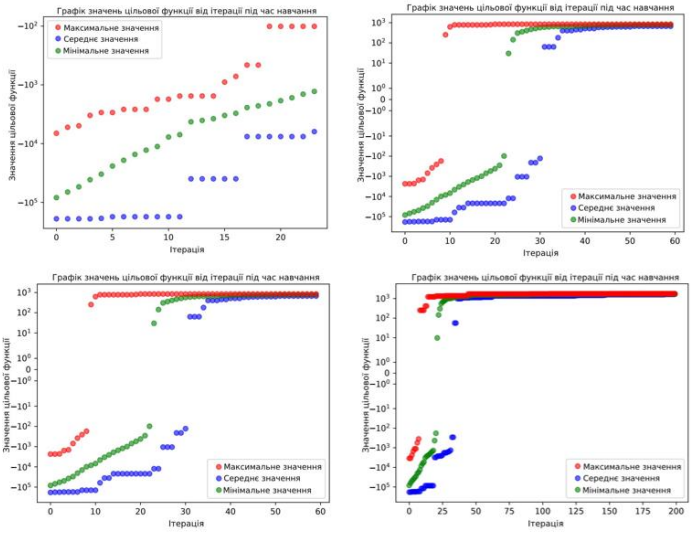
Коефіцієнт вибраних клітин	Найкраще значення	Ітерація з найкращим значенням	Найкращий розв'язок
0,05	1899,58	74	1906,54
0,10	1906,54	24	
0,40	1905,00	57	
0,50	1899,50	46	



Вплив контролюючого таймера

14

Таймер	Найкраще значення	Ітерація з найкращим значенням	Найкращий розв'язок
25	1870.86	19	1915.15
40	1915.15	158	
60	1908.33	199	
80	1897,93	151	



Аналіз результатів

15

Вплив розміру популяції

- Сильний вплив. Зі збільшенням розміру популяції значно покращується найкращий можливий результат.
- Для невеликих груп алгоритм закінчувався до 200-ї (значення за замовчуванням) ітерації. Вся популяція "застряє" на локальному мінімумі і не може з нього вийти (з інших тестів відомо, що кращі рішення існують).
- Щоб уникнути подібних ситуацій у майбутньому, коефіцієнт мутації повинен бути краще підібраний.

Вплив кількості ітерацій

- Суттєво не впливає на рішення. Після певної кількості ітерацій найкраще значення залишається незмінним.
- Достатньо, щоб кількість ітерацій була досить великою, щоб алгоритм дійшов до точки, коли значне поліпшення зупиняється.

Ймовірно, було б краще, якби кількість ітерацій залежала від інших параметрів. Те, що кількість ітерацій безпосередньо не впливає найкраще значення, отримане за допомогою алгоритму, можна підтвердити порівнянням тесту на 40 і 100 ітерацій. При 40 ітераціях рішення «впали» в кращий локальний мінімум, і найкращий результат був близько 1200, тоді як при 100 ітераціях рішення "впали" в гірший локальний мінімум, і після досягнення межі 800, незважаючи на велику кількість ітерацій, вони не змогли суттєво покращитися.

Аналіз результатів

16

Вплив коефіцієнту клонування

- Певною мірою впливає на якість рішення. Більше клонів – обчислювальна складність суттєво зростає.
- Із збільшенням коефіцієнту покращення якості, схоже, знижується, що дозволяє припустити, що відповідний оптимум існує.
- Найкраще значення для коефіцієнту клонування 0.8, яке менше ніж для коефіцієнту 0.6, здається випадковістю, а не зниженням якості рішення внаслідок зростання коефіцієнту. До таких висновків можна дійти, спираючись на аналіз роботи всього алгоритму, де збільшення коефіцієнта не може послабити якість найкращого рішення.

Вплив змін коефіцієнта обраних клітин

- Не має суттєвого впливу на якість найкращого рішення. Випливає з алгоритму

Вплив зміни сторожового таймера

- Вплив малий. Ефективний лише при малих значеннях, коли алгоритму не вдається виправити рішення за кілька кроків, а сторожовий таймер досить малий, щоб припинити алгоритм до досягнення максимуму значення.
- Не вдалося побачити ситуацію, в якій найкраще рішення не змінювалось більше 40 ітерацій, а потім раптом значно покращилось

17

Висновки

- Досліджено предметну область ШІС, зокрема в задачах оптимізації
- Створено модель виробництва і сформульовано задачу оптимізації – підвищення прибутку.
- Алгоритм CLONALG адаптовано для вирішення поставленої задачі.
- Створено програмний додаток для проведення експерименту
- Алгоритм виявився здатним вирішити поставлену задачу оптимізації виробництва та може бути успішно використаний і для інших задач оптимізації.
- В подальших дослідженнях буде доцільно протестувати алгоритм CLONALG з іншими цільовими функціями (наприклад, мінімізації витраченого часу на роботу, мінімізації витрат на матеріал). Також варто порівняти отримані результати з результатами інших загальновживаних методів оптимізації.

ДОДАТОК Г

Електронні матеріали