



Известия высших учебных заведений. Прикладная нелинейная динамика. 2021. Т. 29, № 3
Izvestiya Vysshikh Uchebnykh Zavedeniy. Applied Nonlinear Dynamics. 2021;29(3)

Article

DOI: 10.18500/0869-6632-2021-29-3-376-385

Boundaries of computational complexity and optimal cluster's quantity for controlled swarm in non-cooperative games

O. M. Kiselev

Innopolis University, Russia
Institute of Mathematics with Computer Center UFRC RAS, Ufa, Russia
E-mail: ok@ufanet.ru

Received 27.10.2020, accepted 23.12.2020, published 31.05.2021

Abstract. The purpose of the work is to determine the relationship between the computational complexity of controlling a swarm of particles and the available computational resources for choosing the optimal control strategy. We find formulas for the relationship between the available computational complexity, the number of clusters in the swarm, the number of interacting players, and the depth of calculations to search the suboptimal control. *Methods.* To find the optimal control, the method of maximizing the target objective function is used. The computational complexity of the objective function is determined for suboptimal control on the grid for the maximum number of particle swarms and the minimum of the grid size. To study the swarm dynamics in the configuration space we investigate the properties of the convex hull of the swarm using elements of the particle diffusion theory. *Results.* The objective function of controlling a swarm of particles in the conditions of interaction with stationary objects and in the presence of competitive swarms is constructed. It is shown that the swarm of particles spread in the configuration space due to instability and mixing in the general situation. Formulas for the maximum possible size of clusters and the number of particles in clusters, the depth of calculation of control steps and the detail of the grid are obtained. The connection between the dynamics of the swarm of controlled particles and the theory of Smolukhowski's coagulation in colloidal solutions is shown. *Conclusion.* Constraints on the computational complexity of the control lead to a restriction on the size of the iteration grid for finding the minimax and to a restriction on the number of swarm clusters for which the optimal strategy can be chosen. The clustering capability of the swarm leads to the fact that the product of the number of positions in the grid in the optimization of the number of clusters in a swarm depth calculation steps should be no more than the order of the logarithm of acceptable computational complexity.

Keywords: Swarm of clusters, optimal control, computational complexity.

Acknowledgements. I thank Prof. V. Y. Novokshenov for pointing out a close analogy of clustering objects in a swarm with Smolukhowsky's theory of coagulation.

For citation: Kiselev OM. Boundaries of computational complexity and optimal cluster's quantity for controlled swarm in non-cooperative games. Izvestiya VUZ. Applied Nonlinear Dynamics. 2021;29(3):376–385.

DOI: 10.18500/0869-6632-2021-29-3-376-385

This is an open access article distributed under the terms of Creative Commons Attribution License (CC-BY 4.0).

Ограничения вычислительной сложности и оптимальные размеры кластера для управления роем в некооперативных играх

О. М. Киселев

Университет Иннополис, Россия
Институт математики с вычислительным центром УФИЦ РАН, Уфа, Россия
E-mail: ok@ufanet.ru

Поступила в редакцию 27.10.2020, принята к публикации 23.12.2020, опубликована 31.05.2021

Аннотация. Цель работы – определить зависимость между вычислительной сложностью управления роем частиц и доступными вычислительными ресурсами для выбора оптимальной стратегии управления. Получить формулы связи между доступной вычислительной сложностью, количеством кластеров в рое, числом взаимодействующих игроков и глубиной вычислений при поиске субоптимального управления. **Методы.** Для поиска оптимального управления используется метод максимизации целевой функции. Вычислительная сложность целевой функции определяется для субоптимального управления на решетке для максимального числа роев частиц и минимальном размере решетки. Для исследования динамики роя в конфигурационном пространстве исследуются свойства выпуклой оболочки роя с помощью элементов теории диффузии частиц. **Результаты.** Построена целевая функция управления роем частиц в условиях взаимодействия со стационарными объектами и в присутствии конкурентных роев. Показано, что в ситуации общего положения рой частиц размывается в конфигурационном пространстве из-за неустойчивости и перемешивания. Получены формулы для максимально возможного размера кластеров и числа частиц в кластерах, связывающие глубину просчета шагов управления и детализацию решетки. Показана связь между динамикой роя управляемых частиц и теорией коагуляции Смолуховского в коллоидных растворах. **Заключение.** Ограничения на вычислительную сложность управления приводят к ограничению размеров решетки перебора для поиска минимакса и к ограничению количества кластеров роя, для которых возможен выбор оптимальной стратегии. Возможность кластеризации роя приводит к тому, что произведение числа узлов решетки оптимизации, количества кластеров в рое глубины вычислений по шагам должно быть не более, чем порядок логарифма от допустимой вычислительной сложности.

Ключевые слова: рой объектов, оптимальное управление, вычислительная сложность.

Благодарности. Я благодарю проф. В. Ю. Новокшенова за указание на близкую аналогию кластеризации объектов в рое с теорией коагуляции Смолуховского.

Для цитирования: Киселев О. М. Ограничения вычислительной сложности и оптимальные размеры кластера для управления роем в некооперативных играх // Известия вузов. ПНД. 2021. Т. 29, № 3. С. 376–385.
DOI: 10.18500/0869-6632-2021-29-3-376-385

Статья опубликована на условиях лицензии Creative Commons Attribution License (CC-BY 4.0).

Introduction

The problem of optimizing swarm control involves studying the behaviour of a large number of similar objects. These objects are located at different points in the phase space. Therefore, optimization engages in solving the problem of finding the best strategy for each swarm object or for individual swarm clusters of similar objects. Such problems require a lot of computing power. In non-cooperative real-time games, first, computing resources are significantly limited in terms of time and computational speed, and second, the size of allocated memory is also limited.

General questions about the behaviours of groups of mobile robots were also considered in early works. There are many references to the use of this approach in controlling a group of mobile robots [1, 2]. Such problems are characterized by nonlinearity, higher dimension of the search space and complex topology [3]. More work one can see, for example, in the reviews [4] and [5].

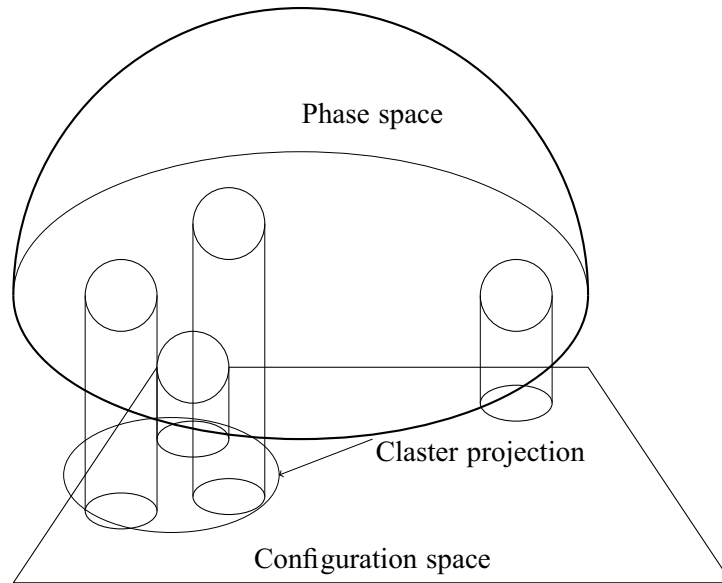


Fig. 1. The phase space is the joining up the configuration space \mathbf{X} and the additional space of internal parameters \mathbf{Y} . The phase space is the following product $\mathbf{X} \times \mathbf{Y}$

It is clear that the use of large groups leads to resource constraints and the problem of their optimal use. For a more up-to-date view of tasks for robot swarms, see, for example, the review [6]. Some preliminary considerations related to computational complexity for swarm control and the present work are outlined in the report [7]. Here we obtain a formula for the optimal number of swarm clusters under a given constraint of computational complexity.

Formally the problem is largely based on the competitions of programmed bots for the game Almost Agar IO, which was organized by Mail.ru group [8] on the platform <http://aicups.ru>. Questions about building a strategy can be found in the article of the winner of the competition, A. Dichkovsky [9].

1. The formal side of the problem

Consider a swarm of N objects of the same type. Let the state of each object from the swarm be characterized by a set of *configuration parameters* $\mathbf{X} \in \mathcal{X} \subset \mathbb{F}^n$, *internal parameters* $\mathbf{Y} \in \mathcal{Y} \subset \{\mathbb{N}^l \times \mathbb{F}^m\}$ and *control vector* $\mathbf{u} \in \mathcal{U}$. Here \mathbb{F} is the notation of a lattice of numbers of **floating point**, \mathcal{U} is the set of valid values for the control vector \mathbf{u} . Assume that the configuration space is metric.

We assume that objects located at a distance of less than R in the configuration space for more than k control cycles form a *cluster*. A cluster is a composite object in space $\{\mathcal{X}, \mathcal{Y}\}$. The number of single objects k combined in a cluster is one of the components of the set of internal parameters $\mathbf{Y}^{(1)} = k$.

Information about the environment in the configuration space is visible for a cluster at a distance of no more than $\rho = \rho(k)$, where k is the number of individual objects in the cluster. For a single object, it is obvious: $\mathbf{Y}^{(1)} = 1$.

Let the control be discrete. Then the sequence of steps can be considered as a mapping of the set $\{\mathcal{X}, \mathcal{Y}\}$ to itself:

$$\{\mathbf{X}_{i+1}, \mathbf{Y}_{i+1}\} = \mathbf{F}(\mathbf{X}_i, \mathbf{Y}_i, \mathbf{u}_i), \quad i \in \overline{1, \dots, I}, \quad I \in \mathbb{N}.$$

The computational complexity of this map is denoted by $N(\mathbf{F})$.

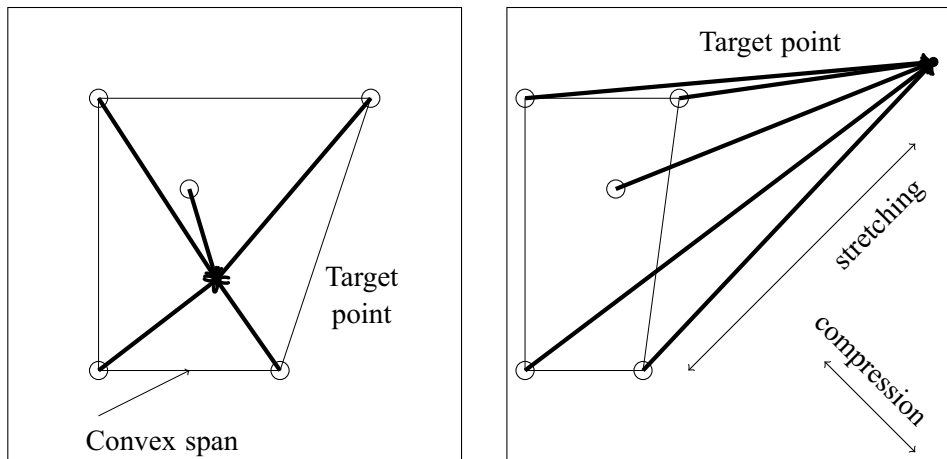


Fig. 2. On the left picture the target point lies in the convex span of the swarm. On the right one the target point lies out of the span

Let's assume that points in the game are awarded for reaching a state from some *prize set* in the configuration space $\mathcal{B} \in \mathcal{X}$. Then the goal function of the game is to get the maximum number of points for I steps.

Let there be only one swarm of objects. When any of the swarm objects reaches a state from the set \mathcal{B} , a point is awarded to the all swarm.

For each of the swarm objects, the game in this setting is a non-zero-sum cooperative game. In other words, when any of the swarm objects are reached, points are awarded to the all swarm.

Let the configuration space contains κ of competing swarms. We will denote a controlled swarm as a swarm $\mathbf{1}$. The positions of clusters of competing swarms are known only for clusters located at a distance of no more than $\rho(k)$ from each of the swarm clusters $\mathbf{1}$.

According to the game's condition, the collision of clusters of competing swarms leads to the absorption of a cluster that combines a smaller number of objects.

In this case, there is a *penalty set* $\mathcal{P}_i \in \mathcal{X}$ in the configuration space. The set $\mathcal{P}_i \in \mathcal{X}$ can change its position and characteristics depending on the control step i . This set consists of areas of configuration space that are occupied by clusters of competing swarms that are larger than the $\mathbf{1}$ swarm clusters that can fall into this area.

In addition to the penalty set, there is an additional prize set in the configuration space with competing swarms $\tilde{\mathcal{B}}_i \in \mathcal{X}$. This set is occupied by clusters of competing swarms that are smaller than the swarm clusters $\mathbf{1}$ that can fall into $\tilde{\mathcal{B}}_i$.

2. Swarm dynamics in the configuration space

As a control, we will select a point in the configuration space to which all the objects that make up the swarm tend. Also, assume that moving in the target direction depends on the size of the cluster. In one step, clusters with a smaller dimension move in the target direction further than the clusters of the larger dimension.

Let V_i be the speed of the cluster in the configuration space, m_i be the mass of the cluster, and P_i be the position of the cluster with radius $r_i = r(m_i)$ in configuration space, F_j – coordinates of free resources for the swarm in space.

In the simplest case, in the presence of only one swarm, when the swarm clusters are not divided into smaller ones and are not combined into large clusters, then the movement of each cluster is determined by the system:

$$\begin{aligned} |P_i - u| \neq 0, \quad f_i &= \frac{P_i - u}{|P_i - u|}, \\ \Delta P_i &= V_i dt, \quad m_i \Delta V_i = f_i dt, \quad u \in \mathcal{U}. \end{aligned} \quad (1)$$

Let S_i be the area of the configuration space that is swept when the i -th cluster moves, then the change in the cluster mass depends on how many free resources are in the swept area of space:

$$\Delta m_i = \kappa \sum_j \delta_{ij}(F_j \in S_i), \quad \kappa \in \mathbb{R}, \quad \kappa > 0. \quad (2)$$

If a cluster consists of n objects ($n > 1$), then it can be divided into two smaller clusters in a given proportion, consisting of clusters of n_1 and n_2 objects. If the cluster is divided, for example, $n_1 + n_2 = n$ and $n_1 = \lfloor n/2 \rfloor$. When dividing, clusters change their internal parameters \mathcal{Y} according to a well-known rule:

$$\mathcal{Y}_{i+1}|_{\mathcal{Y}^{(1)}=n_1} = G_1(\mathcal{Y}_i), \quad \mathcal{Y}_{i+1}|_{\mathcal{Y}^{(1)}=n_2} = G_2(\mathcal{Y}_i).$$

In the dynamics formulas (1) and (2), two clusters must be considered instead of one cluster. Let's denote them i' and i'' . For each cluster obtained after division, the parameters are uniquely determined based on the general rules of the game:

$$\begin{aligned} V_{i'} &= V_{i'}(V_i, m_i), \quad V_{i''} = V_{i''}(V_i, m_i), \\ m_{i'} &= m_{i'}(V_i, m_i), \quad m_{i''} = m_{i''}(V_i, m_i). \end{aligned}$$

The reverse case is when k swarm clusters with numbers $\{i_1, \dots, i_k\}$ are combined into a larger cluster i_0 . In this case, the speed, mass, and size of the new cluster are determined from the game rules:

$$V_{i_0} = V_{i_0}(V_{i_1}, \dots, V_{i_k}), \quad m_{i_0} = m_{i_0}(m_{i_1}, \dots, m_{i_k}).$$

If the target point is outside the convex hull of the swarm in the configuration space, then the mapping is compressive in the direction orthogonal to the direction from the center of mass of the swarm to the target point in the configuration space. In the direction from the center of mass of the swarm to the target point, the swarm is stretched due to different mappings for clusters of different dimensions that make up the swarm.

Successive changes to the target point lead to *mixing* in the configuration space and an increase in the volume of the configuration space located inside the convex hull of the swarm.

If the target point is inside the convex hull of the swarm, then the diameter of the convex hull of the swarm decreases over the game step. When compressing over l steps at $l > k$, the size of the clusters that make up the swarm may increase, and the number of individual swarm objects may decrease accordingly.

In a general situation, the objects of the swarm are heterogeneous. At any given time, a swarm may contain clusters consisting of a different number of unique swarm objects. Let the swarm be a union of M clusters $\mathfrak{R} = \{\cup_{i=1}^M \mathcal{R}^{(k_i)}\}$, where $\sum_i k_i = N$. The speed of movement of the \mathcal{V}_k cluster in the configuration space depends on the size of the cluster.

We estimate the growth of the volume of the configuration space inside the convex hull of the swarm in J consecutive steps with an external target point. Changes in the volume of the configuration space inside the convex hull are estimated from above by

$$\Delta\mathcal{V} = \kappa_1(V_{k_m} - V_{k_M}),$$

where k_M is the size of the maximum cluster swarm, and k_m is the size of the minimum cluster swarm.

For a target point inside the convex hull of a swarm, a lower bound for changing the volume inside the convex hull of a swarm in configuration space:

$$\Delta\mathcal{V} = -\kappa_0 V_{k_M}.$$

Assume that at the current moment t , the volume of the configuration space inside the convex hull of the swarm is S_t . Let the probability of declaring a target point inside a convex hull be inversely proportional to the volume of this hull. Then a stable dynamic equilibrium for the volume of the convex hull of the swarm is achieved under the condition

$$\Delta\mathcal{V}_{t_{i+1}} = \Delta\mathcal{V}_{t_i},$$

or

$$\kappa_1 V_{k_m} = (\kappa_1 + \kappa_0) V_{k_M}.$$

To calculate the possible clustering rate, we can use an analogy with the coagulation theory from the fast coagulation of Smoluhowski [12] when the swarm cluster sizes are saturated for optimization. Indeed, although swarm is generally under control, however, each individual swarm cluster shows elements of chaotic dynamics due to successive extensions and contractions of the swarm's convex hull. In this case, clustering is caused by the "natural" process of transition to the maximum available computational complexity of management.

Calculate the speed of the swarm as a whole:

$$V = \frac{1}{N} \sum_{i=1}^N V_i.$$

The average speed of the clusters in a swarm:

$$v = \frac{1}{N} \sum_i (V_k - V).$$

Take the velocity as a unit of time, then the diffusion coefficient:

$$D = \frac{1}{2} |v|^2.$$

Then according to the clustering theory [12] the rate of cluster formation:

$$S \approx 4\pi R D \left(\frac{N}{\text{Mes}(\mathcal{V})} \right)^2.$$

Here R is the average cluster radius.

3. Suboptimal control

Control is defined by the condition:

$$\sum_i \Delta m_i \rightarrow \max.$$

We take as a suboptimal control an algorithm that computes the swarm objective function $\mathbf{1}$ in q control steps.

Let's denote the number of swarm bonus points $\mathbf{1}$ received at the i -th step: b_i , and the number of penalty points at the i -th step: p_i . Then the objective function is:

$$H_{i,q} = \sum_{k=i}^{i+q} (b_k - p_k).$$

4. Estimation of computational complexity

Consider the case without competitive swarms. Let's denote the measure of the set of admissible controls $p = \text{Mes}(\mathcal{U})$. The computational complexity of the objective function for a single swarm object:

$$N(H_{pq}) = O(N^q(\mathbf{F})) \equiv O(n^q).$$

Here $n = N(\mathbf{F})$.

For a complete search of all possible strategies at one step, the complexity $O(np)$ is obtained. To account for strategies containing q control steps, the complexity increases in a power-law fashion:

$$C = O((np)^q).$$

Part of the components of the control vector \mathbf{u}_k lies in the set \mathcal{X} . This set contains n component of \mathbb{F}^n . Although the set \mathbb{F} is finite in itself, however, it is too large to completely recalculate all possible variants for \mathbf{u}_k .

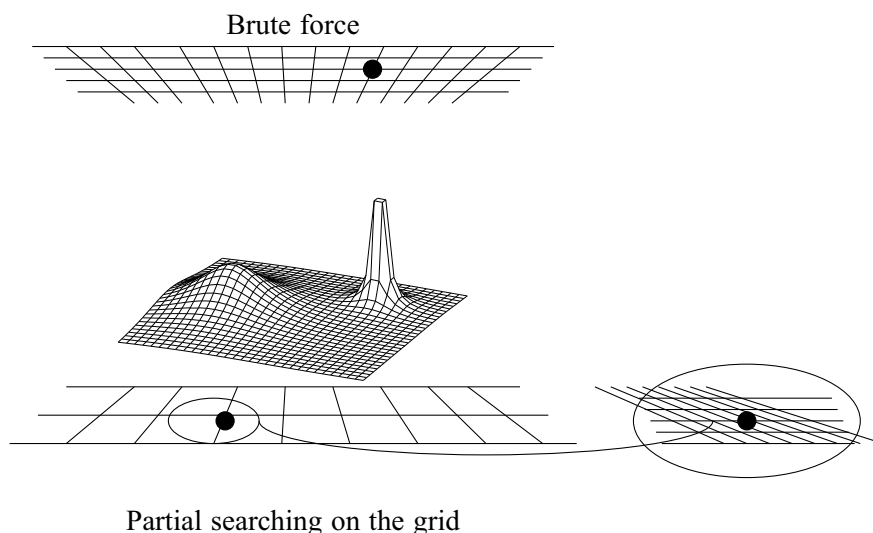


Fig. 3. The projection on the brute force grid is shown on the top. The partial search grid is shown on the bottom. Such approach leads to suboptimal position

The number of objects in the swarm is also assumed to be very large for a complete recalculation of options for all swarm objects.

Therefore, instead of optimizing with a full search for the swarm and control, we will accept a partial search. To do this, consider a sparse grid in phase space for swarm objects and for control parameters. On this rare grid, the minimum is determined. Then, in a ball centered at the point of the local minimum and with a radius equal to the grid step of the previous iteration, a new grid with a smaller step is arranged, and again a search is performed to find the minimum.

Let's denote the number of nodes in the grid k of the level p_k . Then the iterative search process has the computational complexity

$$C_{kr} = O((np_k)^{rq}),$$

where r is the number of localization iterations.

If there are competing swarms, then the computational complexity of the objective function for a single swarm object **1** increases due to taking into account the visible objects of the competing swarms whose convex hull contains the specified swarm object **1**. Constructing a convex hull using the sequential application of localization meshes is similar to the process described above. The complexity can be estimated from above:

$$c_{kr} = O((snp_k)^{rq}),$$

where s is the number of competitive swarms.

Finally, for a non-cooperative game with s swarms, the complexity estimate for finding suboptimal control over a r times-shrinking grid with p_k nodes in q steps is:

$$C = O((sn^2p_k^2)^{rq}).$$

If we allow dividing clusters into two clusters per turn and combining clusters of two clusters per turn, then without taking into account competitive swarms:

$$C = O(((np)!)^{rq}).$$

Taking into account competitive swarms and the division and merging of their clusters:

$$C = O((s((np)^2)!)^{rq}).$$

5. Restrictions on the control step

There are restrictions on the computing resources used. These are restrictions on the memory used, the limited operating frequency of the processor, and the processor time allocated for calculating the control action.

All of these constraints are eventually recalculated to the computational complexity constraint when choosing a control. In a general situation, the growth of the swarm leads to an increase in the computational complexity of control. There are two ways to limit the increase in control complexity.

1. By combining individual swarm objects into clusters, the dynamics of the swarm as a set of cluster movements deteriorates.
2. By performing calculations on an increasingly small scale grid, a control efficiency is deteriorating.

Let N be the number of objects in the cloud. The change in the number of objects is determined first by the measure of the configuration space bounded by the convex hull of the swarm.

The larger the measure of the configuration space, the more opportunities there are to use resources located inside the convex hull. However, if the swarm elements are scattered across the configuration space, then the management efficiency decreases.

The higher the computational complexity of the control leads to the higher the possibility of swarm growth. However, if the threshold value is exceeded, the computational complexity is too high, and there is a delay in calculations, which leads to losses in the swarm size.

- The machine time for calculation per control step is limited.
- The amount of memory for the control program is limited.

Constraints are eventually recalculated to the computational complexity constraint when selecting a control.

There are two ways to limit the increase in control complexity.

1. By combining individual swarm objects into clusters, the dynamics of the swarm as a set of cluster movements deteriorates.
2. By performing calculations on an increasingly small scale grid, control efficiency is deteriorating.

Let N be the maximum available computational complexity. Then for swarms without clustering:

$$N \gg (sn^2p_k^2)^{rq}, \quad \frac{1}{s} \sqrt[2rq]{N} \gg np.$$

For swarms with regards to clustering:

$$N \gg (s((np)^2)!)^{rq}, \quad \frac{1}{s} \sqrt[2rq]{N} \gg (n^2p^2)!.$$

Let $np = z$, then for large z , you can use the well-known Stirling's formula for the asymptotics of the factorial (see e.g. [11]):

$$\frac{1}{s} \sqrt[2rq]{N} \gg z \frac{z^{2z}}{e^{2z}}$$

or

$$\frac{1}{rq} \log(N) \gg (1 + 2z) \log(z) - 2z, \quad \log(N) \gg rqz \log(z).$$

In the end, we get:

$$\log(N) \gg rqp \log(pn). \tag{3}$$

6. Optimal strategy

Let's number all our moves with the index i . After calculations on the lattice, we get the matrix of outcomes for the objective function

$$H_{i,q} = \sum_{k=i}^{i+q} (b_k - p_k).$$

- Optimal strategy – choose the control that achieves the maximum of the minimum values of the function $H_{i,q}$, [10].
- Greedy strategy – choose the control that reaches the maximum of the target function.
- Maximum probability strategy – choose a control for which the sum of the values of the objective function $H_{i,q}$ for all competitors' responses is maximal.

Conclusion

Restrictions on the computational complexity of the control lead to a restriction on the size of the search grid for minimax and to a restriction on the number of swarm clusters for which the optimal strategy can be selected. The possibility of clustering the swarm leads to the fact that the product of the number of nodes in the optimization grid, the number of clusters in the swarm of the depth of calculations in steps should be no more than the order of the logarithm of the allowable computational complexity.

References

1. Teruel E, Aragues R, López-Nicolás G. A distributed robot swarm control for dynamic region coverage. *Robotics and Autonomous Systems*. 2019;119:51–63. DOI: 10.1016/j.robot.2019.06.002.
2. Hu J, Lanzon A. An innovative tri-rotor drone and associated distributed aerial drone swarm control. *Robotics and Autonomous Systems*. 2018;103:162–174. DOI: 10.1016/j.robot.2018.02.019.
3. Yu D, Chen CLP, Ren CE, Sui S. Swarm control for self-organized system with fixed and switching topology. *IEEE Transactions on Cybernetics*. 2020;50(10):4481–4494. DOI: 10.1109/TCYB.2019.2952913.
4. Cao YU, Fukunaga AS, Kahng A. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*. 1997;4(1):7–27. DOI: 10.1023/A:1008855018923.
5. Gazi V, Fidan B, Marques L, Ordóñez R. Robot Swarms: Dynamics and Control. In book: *Mobile Robots for Dynamic Environments*. Chapter 4. ASME; 2015. P. 79–125. DOI: 10.1115/1.860526_ch4.
6. Bayindir L. A review of swarm robotics tasks. *Neurocomputing*. 2016;172:292–321. DOI: 10.1016/j.neucom.2015.05.116.
7. Kiselev O. Estimation of computational complexity for sub-optimal swarm control in non-cooperative games. 2020 4th Scientific School on Dynamics of Complex Networks and their Application in Intellectual Robotics (DCNAIR). 7-9 Sept. 2020, Innopolis, Russia. P. 133–134. DOI: 10.1109/DCNAIR50402.2020.9216775.
8. MailRuChamps/miniaicups [Electronic resource]. GitHub Inc.; 2018. Access mode: <https://github.com/mailruchamps/miniaicups/tree/master/agario>.
9. Dichkovskii A. Mini ai cup 2 or almost AgarIO – what could have been done to win [Electronic resource]. Habr; 2018. Access mode: <https://habr.com/ru/post/420737>.
10. von Neumann J, Morgenstern O. *Theory of Games and Economic Behavior*. Princeton: Princeton University Press; 2007. 776 p.
11. Glebov SG, Kiselev OM, Tarkhanov NN. *Nonlinear Equations with Small Parameter*. Vol. 16 of De Gruyter Series in Nonlinear Analysis and Applications. De Gruyter; 2017. 335 p.
12. Smoluchowski MV. Drei Vorträge über Diffusion, Brownsche Bewegung und Koagulation von Kolloidteilchen. *Physik. Zeit.* 1916;17:557–585 (in German).



Киселев Олег Михайлович – доктор физико-математических наук, ведущий научный сотрудник ИМВЦ УФИЦ РАН. Научные интересы – нелинейная динамика, асимптотические методы, теория управления компьютерными ботами. Опубликовал свыше 50 научных работ по указанным направлениям.

Россия, 420500 Иннополис, ул. Университетская, 1
Университет Иннополис
Россия, 450008 Уфа, ул. Чернышевского, 112
Институт математики с вычислительным центром
E-mail: ok@ufanet.ru
ORCID: 0000-0003-1504-7007