

УДК 681.3

*Крыжановский М.В., Микаэлян А.Л.*  
Институт оптико-нейронных технологий РАН, Москва  
E-mail: iont@iont.ru

## Нейросетевой подход к решению задачи управления группой агентов при выборе целей

В работе рассматривается задача планирования действий группы агентов при выборе целей. Предложен нейросетевой алгоритм, работающий заметно быстрее, чем имеющиеся в настоящий момент алгоритмы. Алгоритм основан на том, что действия каждого отдельного агента направлены на оптимизацию всей системы в целом. Показано, что предложенный подход существенно упрощает решение задачи по децентрализованному управлению и в  $m^2$  раз уменьшает вычислительную сложность алгоритма, где  $m$  – число целей. Предложена модификация оптимизируемого функционала, позволяющая обойти ряд стандартных трудностей, связанных с выбором коэффициента Лагранжа.

### Введение

Основные принципы децентрализованного коллективного управления группой агентов заключаются в том, что каждый объект группы должен самостоятельно решать задачу планирования своих действий в текущей ситуации. Выбор управления агентом осуществляется только на основе информации о текущей ситуации в среде, текущих состояниях и предполагаемых действиях других агентов группы на небольшом интервале времени. Под оптимальным управлением понимается такой выбор действия каждого из объектов, которое дает максимально возможное приращение в целевой функционал.

На основе этих принципов в [1,2] было продемонстрировано решение модельной задачи по управлению группой агентов при выборе целей, алгоритм решения которой был основан на частичном переборе. В данной работе предлагается иной подход, упрощающий решение этой задачи, позволивший добиться двух преимуществ. Во-первых, в развитом в настоящей работе нейросетевом подходе действия каждого из агентов направлены на улучшение состояния всей мультиагентной системы в целом, а не своего личного состояния. Ниже будет показано, что предлагаемый алгоритм решения задачи существенно упрощает решение задачи по децентрализованному управлению, что выражается в радикальном уменьшении вычислительной сложности алгоритма (скорость алгоритма увеличивается в  $m^2$  раз, где  $m$  – число целей). Во-вторых, предложена модификация оптимизируемого функционала, позволяющая ограничить вхождение системы в запретную область штрафов и, тем самым, позволяющая обойти ряд стандартных трудностей, связанных с выбором коэффициента Лагранжа.

### Постановка задачи при централизованном управлении.

Пусть имеется  $m$  целей и коллектив из  $N$  агентов  $R_k$  ( $k = 1, \dots, N$ ), которые должны эти цели поразить. Предполагается, что действия агентов происходят в

условиях заранее неизвестного противодействия противника. В этом случае необходимо рассматривать задачу с переменным количеством участников, поскольку часть агентов может быть уничтожена. Пусть ущерб от поражения  $i$ -й цели ( $i=1, \dots, m$ ) определяется выражением

$$\Phi_i = \Phi_i(n_i) \quad (1)$$

где  $n_i$  – число агентов, поразивших  $i$ -ю цель, а  $\Phi_i(n_i)$  – некие нелинейные монотонно нарастающие функции. Возможный вид этих функций представлен на рис.1, где  $n_{\max}$  – число агентов, попадание которых в данную цель гарантирует ее уничтожение, а максимальное значение ущерба, достигаемое в результате уничтожения  $i$ -й цели, нормировано на единицу  $\Phi(n_{\max}) = 1$ .

В качестве функционала для оптимизации необходимо выбрать величину суммарного ущерба, наносимого противнику в зависимости от степени поражения всех целей. Иными словами, задача оптимизации действий мультиагентной системы сводится к отысканию максимума функции

$$\Phi = \sum_{i=1}^m \Phi_i(n_i) \quad (2)$$

при очевидном условии

$$\sum_{i=1}^m n_i = N \quad (3)$$

Задача определения максимума функции  $\Phi$  при условии (3) сводится к задаче определения максимума функционала

$$W = \Phi - \mu(N - \sum n_i)^2 \quad (4)$$

где  $\mu$  – множитель Лагранжа. Поставленная задача обычно решается хорошо разработанными методами градиентного спуска [3], т.е. вычисляется градиент функционала и находится приращение переменной  $\delta \mathbf{n} = \alpha dW / d\mathbf{n}$ , обеспечивающее оптимизацию. Однако в рассматриваемом нами случае такой подход не применим, поскольку поиск экстремума  $W$  ведется в конфигурационном пространстве переменных. Действительно, компоненты вектора  $\mathbf{n} = (n_1, n_2, \dots, n_m)$  являются целыми числами и их приращения могут быть только целочисленными, т.е их следует описывать некоторым вектором  $\mathbf{u}$ , компоненты которого принимают значения  $u_i = 0, \pm 1$ . Поэтому, для нахождения оптимума функционала  $W$  будем использовать следующий итерационный процесс. Зададим начальное значение числа агентов  $N_0$  и некоторое случайное начальное распределение по целям  $\mathbf{n} = \mathbf{n}_0$ . На первом шаге итерационного процесса рассмотрим представление функционала

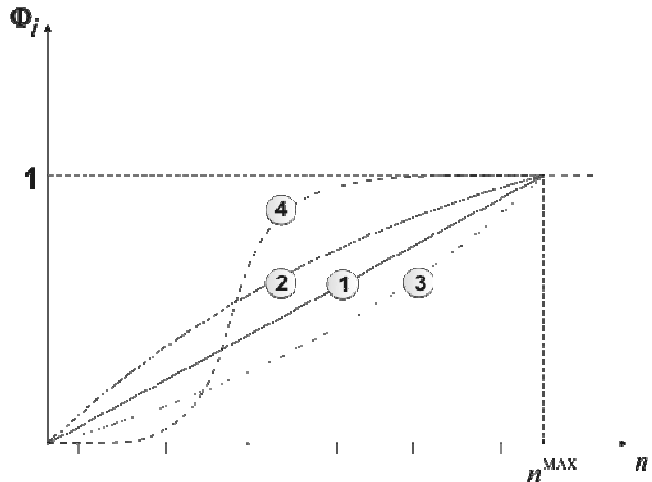


Рис.1. Возможные виды функции  $\Phi_i(n)$

$W = W(\mathbf{n})$  в окрестности точки  $\mathbf{n}_0$ , положив  $\mathbf{n} = \mathbf{n}_0 + \mathbf{u}$ , где  $\mathbf{u}$  – искомый вектор приращений. Тогда, проводя в (4) разложение в окрестности точки  $\mathbf{n}_0$  и заменяя малые приращения переменных  $d\mathbf{n}$  компонентами вектора  $\mathbf{u}$ , искомое представление с точностью до несущественной константы представим в виде

$$W(\mathbf{u}) = \sum_{i=1}^m p_i u_i - \frac{\mu}{2} \left( N_0 - N + \sum_{i=1}^m u_i \right)^2 \quad (5)$$

или в виде

$$W(\mathbf{u}) = -\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m T_{ij} u_i u_j + \sum_{i=1}^m h_i u_i \quad (6)$$

где введены следующие обозначения

$$h_i = p_i + \mu(N - N_0), \quad T_{ij} = \mu, \quad p_i = d\Phi_i / dn_i |_{\mathbf{n}=\mathbf{n}_0} \quad (7)$$

Как нетрудно заметить, выражение (6) совпадает с определением понятия энергии в нейросетевой модели Хопфилда [4], если принять, что  $T_{ij}$  – это матрица межсвязей, а  $h_i$  – это пороги нейронов. Соответственно этому, поиск максимума квадратичного функционала (5) будем вести следуя обычной [5-9] нейросетевой процедуре оптимизации: нейронная сеть инициализируется в начальное состояние  $u_i = 0$  и далее запускается процедура подъема по энергетической поверхности (6). Определив вектор  $\mathbf{u}$ , доставляющий функционалу  $W = W(\mathbf{n})$  максимум в окрестности точки  $\mathbf{n}_0$ , мы переходим в окрестность новой точки  $\mathbf{n}_1 = \mathbf{n}_0 + \mathbf{u}$ : в этой точке находятся новые значения величин величины  $p_i$  (производные функций  $\Phi_i$  в точке  $\mathbf{n}_1$ ) и снова проводится нейросетевая оптимизация функционала (6). Этот итеративный процесс продолжается до получения решения с необходимой точностью.

*Модификация алгоритма.* Однако такое (прямое) использование итерационного процесса определения вектора  $\mathbf{u}$  затруднительно в силу трудности подбора множителя Лагранжа в функции штрафа и неудобств, связанных с его применением на случай децентрализованного управления. Поэтому необходимо модифицировать выбор вектора  $\mathbf{u}$ , так чтобы влияние штрафа после одной итерации было минимальным. Для получения необходимых выражений рассмотрим, как используется градиентный метод для этой же задачи в области вещественных чисел  $n_i$ . При решении задачи градиентным методом приращения считаются по известным формулам, которые в принятых здесь обозначениях принимают вид:

$$u_i = \alpha \partial W / \partial u_i = \alpha (p_i - r_0) \quad (8)$$

$$r_0 = \mu \left( N_0 - N + \sum_{i=1}^m u_i \right) \quad (9)$$

Модифицируем выбор вектора  $\mathbf{u}$  так, чтобы влияние штрафа было минимальным. Для этого заменим выражение (8) новым

$$u_i = \alpha (p_i - r_0 + \bar{p}) \quad (10)$$

и попробуем оптимизировать процесс подходящим выбором параметра  $\bar{p}$ . Для этого определим значение штрафа после одного шага итерации:

$$u_i^{(1)} = u_i^{(0)} + \delta u_i \quad (11)$$

$$\delta u_i = \alpha (p_i - r_0 + \bar{p}) \quad (12)$$

$$r_0 = \mu \left( N_0 - N + \sum_{i=1}^m u_i^{(0)} \right) \quad (13)$$

$$r_1 = r_0 (1 - m\mu\alpha) + \mu\alpha \left( m\bar{p} - \sum_{i=1}^m p_i \right) \quad (14)$$

Из последнего выражения следует, что если принять

$$\bar{p} = m^{-1} \sum_{i=1}^m p_i \quad (15)$$

то значение штрафа при определенном выборе величины  $\mu\alpha$  будет уменьшаться и в случае  $r_0 = 0$  получим  $r_1 = 0$ .

Применим этот же прием и для работы в конфигурационном пространстве. По аналогии с модифицированным правилом (10) задания вектора приращений  $\mathbf{u}$  мы модифицируем искомым функционал  $W(\mathbf{u})$  к виду:

$$W(\mathbf{u}) = \sum_{i=1}^m (p_i - \bar{p}) u_i - \frac{\mu}{2} \left( N_0 - N + \sum_{i=1}^m u_i \right)^2 \quad (16)$$

Как видим, модификация свелась к простой замене величин  $p_i$  их отклонениями от среднего значения  $\bar{p}$ . Помимо этого, необходимо также модифицировать и условие инициализации сети Хопфилда, задавая начальные условия в виде  $u_i = \text{sgn}(p_i - \bar{p})$ . Многочисленные эксперименты показывают, что модификация функционала  $W(\mathbf{u})$  к виду (16) позволяет решать задачу оптимизации без обычных проблем, связанных с выбором множителя Лагранжа. Результаты этих экспериментов приведены в следующем разделе.

## Результаты численных экспериментов.

Анализ результатов экспериментов показывает, что поиск максимума функционала  $W(\mathbf{n})$  протекает успешно: достигается глобальный экстремум в случае применения функций  $\Phi$  вида 1÷3 (рис.1) и локальный- при использовании функций 4-го вида.

На рисунках 2-3 представлены результаты одного такого численного эксперимента. Эксперимент проводился при следующих условиях: количество целей  $m = 5$ ; количество агентов  $N = 15$ ; функции ущерба для  $i$ -й цели задавались в виде

$$\Phi_i = 1 - \exp(-n_i / \tau_i) \quad (17)$$

где  $\tau_i = \tau_0 / i$ , а постоянная экспоненты принималась равной  $\tau_0 = 10$ . На рис.2 представлено распределение по целям полученное после решения задачи при начальном распределении  $n_1(0) = N$ ,  $n_i(0) = 0$  для  $i \neq 1$ . На рис.3 показано изменение со временем суммарного ущерба  $\Phi = \Phi(t)$ , соответствующее перераспределению агентов по целям в ходе решения задачи.

Выяснилось, однако, что такой метод решения как правило дает локальный максимум. Для иллюстрации этого утверждения условия эксперимента были

видоизменены: для разных целей вводились разные функции ущерба. Например, при тех же параметрах  $m$ ,  $N$  функции ущерба для целей  $i=1, \dots, 4$  задавались

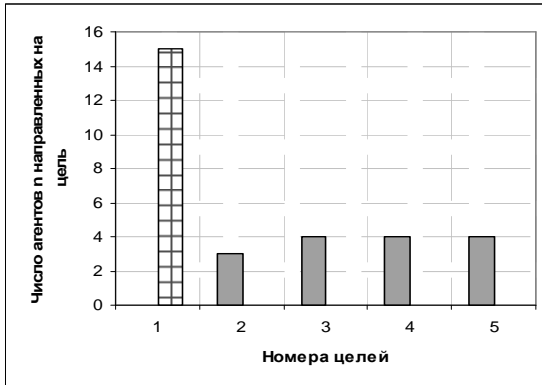


Рис. 2. Распределение по целям в результате оптимизации: заштрихованная область – начальное распределение, сплошная заливка – полученное решение.

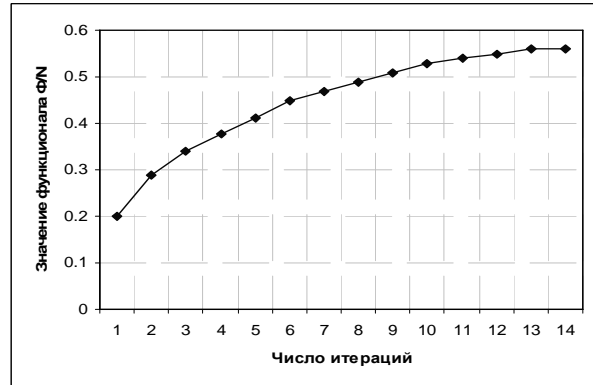


Рис. 3. Изменение значения функционала  $\Phi/N$  с течением времени (процесс оптимизации).

выражением (17), а функция  $\Phi_5$  была заменена на сигмоидальную  $\Phi_5 = \Phi(0) + \text{th}(n/\theta)$ . Выяснилось, что получаемое решение зависит от начальных условий, т.е. глобальный оптимум достигается не всегда. Например, при одном начальном распределении ( $n_1 = N$ ,  $n_i = 0$  для  $i \neq 1$  при  $t = 0$ ) было получено оптимальное решение ( $n_1 = 5$ ,  $n_2 = 4$ ,  $n_3 = n_4 = 3$ ,  $n_5 = 0$ ) с максимальным значением функции ущерба  $\Phi = 0.24$ , а при других начальных условиях ( $n_1 = n_2 = n_3 = n_4 = 0$  и  $n_5 = N$  при  $t = 0$ ) было получено иное конечное распределение по целям  $n_1 = n_2 = n_3 = n_4 = 0$  с большим значением функционала  $\Phi = 0.32$ .

*Видоизменение алгоритма.* Чтобы избежать попадания в локальный экстремум и, по-возможности, создать алгоритм, доставляющий глобальный оптимум, расчет величин  $p_i$  был изменен. Для этого изменялся сам вид функции  $\Phi_i(n)$  участвующей в вычислении производной. По функции  $\Phi_i(n)$  определялась производная  $p_i = d\Phi_i / dn$  для всего диапазона переменной  $0 \leq n \leq N$  и находились ее максимальное значение  $p_i^{\max}$  и соответствующая координата  $n_i^{\max}$ . В диапазоне  $0 \leq n \leq n_i^{\max}$  полагалось  $p_i = p_i^{\max}$ , а в остальной части диапазона производная  $p_i$  определялась прежним выражением. Такое видоизменение расчета позволило находить глобальный оптимум для функции  $\Phi$ . Рисунки 4÷5 иллюстрируют это положение.

Таким образом, в условиях централизованного управления мы решили задачу определения количества агентов направляемых на каждый объект-цель. При этом, определение цели для каждого из агентов определяется просто. Агенты, имеющие направление на цель, для которой количество объектов не изменилось или увеличилось, свое направление не изменяют. Для агентов, принадлежащих другим целям, переопределение направления определяется их порядковыми номерами и "расстояниями" к ближайшим целям.

## Решение задачи для агентов, управляемых автономно

Решение задачи по выбору цели для агентов, управляемых автономно, ведется на основе решения задачи при централизованном управлении, для которого характерно представление о целях как о самостоятельных объектах конкурирующих между собой за увеличение количества агентов направленных на нее. Поэтому выражение (4) для  $W(\mathbf{u})$  мы представим как сумму

$$W(\mathbf{u}) = \sum_{i=1}^m W_i(\mathbf{u}) \quad (18)$$

где для каждой цели вводится свой оптимизационный функционал

$$W_i(\mathbf{u}) = (p_i - \bar{p}) u_i - \frac{\mu}{2} (N_0 - N + u_i) \left( N_0 - N + \sum_{s=1}^m u_s \right) \quad (19)$$

Для того, чтобы агент мог определять свое движение он должен иметь информацию о целях, информацию о текущих состояниях всех агентов и об их намерениях изменить направления. Поэтому каждый агент имеет таблицу, которая описывает состояние целей: функция каждой  $i$ -ой цели  $\Phi_i$ , количество агентов направленных на нее и их номера, производную от функции цели и требование об изменении числа агентов (компоненты вектора  $\mathbf{u}$ ). Содержание таблиц до начала движения задается извне.

Обмениваясь информацией с другими членами коллектива в процессе движения агент  $R_k$ , получает производные  $p_i$  и вычисляет ее среднее значение  $\bar{p}$ . Это дает возможность определить агентам начальное приближение вектору  $\mathbf{u}$ . На этой основе агент  $R_k$ , направленный на  $i$ -ю цель определяет величину  $W_i(\mathbf{u})$  и, соответственно, компоненту  $u_i$ . После определения агентами вектора  $\mathbf{u}$  и занесения его в таблицу наступает этап принятия решения.

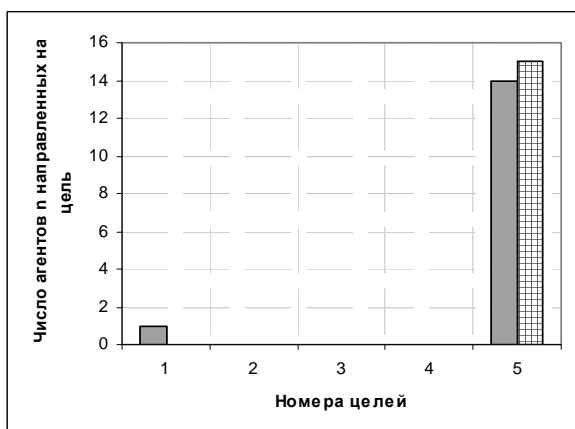


Рис.4. На рисунке показан результат оптимизации. Распределение по целям: заштрихованная область – начальное распределение, сплошная заливка – распределение после оптимизации.

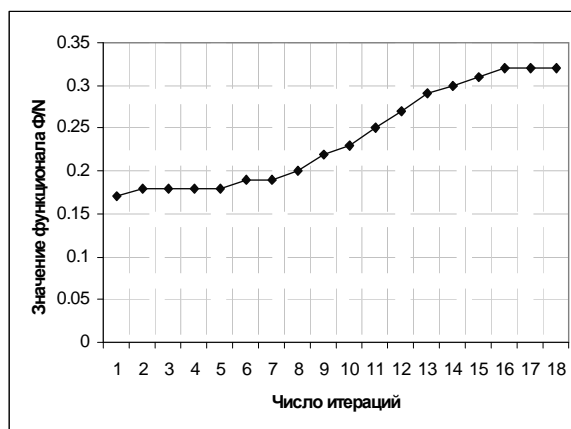


Рис.5. Изменение значения функционала  $\Phi/N$  с течением времени.

Перераспределение целей между агентами коллектива определяется путем анализа компонент векторов  $\mathbf{p}$  и  $\mathbf{u}$ . В первую очередь удовлетворяются потребности в изменении пары целей, для которых отклонение от среднего  $|p_i - \bar{p}|$  максимально и у которых компоненты  $\mathbf{u}$  разных знаков. Агенты, принадлежащие к целям, для которых компоненты  $\mathbf{u}$  отрицательны, изменяют направление, а именно: среди агентов, принадлежащих  $i$ -й цели, изменению цели подлежит агент, имеющий максимальный порядковый номер. Приняв решение о выборе цели, агент сообщает о нем другим членам коллектива. Таблицы агентов модифицируются, а соответствующие компоненты вектора  $\mathbf{u}$  обнуляются. После изменения направления агентов коллектива, процесс, описанный выше, повторяется.

## Заключение

Приведенный выше алгоритм существенно упрощает решение задачи по децентрализованному управлению и на порядки уменьшает вычислительную сложность алгоритма. Анализ вычислительной сложности и результаты множественных экспериментов показывают, что скорость алгоритма с ростом числа целей  $m$  увеличивается в  $m^2$  раз, по сравнению с стандартным подходом.

Предложенная модификация оптимизируемого функционала, заключающаяся в замене градиентов ( $p_i$ ) их отклонением от среднего ( $|p_i - \bar{p}|$ ), резко сузила возможность вхождения системы в запретную область штрафов. При этом, алгоритм стал значительно менее чувствительным к выбору значения коэффициента Лагранжа, позволяя избегать стандартных проблем, возникающих при оптимизации функционалов с штрафными членами.

Работа выполнена при поддержке РФФИ (проекты №04-07-90038 и №01-01-00109)

## Литература

- [1]. И.А.Каляев, А.Р.Гайдук, С.Г.Капустян. Распределенные системы планирования действий коллективов роботов. // М.: Янус-К, 2002.
- [2]. Каляев И.А. Использование принципов коллективного принятия решений при управлении группой автоматических лифтов. // Мехатроника, №4, 2001.
- [3]. G.E.Forsythe, C.B.Moler. Computer Solution of Linear Algebraic Systems. Englewood Cliffs, New Jersey, PrenticeHall, 1967.
- [4]. J.J.Hopfield. Neural Networks and physical systems with emergent collective computational abilities. Proc.Nat.Acad.Sci.USA. 79, 2554-2558 (1982).
- [5]. J.J.Hopfield, D.W.Tank. Neural computation of decisions in optimization problems. Biological Cybernetics 52, 141-152 (1985).
- [5]. J.J.Hopfield, D.W.Tank. Computing with neural circuits: A Model. Science 233, 625-633 (1986).
- [6]. Y.Fu, P.W.Anderson. Application of statistical mechanics to NP-complete problems in combinatorial optimization. Journal of Physics A. 19, 1605-1620 (1986).
- [7]. T.Poggio, F.Girosi. regularization algorithms for learning that are equivalent to multilayer networks. Science 247, 978-982 (1990).

- [8]. Б.В.Крыжановский, Б.М. Магомедов, А.Л.Микаэлян. Доменная модель нейронной сети. ДАН 2005, т.401, №4, с.462-466.
- [9]. В.Kryzhanovsky, В.Magomedov. Application of domain neural network to optimization tasks. Proc. of XVII International Conf. on Artificial Neural Networks, ICANN 2005. Poland. LNCS 3697, Part II, pp.397-403.

# NEURAL-NETWORK APPROACH TO THE TARGET ASSIGNMENT PROBLEM IN MULTYAGENT SYSTEM

M.V.Kryzhanovsky, A.L.Mikaelian

Institute of Optical Neural Technologies RAS, Vavilova st. 44/2, Moscow 199333  
iont@iont.ru

**Abstract.** A neural network approach to the problem of discrete optimization of multi-agent system's functional is discussed. The approach ability is illustrated by solving the problem of assigning targets among a number of agents. The proposed neural network algorithm makes it considerably easier to solve the problem of decentralized control and decreases the amount of computations in  $m^2$  times, where  $m$  is the number of targets. The algorithm is based on the fact that the behavior of each agent results in an optimization of the state of the system as a whole. The modification of the functional proposed here allows one to avoid usual problems in selecting the Lagrangian coefficient.