

УДК 519.68
DOI: 10.15827/0236-235X.129.020-026

Дата подачи статьи: 05.09.19
2020. Т. 33. № 1. С. 020–026

Использование алгоритмов оптимизации с самообучением для управления динамически изменяющимися системами

*В.А. Костенко*¹, к.т.н., доцент факультета ВМК, kostmsu@gmail.com

¹ *Московский государственный университет им. М.В. Ломоносова, г. Москва, 119991, Россия*

В статье предлагается подход к управлению динамически изменяющимися системами. В ходе работы они могут изменять свое состояние: состав оборудования, нагрузку и выполняемые функции. Необходимо выбирать значения управляющих параметров в зависимости от состояния системы таким образом, чтобы обеспечить требуемые значения характеристик ее работы.

Для решения этой задачи авторы предлагают использовать оптимизационные алгоритмы с самообучением: направленного случайного поиска с самообучением и муравьиные. Эти алгоритмы, действуя методом проб и ошибок, позволяют настраиваться на текущее состояние системы за счет введения в алгоритмы памяти об удачном и неудачном выполнении предыдущих шагов.

Основная идея применения алгоритмов с самообучением для управления системой заключается в том, что задача управления рассматривается как задача безусловной оптимизации. Элементами вектора оптимизируемых переменных являются управляющие параметры системы. Шаги алгоритмов рассматриваются как возможные действия по управлению системой (операции изменения значений управляющих параметров). Целевая функция может быть задана одним из двух способов: как суммарное среднее отклонение характеристик работы системы от требуемых (важность характеристик можно учесть с помощью весов при суммировании отклонений) и как максимальное отклонение характеристик работы системы от требуемых.

Предложенный подход к управлению допускает нестабильное поведение окружающей среды, ограниченность информации об управляемой системе и позволяет учитывать наличие многих характеристик работы системы, значения которых требуется поддерживать в заданных пределах. Если отказ элементов системы не приводит к отказу системы в целом, а лишь ухудшает значения характеристик работы системы, то при использовании данного подхода отклонения значений характеристик от требуемых будут минимизироваться.

Ограничением данного подхода является допустимость переходного процесса при смене состояния системы. При изменении состояния системы алгоритму потребуется ряд шагов для переобучения. Некоторые шаги могут приводить к нарушению требуемых характеристик работы системы.

Ключевые слова: динамическая система, управление, математическое программирование, муравьиные алгоритмы.

Динамически изменяющаяся система характеризуется состоянием (состав оборудования, нагрузка на систему, поведение окружающей среды, выполняемые системой функции), значениями характеристик работы системы и данными управляющих параметров.

Состояние системы меняется, поэтому требуется выбирать управляющие параметры таким образом, чтобы обеспечить требуемые значения характеристик ее работы.

Примерами таких систем являются технологические процессы, компьютерные сети, птицефабрики, здания с автоматическим управлением климатом. Так, нагрузка на компьютерную сеть задается трафиком, на систему автоматического управления климатом – количеством людей, рассеиваемой мощностью обо-

рудования, работающего в данный момент в здании. В качестве характеристик работы системы для здания с автоматическим управлением климатом выступают внутренние температура, влажность, количество углекислого газа, а для компьютерной сети – задержка передачи пакетов, количество потерянных пакетов. Примером управляющих параметров для здания с автоматическим управлением климатом являются температура обогревателей, степень открытия форточек, обороты двигателей вентиляционной системы.

Проблемой применения известных в теории автоматического управления методов [1] для управления динамически изменяющимися системами является то, что состояние системы меняется со временем и плохо прогнозируется

(возможна большая ошибка прогноза) или вообще не прогнозируется. Проблемой применения методов, использующих обучающую выборку (например, нейросетей [2–4], метода опорных векторов [5], аксиоматического подхода [6–9]) является то, что обучение ведется на одной системе, а полученные алгоритмы применяются к другой.

Для решения этой задачи предполагается использовать оптимизационные алгоритмы с самообучением: алгоритмы направленного случайного поиска с самообучением и муравьиные алгоритмы. Эти алгоритмы, действуя методом проб и ошибок, позволяют настраиваться на текущее состояние системы за счет введения памяти об удачном и неудачном выполнении предыдущих шагов.

Основная идея применения алгоритмов с самообучением для управления системой заключается в том, что задача управления рассматривается как задача оптимизации, а пробные шаги алгоритмов – как возможные действия управления системой (операции изменения значений управляющих параметров).

Задача управления как задача безусловной оптимизации

Задача безусловной оптимизации заключается [10] в нахождении компонентов вектора $X = (x_1, \dots, x_n)$ (оптимизируемых параметров), минимизирующих целевую функцию $f(X)$.

Введем следующие обозначения: $S(t)$ – состояние системы; $Y(t) = (y_1(t), \dots, y_k(t))$ – вектор значений характеристик работы системы; $X(t) = (x_1(t), \dots, x_n(t))$ – вектор значений управляющих параметров; $Y^*(t) = (y_1^*(t), \dots, y_k^*(t))$ – вектор требуемых значений характеристик работы системы.

Значения характеристик работы системы изменяются со временем в зависимости от состояния системы, значений управляющих параметров и текущих характеристик: $Y(t + \Delta) = F(Y(t), X(t), S(t))$.

Здесь $1/\Delta$ – частота изменения управляющих параметров для коррекции работы системы. Значение Δ определяется инерционностью системы.

Информация о состоянии системы $S(t)$ недоступна или ограничена. Как следствие – функция $F(Y(t), X(t), S(t))$ неизвестна.

Под инерционностью системы будем понимать скорость изменения значений характеристик работы системы при изменении значений управляющих параметров. Поскольку функция

$F(Y(t), X(t), S(t))$ и ее аргумент $S(t)$ неизвестны, оценка значения Δ может быть получена только экспериментально, если она не зависит от $S(t)$. В противном случае Δ необходимо рассматривать как оптимизируемый параметр, значение которого должно изменяться динамически в ходе работы алгоритма.

Пусть $C_k \cdot \tau$ – время работы алгоритма изменения управляющих параметров, тогда должно выполняться условие $\Delta \geq C_k \cdot \tau$, где C_k – число итераций алгоритма; τ – время выполнения одной итерации.

Задачу управления изменяющимися динамическими системами можно сформулировать как задачу безусловной оптимизации.

Элементами вектора оптимизируемых параметров являются управляющие параметры системы.

Целевую функцию $f(X)$ можно определить одним из следующих способов.

- Суммарное среднее отклонение характеристик работы системы от требуемых:

$$\min_Y \sum_{i=1}^k |y_i - y_i^*|.$$

Важность характеристик можно учесть с помощью весов при суммировании отклонений.

- Максимальное отклонение характеристик работы системы от требуемых:

$$\min_Y (\max_i |y_i - y_i^*|).$$

Алгоритм направленного случайного поиска с самообучением

Рассмотрим основные принципы построения алгоритмов направленного случайного поиска с самообучением [11].

Основой методов направленного случайного поиска служит итерационный процесс:

$$X_{k+1} = X_k + \alpha_k \cdot \frac{\xi}{\|\xi\|}, \quad k = 0, 1, \dots,$$

где α_k – величина шага; $\xi = (\xi_1, \dots, \xi_n)$ – некоторая реализация n -мерного случайного вектора ξ .

Алгоритмы случайного направленного поиска с самообучением заключаются в перестройке вероятностных характеристик поиска, то есть в определенном целенаправленном воздействии на случайный вектор ξ .

Это достигается введением вектора памяти $P^k = (p_1^k, p_2^k, \dots, p_n^k)$, где p_j^k – вероятность выбора положительного направления по j -й координате на k -м шаге.

Алгоритм направленного случайного поиска с самообучением определяется тремя элементами:

- алгоритмом выбора пробных состояний на текущем шаге;
- решающим правилом, по которому на каждом шаге выбирается новое текущее приближение решения;
- алгоритмом самообучения, корректирующим вектор памяти с точки зрения информации, полученной на текущем шаге.

Рассмотрим алгоритм самообучения с произвольным законом изменения вероятности [11].

Пусть $p_j^{(i)} = p(w_j^{(i)})$. Вид функции может быть различным, но она должна быть монотонной и неубывающей.

Линейная зависимость:

$$p(w) = \begin{cases} 0 & \text{при } w < -1, \\ \frac{1}{2} (1 - w) & \text{при } -1 \leq w < 1, \\ 1 & \text{при } 1 \leq w. \end{cases}$$

Экспоненциальная зависимость:

$$p(w) = \begin{cases} \frac{1}{2} e^{aw} & \text{при } w \leq 0, \\ 1 - \frac{1}{2} e^{-aw} & \text{при } w > 0; \end{cases}$$

$$w_j^{(i+1)} = w_j^{(i)} - \delta \cdot \text{sign}(\Delta x_j^{(i)} \cdot \Delta f^{(i)})$$

где δ – шаг, определяющий скорость обучения.

Таким образом, алгоритм в ходе своей работы осуществляет перестройку вероятностных характеристик поиска путем целенаправленного воздействия на случайный вектор ξ . Он уже перестает быть равновероятным и в результате самообучения приобретает определенное преимущество в направлениях наилучших шагов. Это достигается введением вектора памяти. Алгоритм рекуррентно корректирует значения компонентов этого вектора на каждой итерации в зависимости от того, насколько удачным/неудачным (изменилось значение целевой функции) был сделанный шаг.

Муравьиный алгоритм

Муравьиные алгоритмы позволяют автоматически настраиваться на пример задачи (заданы конкретные значения исходных данных) путем дополнительной разметки исходных данных, которая используется для построения решения на каждой итерации алгоритма и уточняется по мере увеличения числа итераций.

Идея муравьиных алгоритмов [12, 13] основана на моделировании поведения муравьев при нахождении кратчайшего пути от муравейника к источнику пищи. Муравьи при перемещении оставляют особое вещество – феромон, который используется в дальнейшем другими муравьями при выборе пути. Чем выше концентрация феромона на том или ином пути, тем больше вероятность того, что муравьи выберут для движения именно этот путь.

Для применения муравьиных алгоритмов задачу надо свести к нахождению в графе замкнутого пути минимальной длины, в который каждая вершина входит однократно (задача коммивояжера). Общую схему работы муравьиных алгоритмов можно представить следующим образом:

- задание начального количества феромона на ребрах графа, количества и начального положения муравьев;
- построение муравьями пути (каждый муравей строит путь независимо от остальных);
- обновление количества феромона на ребрах;
- при невыполнении условия останова возврат к построению пути.

Операция построения муравьем пути. Муравей строит путь, переходя из одной вершины в другую. Пройденные муравьем вершины добавляются в табу-список (память муравья), чтобы избежать их повторного посещения. Вероятность перехода муравья из i -й вершины в j -ю зависит от количества феромона на данном ребре, значения локальной целевой функции на ребре и состояния табу-списка. Вероятность перехода k -го муравья из i -й вершины в j -ю на t -й итерации алгоритма рассчитывается по следующей формуле:

$$P_{ij,k}(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij}(t))^\beta}{\sum_{l \in J_k} (\tau_{il}(t))^\alpha \cdot (\eta_{il}(t))^\beta}, & j \notin L_k, \\ 0, & j \in L_k. \end{cases}$$

Здесь $\tau_{ij}(t)$ – количество феромона на ребре (i, j) ; $\eta_{ij}(t)$ – значение локальной целевой функции на ребре (i, j) ; $\alpha \geq 0$ и $\beta \geq 0$ – параметры алгоритма, определяющие важность феромонного следа и локальной целевой функции; L_k – множество вершин, включенных в табу-список муравья k .

Операция обновления количества феромона на ребрах. После того, как все муравьи завершили построение путей, обновляется количество феромона на ребрах:

$$\tau_{ij}(t+1) = (1-p) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij,k}(t),$$

$$\Delta\tau_{ij,k}(t) = \begin{cases} F(T_k(t)), (i, j) \in T_k(t), \\ 0, (i, j) \notin T_k(t). \end{cases}$$

Здесь $T_k(t)$ – путь, построенный k -м муравьем; $F(T)$ – целевая функция, определяющая качество пути; m – количество муравьев; $p \in [0, 1]$ – коэффициент испарения феромонов. Испарение феромонов вводится во избежание попадания алгоритма в локальный оптимум, когда первый найденный путь с относительно хорошим значением целевой функции становится единственно значимым.

Рассмотрим построение муравьиных алгоритмов для задачи минимизации суммарного запаздывания работ [14]. По такому же принципу может быть построен муравьиный алгоритм для управления динамически изменяющимися системами.

Имеется множество работ $T = \{T_1, \dots, T_n\}$, для каждой из которых заданы время выполнения t_i и директивные сроки выполнения $[0, f_i]$.

Требуется для каждой работы определить время начала выполнения. Критерий оптимальности – суммарное время запаздывания

$$F(\pi) = \sum_{j=1}^{NW} \max(0, c_j(\pi) - f_j).$$

Вводятся два класса вершин: вершины работы и вершины позиции в упорядоченном списке работ. Табу-список – размещенные работы и занятые позиции (каждая вершина присутствует в маршруте только один раз). Переход к следующей позиции в списке осуществляется последовательно. Расписание однозначно определяется последовательностью работ в упорядоченном списке, построенном муравьиным алгоритмом. В качестве локальных целевых функций на ребрах графа могут быть использованы: $h^1_{ij} = 1/f_j$ – минимальный директивный срок завершения, $h^2_{ij} = 1/t_j$ – минимальное время выполнения, $h^3_{ij} = 1/\max(S + t_j, f_j)$ – выбор минимального запаздывания работы (S – время завершения всех размещенных работ).

Проблема переходного процесса

При изменении состояния системы алгоритму необходим ряд пробных шагов для переобучения. Некоторые пробные шаги могут приводить к нарушению требуемых характеристик работы системы.

Если есть модель системы, избежать этой

проблемы можно, обучаясь на модели, а не на показаниях датчиков системы.

Также эта проблема может быть ослаблена заданием условий применения операций изменения управляющих параметров. Например, если температура в здании выше требуемой, то применение операции увеличения температуры нагревателей надо запрещать.

Если требуемые значения характеристик работы системы задаются в виде интервала, то можно попытаться настроить алгоритм так, чтобы во время переобучения значения характеристик находились в заданном интервале. Для алгоритмов случайного поиска подбирается значение шага, определяющего скорость обучения. Для муравьиных алгоритмов подбираются значения коэффициента испарения феромонов и значения параметров, определяющие важность феромонного следа и локальной целевой функции.

Программная система и требования к аппаратной платформе

Программная система управления динамически изменяющимися системами включает следующие основные модули:

- модуль ввода информации от датчиков и исполнительных устройств;
- вычислительный модуль;
- модуль передачи информации исполнительным устройствам;
- модуль задания параметров алгоритма;
- модуль диагностики датчиков и исполнительных устройств.

Модуль ввода информации от датчиков и исполнительных устройств выполняет:

- первичную обработку информации, получаемой от датчиков: нормализацию данных и фильтрацию;
- формирование данных для вычислительного модуля;
- получение данных от вычислительного модуля и формирование сообщений для контроллеров исполнительных устройств.

В вычислительном модуле запускаются алгоритмы выработки управляющих параметров для исполнительных устройств.

Модуль задания параметров алгоритмов позволяет:

- выбирать алгоритм;
- задавать параметры алгоритма;
- задавать допустимый диапазон изменения значений каждого управляющего параметра.

Модуль диагностики датчиков и исполнительных устройств осуществляет контроль их исправности.

Отличительными требованиями, предъявляемыми к аппаратной платформе, являются:

- наличие модулей *аналого-цифровых* (АЦП) и *цифро-аналоговых преобразователей* (ЦАП);

- наличие широкой номенклатуры интерфейсов связи;

- ориентация вычислительной системы на эксплуатацию в промышленных условиях;

- отсутствие регулярного системного администрирования вычислительной системы.

Первые два требования обусловлены широкой номенклатурой датчиков и исполнительных устройств, используемых на управляемом объекте, следующие два – длительной автономной работой системы управления в жестких условиях эксплуатации.

Наиболее полно всем этим требованиям удовлетворяет *гетерогенная вычислительная платформа* (ГВП), разработанная в НИИ ВК им. М.А. Карцева [15, 16].

В состав ГВП могут входить функциональные модули:

- коммутации;

- центрального процессора (на базе процессора Intel i7, 4 ядра);

- реконфигурируемого процессора (на базе ПЛИС Xilinx Virtex 6) с возможностью расширения функциональных возможностей путем установки мезонинных модулей АЦП/ЦАП;

- графического процессора (на базе NVidia Quadro K2100M);

- процессора «Эльбрус» (на базе процессора «Эльбрус-4С»);

- процессора «Байкал» (на базе процессора «Байкал-Т1»);

- интерфейсный оптический PCIe;

- носителя SSD диска (интерфейс SATA3).

В зависимости от требований к условиям эксплуатации ГВП доступна в трех исполнениях:

- с вентиляторным охлаждением (предназначена для эксплуатации в отапливаемых помещениях, температурный диапазон – от 0 до +40 °С, электропитание – однофазная сеть переменного тока, напряжение питающей сети – от 187 до 242 В);

- с кондуктивным отводом тепла (для эксплуатации во встроенных системах, температурный диапазон – от –55 до +55 °С, электропитание – сеть постоянного тока, напряжение питающей сети – от 18 до 36 В);

- с гибридным отводом тепла (температурный диапазон – от –55 до +55 °С, электропитание – сеть постоянного тока, напряжение питающей сети – от 18 до 36 В).

Заключение

Предложенный подход к управлению допускает нестабильное поведение окружающей среды, ограниченность информации об управляемой системе и позволяет учитывать наличие многих характеристик работы системы, значения которых требуется поддерживать в заданных пределах. Если отказ элементов системы не приводит к отказу системы в целом, а только ухудшает значения характеристик работы системы, то при использовании данного подхода отклонения значений характеристик от требуемых будут минимизироваться.

Работа выполнена при финансовой поддержке РФФИ, грант № 19-07-00614.

Литература

1. Ким Д.П. Алгебраические методы синтеза систем автоматического управления. М.: Физматлит, 2014. 192 с.
2. Уоссермен Ф. Нейрокомпьютерная техника. Теория и практика. М.: Мир, 1992. 184 с.
3. Крыжановский Б.В., Микаэлян А.Л. О распознающей способности нейросети на нейронах с параметрическим преобразованием частот // Доклады РАН. 2002. Т. 383. № 3. С. 318–321.
4. Осовский С. Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. 216 с.
5. Berndt D.J., Clifford J. Using dynamic time warping to find patterns in time series. Proc. AAAI Technical Report WS KDD-94, 1994, pp. 229–248.
6. Shcherbinin V.V., Kostenko V.A. A genetic algorithm for training recognizers of latent abnormal behavior of dynamic systems. Proc. 7th Int. Joint Conf. Computational Intelligence, Lisbon, Portugal, 2015, vol. 1, pp. 358–365.

7. Kostenko V.A., Shcherbinin V.V. Training methods and algorithms for recognition of nonlinearly distorted phase trajectories of dynamic systems. *Optical Memory and Neural Networks*, 2013, vol. 22, no. 1, pp. 8–20.
8. Костенко В.А., Коваленко Д.С. Алгоритмы распознавания нештатного поведения динамических систем, устойчивые к нелинейным искажениям фазовых траекторий системы // Передовые информационные технологии, средства и системы автоматизации и их внедрение на российских предприятиях: сб. тр. Междунар. науч.-практич. конф. М.: Изд-во ИПУ РАН, 2011. С. 897–905.
9. Коваленко Д.С., Костенко В.А., Щербинин В.В. Параметрическое семейство алгоритмов распознавания нелинейно искаженных фазовых траекторий динамических систем // *Нейроинформатика*: сб. тр. XIV Всерос. науч.-технич. конф. М.: Изд-во НИЯУ МИФИ, 2012. Ч. 1. С. 266–276.
10. Мину М. Математическое программирование. Теория и алгоритмы. М.: Наука, 1990. 488 с.
11. Растринин Л.А. Статистические методы поиска. М.: Наука, 1968. 398 с.
12. Dorigo M. *Optimization, Learning and Natural Algorithms*. Dipartimento di Elettronica, Milano, 1992, 140 p.
13. Штовба С.Д. Муравьиные алгоритмы: теория и применение // *Программирование*. 2005. № 4. С. 3–18.
14. Гафаров Е.Р. Гибридный алгоритм решения задачи минимизации суммарного запаздывания для одного прибора // *Информационные технологии*. 2007. № 1. С. 30–37.
15. Барыбин А.К., Лобанов В.Н., Чельдиев М.И., Чучкалов П.Б. Реконфигурируемая вычислительная платформа с разнородной архитектурой // *Вопросы радиоэлектроники*. 2016. № 7. Т. 2. С. 70–77.
16. Баранов Л.Д., Лобанов В.Н., Чельдиев М.И. Применение многопроцессорной вычислительной платформы с гетерогенной архитектурой для решения задач гидроакустики и радиолокации // *Вопросы радиоэлектроники*. 2018. № 5. С. 7–16.

Software & Systems
DOI: 10.15827/0236-235X.129.020-026

Received 05.09.19
2020, vol. 33, no. 1, pp. 020–026

Using self-learning optimization algorithms to manage dynamically changing systems

V.A. *Kostenko*¹, Ph.D. (Engineering), Associate Professor, kostmsu@gmail.com

¹ *Lomonosov Moscow State University (MSU), Moscow, 119991, Russian Federation*

Abstract. The paper proposes an approach to the management of dynamically changing systems. Such systems has a response that during the system operation their state may: the list of the equipment, the load of the system and the functions performed by the system. It is required to choose the values of control parameters depending on the system state in such a way as to provide the required values of the characteristics of the system.

To solve this problem, the authors plan to use optimization algorithms with self-learning: directional random search algorithms with self-learning and ant algorithms. These algorithms, operating by trial and error method, allow you to tune in to the current state of the system. The introducing into the algorithms the memory of successes and failures on the previous steps helps to achieve it.

The main idea of using self-learning algorithms to control the system is that we will consider the control problem as an unconditional optimization problem. The elements of the vector of optimized variables are the control parameters of the system. The authors consider the algorithm steps as possible actions of system control (operations of changing control parameter values). The objective function can be set in one of two ways: by the calculation of the total mean probable characteristics error of the system from the required ones. The weights help take into account the characteristics importance by summing deviations; the researching of the maximum characteristics deviation of the system operation from the required ones.

The proposed management approach allows the unstable environmental behavior, limited managed system information and allows you to take into account the presence of many system characteristics, their values must be maintained within specified limits. If the system elements failure does not lead to the whole system failure, but only worsens the system characteristics values, this approach will minimize the characteristics values deviations from the required.

A limitation of this approach is the transition process allow ability by changing the system state. In this way, the algorithm will require a number of steps for retraining. Some steps cause disturbance the required system performance.

Keywords: dynamic system, control, mathematical programming, ant colony algorithms.

Acknowledgements. This work is financially supported by RFBR, project no. 19-07-00614.

References

1. Kim D.P. *Algebraic Methods for the Synthesis of Automatic Control Systems*. Moscow, Fizmatlit Publ., 2014, 192 p.
2. Wasserman P.D. *Neurocomputer Technology: Theory and Practice*. NY, Van Nostrand Reinhold Publ., 1989, 230 p. (Rus. ed.: Moscow, Mir Publ., 1992, 184 p.).
3. Kryzhanovsky B.V., Mikaelyan A.L. On the recognition ability of a neural network on neurons with parametric frequency conversion. *Doklady Mathematics*. 2002, vol. 383, no. 3, pp. 318–321 (in Russ.).
4. Osovsky S. *Neural Networks for Information Processing*. Moscow, Finansy i Statistika Publ., 2002, 216 p.
5. Berndt D.J., Clifford J. Using dynamic time warping to find patterns in time series. *Proc. AAAI Technical Report WS KDD-94*. 1994, pp. 229–248.
6. Shcherbinin V.V., Kostenko V.A. A genetic algorithm for training recognizers of latent abnormal behavior of dynamic systems. *Proc. 7th Int. Joint Conf. Computational Intelligence*. Lisbon, Portugal, 2015, vol. 1, no. 1, pp. 358–365.
7. Kostenko V.A., Shcherbinin V.V. Training methods and algorithms for recognition of nonlinearly distorted phase trajectories of dynamic systems. *Optical Memory and Neural Networks*. 2013, vol. 22, no. 1, pp. 8–20.
8. Kostenko V.A., Kovalenko D.S. Algorithms for recognizing abnormal behavior of dynamic systems that are resistant to non-linear distortions of the system phase trajectories. *Proc. Int. Sc. and Pract. Conf. AITA*. Moscow, 2011, pp. 897–905 (in Russ.).
9. Kovalenko D.S., Kostenko V.A., Shcherbinin V.V. Parametric family of recognition algorithms for nonlinearly distorted phase trajectories of dynamic systems. *Proc. XIV Conf. Neuroinformatics-2012*. Moscow, 2012, part 1, pp. 266–276 (in Russ.).
10. Minu M. *Mathematical Programming. Theory and Algorithms*. Moscow, Nauka Publ., 1990, 488 p. (in Russ.).
11. Rastrigin L.A. *Statistical Search Methods*. Moscow, Nauka Publ., 1968, 398 p.
12. Dorigo M. *Optimization, Learning and Natural Algorithms*. PhD thesis, Milano, 1992, 140 p.
13. Shtovba S.D. Ant algorithms: theory and application. *Programming and Computer Software*. 2005, no. 4, pp. 3–18 (in Russ.).
14. Gafarov E.R. Hybrid algorithm for solving the problem of minimizing the total delay for a single device. *Information Technologies*. 2007, no. 1, pp. 30–37 (in Russ.).
15. Barybin A.K., Lobanov V.N., Cheldiev M.I., Chuchkalov P.B. Configurable computer platform with heterogeneous architecture. *Issues of Radio Electronics*. 2016, vol. 2, no. 7, pp. 70–77 (in Russ.).
16. Baranov L.D., Lobanov V.N., Cheldiev M.I. Use of multiprocessor computing platform with heterogeneous architecture for solving the problems of hydroacoustics and radiolocation. *Issues of Radio Electronics*. 2018, no. 5, pp. 7–16 (in Russ.).

Для цитирования

Костенко В.А. Использование алгоритмов оптимизации с самообучением для управления динамически изменяющимися системами // Программные продукты и системы. 2020. Т. 33. № 1. С. 020–026. DOI: 10.15827/0236-235X.129.020-026.

For citation

Kostenko V.A. Using self-learning optimization algorithms to manage dynamically changing systems. *Software & Systems*. 2020, vol. 33, no. 1, pp. 020–026 (in Russ.). DOI: 10.15827/0236-235X.129.020-026.