

УДК 156.6

РЕШЕНИЕ ЗАДАЧИ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ ПРИ ВЫПОЛНЕНИИ АДМИНИСТРАТИВНЫХ РЕГЛАМЕНТОВ

В. В. Науменко,

аспирант

В. В. Копытов,

доктор техн. наук, профессор

Северо-Кавказский федеральный университет, г. Ставрополь

Рассматриваются проблемы повышения эффективности системы государственного управления, одной из которых является распределение ресурсов в процессе исполнения государственных функций и предоставления государственных услуг. Для решения данной проблемы предлагается применение оптимизационной модели, направленной на эффективное распределение выполняемых работ между исполнителями. При этом используются методы теории массового обслуживания и алгоритмы поиска оптимального значения целевой функции.

Ключевые слова — административный регламент, функциональная безопасность, распределение ресурсов.

Введение

Современная нормативно-правовая база [1, 2] требует от органов власти исполнения государственных функций и предоставления государственных услуг строго в соответствии с требованиями административных регламентов (АР), делая тем самым АР ключевым элементом системы государственного управления, от эффективности выполнения которого зависит эффективность всей системы государственного управления. В настоящее время информационная система государственного управления является критической социотехнической системой, неэффективная работа которой связана с невозможностью выполнять функции, закрепленные нормативно-правовыми актами и законами, что ухудшает социально-экономическое развитие. Поэтому на функциональную безопасность информационной подсистемы системы государственного и муниципального управления напрямую влияет выполнимость АР, которая в свою очередь выражается в невыполнении требований АР.

Анализ структуры административных регламентов

Рассмотрим структуру АР. Выделяются следующие составляющие АР [3]:

— область применения, которая ограничивается набором соответствующих нормативных регуляторов;

- функциональные цели;
- субъекты выполнения регламента, в интересах которых осуществляется регламент и (или) которые охватываются функциональной целью;
- объекты действия регламента;
- процессы и операции регламента.

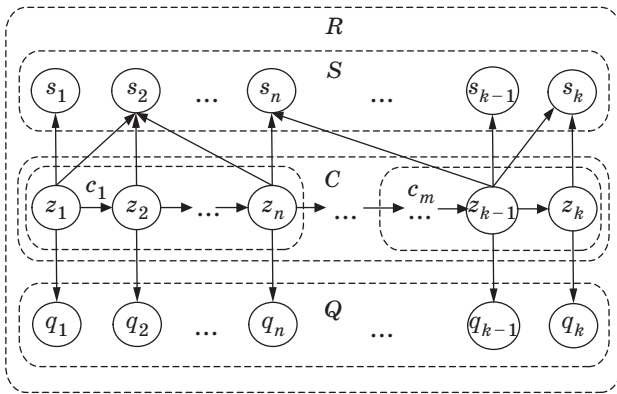
Главное назначение АР заключается в интеграции совокупности процессов и операций, реализуемых субъектами действия над существующими объектами в интересах, определенных нормативными регуляторами и инструкциями, для достижения заданной цели [3]. Поэтому если рассматривать АР как алгоритмический процесс, реализующий государственную услугу, то его основу будут составлять процессы и операции, закрепленные в нормативно-правовом акте АР (рис. 1). Типовая структура АР предполагает наличие описания административных процедур c , каждая из которых достигается путем выполнения определенной последовательности задач z :

$$P(c_m) \rightarrow P\left((z_n)_{n=1}^k\right), k > 0,$$

где P — предикат наличия элемента системы ($P(e)$ принимает значение «истина», если e существует в системе).

При этом сам АР R будет характеризоваться последовательностью процедур:

$$P(R) \rightarrow P\left((c_n)_{n=1}^k\right), k > 0.$$



■ Рис. 1. Типовая структура административного регламента

Каждая задача предполагает совершение определенных операций субъектов над объектами, т. е.

$$P(z_n) \rightarrow P(s_m), P(z_n) \rightarrow P(q_m).$$

Субъект-пользователь инициирует выполнение первой задачи последовательности, в результате чего запускается процесс выполнения АР. При этом субъекты и объекты выступают в качестве ресурсов, необходимых для перехода от одной задачи к другой [4].

Основные проблемы, препятствующие выполнению требований АР, можно разделить на две группы:

- связанные с логической противоречивостью описания процессов АР;
- возникающие в процессе выполнения АР.

Решение проблем логической противоречивости связано с правильным выбором методов и средств моделирования информационных процессов, подходящих для описания АР. Построение адекватной информационной модели АР позволит выявить противоречия в описании до его реализации в виде совокупности информационных процессов, исключая появление в системе ряда угроз функциональной безопасности. Однако проблемы, возникающие в процессе выполнения, являются следствием либо отсутствия, либо неправильного распределения ресурсов системы — субъектов и объектов и требуют решения оптимизационных задач. Решение одной из таких задач предлагается ниже.

Постановка задачи распределения ресурсов

Как уже было отмечено, выполнение некоторой задачи z_n связано с наличием в системе соответствующего субъекта s'_m и объекта q'_k , которые можно определить как необходимые требования для z_n . Однако стоит отметить, что система госу-

дарственного управления является динамической системой, и поэтому соблюдение условий $P(s'_m) \rightarrow P(s_m)$ и $P(q'_k) \rightarrow P(q_k)$ зависит от того, заняты ли s_m и q_k в момент времени t_i , где t_i — время начала выполнения задачи z_n :

$$\tau(t_i) \rightarrow (P(s'_m) \rightarrow P(s_m)) \wedge (P(q'_k) \rightarrow P(q_k)),$$

где $\tau(t_i)$ — предикат, принимающий значение «истина» в момент времени t_i .

При этом, как показывает практика, каждый объект АР уникален и соответствует только одной задаче. Поэтому решение задачи обеспечения функциональной безопасности выполнения АР состоит в согласовании множества объектов — необходимых Q' и существующих в системе Q : $Q' \subset Q$.

Иная ситуация обстоит с субъектами. Каждый субъект в системе государственного управления функционально может выполнять несколько задач АР. Поэтому для обеспечения функциональной безопасности выполнения АР необходимо решение оптимизационной задачи, направленной на эффективное распределение задач между субъектами-исполнителями.

Рассмотрим процедуру решения задачи оптимального распределения субъектов-исполнителей, входящих в группу S_o (с общим количеством субъектов-исполнителей a), где каждый из субъектов-исполнителей $s_n \in S_o$ функционально может быть задействован в выполнении задач из группы Z_l (с общим количеством задач b). Учитывая, что процесс выполнения задач непрерывный, а процесс поступления задач на выполнение — случайный, то АР можно рассматривать как сеть узлов массового обслуживания, где каждый узел — это субъект-исполнитель s_n , который представляет собой систему массового обслуживания типа $M/M/1$, выполняющую поток заявок λ_k , соответствующий задаче $z_k \in Z_l$. Каждая выполненная задача z_k в свою очередь инициирует выполнение следующей задачи z_{k+1} , поэтому, согласно теореме Бёрке, исходящий поток с входящим потоком с параметром λ и показательным распределением интенсивности обслуживания μ является пуассоновским потоком с тем же параметром λ [5]. Для определения интенсивности обслуживания задачи z_k необходимо в качестве исходных данных использовать среднее время выполнения для каждой из выполняемых задач отдельно взятой организации $t(z_k)$:

$$\mu_k = \frac{1}{t(z_k)}. \quad (1)$$

В случае если субъект s_n выполняет несколько задач, то интенсивность обслуживания отдельной задачи определяется с учетом всех входящих потоков, т. е.

$$\mu_k = \frac{p_k}{t(z_k)}, \quad (2)$$

где p_k — вероятность появления заявки из потока λ_k , определяемая отношением потока λ_k к сумме всех потоков, выполняемых субъектом:

$$p_k = \frac{\lambda_k}{\sum_{i=k}^{k+p} \lambda_i}. \quad (3)$$

Среднее время обслуживания для $M/M/1$ будет определяться следующим образом [5]:

$$T_k = \frac{1/\mu_k}{1 - \lambda_k/\mu_k}. \quad (4)$$

В случае если одну задачу (один поток) выполняет несколько субъектов, то система массового обслуживания будет $M/M/m$, где m — число субъектов. Среднее время обслуживания

$$T_k = \frac{(\rho_k m)^{m+1}}{\lambda_k (m-1)! \sum_{n=0}^m \left[\frac{(\rho_k m)^n}{n!} \right] \left[(m-L)^2 - L \right]}, \quad (5)$$

где L — число заявок, ожидающих в очереди; среднее значение определяется по формуле

$$L_q = \frac{\rho_k (m\rho_k)^m}{m!(1-\rho_k)^2} \frac{1}{\left(\sum_{n=0}^{m-1} \frac{(m\rho_k)^n}{n!} + \frac{(m\rho_k)^m}{m!(1-\rho_k)} \right)}, \quad (6)$$

где $\rho_k = \lambda_k / (m\mu_k)$.

Просчитав для каждой задачи среднее время выполнения T_k и сравнив его с максимальным временем $T_{k \max}$, определенным нормативными регуляторами АР, можно сказать, что если $T_k > T_{k \max}$, то задача не выполняется с заданными условиями, т. е. не соблюдаются требования функциональной безопасности. Результат представим в виде коэффициента

$$\alpha_k = \frac{T_k}{T_{k \max}}. \quad (7)$$

Данный показатель характеризует загруженность субъекта-исполнителя при выполнении задачи z_k и позволяет оценить запас времени, который доступен для решения других задач из группы Z_l .

Решение любых задач распределения ресурсов требует выбора определенного критерия в качестве целевой функции, на получение оптимальных значений которого и будет направлена задача.

Необходимо отметить, что государственное управление не имеет формализуемой целевой функции. С этим и связано отсутствие интегрального показателя качества государственного управления, на повышение которого могла быть направлена задача распределения ресурсов. Поэто-

му в качестве целевой функции необходимо использовать показатель, который характеризует, насколько сбалансировано распределение исполнителей между задачами АР. В качестве такого показателя целесообразно использовать дисперсию величины α_k каждой из задач АР:

$$D(\alpha) = \frac{1}{b} \sum_{k=1}^b (\alpha_k - M[\alpha_k])^2. \quad (8)$$

Таким образом, данная задача распределения ресурсов (субъектов между задачами АР) схожа с классической задачей распределения ресурсов (между поставщиками и потребителями) и состоит в определении величин x_{nk} , определяющих поставку n -субъекта k -задаче и принимающих значение 1 либо 0. При этом для каждой из задач АР выполняется условие $\alpha_k \leq 1$, а величина $D(\alpha)$ минимальна.

Анализ методов решения задачи распределения ресурсов

Стоит отметить, что данная задача имеет сложную целевую функцию, которая зависит не только от входных данных ($\lambda_k, t(z_k)$), но и от сложности взаимосвязей «субъект-задача» внутри системы. Поэтому для вычисления оптимального значения целевой функции необходимо отказаться от градиентных методов (методов, использующих значения градиентов или высших производных непрерывных функций, предназначенных для оценки шага и скорости приближения к точке оптимума в итерационных процессах) и применить один из безградиентных методов либо алгоритм открытого поиска. Подобные алгоритмы основаны на оценках критерия качества по множеству точек, расположенных вокруг текущей точки, и выборе одной из них, где критерий принимает наименьшее из всех оцениваемых значений [6].

Теперь рассмотрим процедуру распределения задач между субъектами-исполнителями. Приведем технологическую матрицу, характеризующую распределение ресурсов в системе (табл. 1). В левом столбце приведены все задачи группы Z_l , в верхней строке — все субъекты группы S_o .

■ Таблица 1. Матрица распределения ресурсов в системе

Z_l	S_o					
	s_1	s_2	...	s_n	...	s_a
$z_1 \lambda_1, t(z_1), T_{1\max}$	x_{11}	x_{21}	...	x_{n1}	...	x_{a1}
$z_2 \lambda_2, t(z_2), T_{2\max}$	x_{12}	x_{22}	...	x_{n2}	...	x_{a2}
...
$z_k \lambda_k, t(z_k), T_{k\max}$	x_{1k}	x_{2k}	...	x_{nk}	...	x_{ak}
...
$z_b \lambda_b, t(z_b), T_{b\max}$	x_{1b}	x_{2b}	...	x_{nb}	...	x_{ab}

Для каждой задачи указывается поток заявок λ_k и среднее время выполнения $t(z_k)$. При этом сумма каждой строки должна быть больше 0, так как для любой из задач должен быть определен исполнитель:

$$\begin{cases} x_{11} + x_{21} + \dots + x_{n1} + \dots + x_{a1} > 0; \\ x_{12} + x_{22} + \dots + x_{n2} + \dots + x_{a2} > 0; \\ \dots \\ x_{1k} + x_{2k} + \dots + x_{nk} + \dots + x_{ak} > 0; \\ \dots \\ x_{1b} + x_{2b} + \dots + x_{nb} + \dots + x_{ab} > 0. \end{cases} \quad (9)$$

Решение алгоритма открытого поиска заключается в поиске значения вектора $\mathbf{x} = [x_{11}, \dots, x_{ab}]$, которое обеспечивало бы минимальное значение целевой функции $D(\alpha)$ за 2^{ab} шагов (размещение с повторениями). На каждом шаге определяется соответствие системе неравенств (9). Если вектор \mathbf{x} не удовлетворяет условиям неравенства, то идет переход к следующему шагу, в противном случае вычисляется вектор $\mu = [\mu_1, \dots, \mu_b]$ по выражениям (1)–(3), далее вычисляется вектор $\mathbf{T} = [T_1, \dots, T_b]$ [выражения (4)–(6)], затем вектор $\alpha = [\alpha_1, \dots, \alpha_b]$ и проводится проверка на соответствие условию $\alpha_k \leq 1$ [выражение (7)]. Если любое из $\alpha_k > 1$, то идет переход к следующему шагу, иначе вычисляется значение $D(\alpha)$ [выражение (8)]. После этого выбирается значение вектора \mathbf{x} , соответствующее минимальному значению $D(\alpha)$:

$$\mathbf{x} = \arg \min (D_1(\alpha), \dots, D_{2^{ab}}(\alpha)). \quad (10)$$

Алгоритм открытого поиска позволяет получить все оптимальные значения вектора \mathbf{x} , однако из-за большого количества шагов решения данный метод является слишком трудоемким.

В работе [7] проведен подробный анализ безградиентных методов оптимизации для решения задачи подбора технических средств охраны. Рассматриваемая в работе задача, так же как и решаемая авторами данной статьи, имеет множество дискретных решений и сложную целевую функцию, в связи с чем был выбран генетический алгоритм (ГА).

Применение генетического алгоритма для решения задачи распределения ресурсов при выполнении АР

Рассмотрим применяемые в настоящее время различные модели ГА. Классическим (каноническим) ГА принято считать алгоритм Джона Холланда [8], который имеет следующие характеристики:

- целочисленное кодирование особей (хромосом);
- одинаковая длина всех хромосом в популяции;

- постоянный размер популяции;
- пропорциональный отбор;
- односточечный кроссовер;
- битовая мутация;
- формирование следующего поколения из потомков текущего поколения.

Помимо классического ГА существуют и другие модели: гибридный, Genitor, СНС ГА и др. Они различаются принципами отбора и формирования нового поколения особей, операторами мутации, кодированием генов и т. д. В Genitor-модели (созданной Д. Уитли [9]) используется специфичная стратегия отбора. Вначале производится оценка особей начальной популяции. Далее выбираются случайным образом две особи, которые скрещиваются, причем получается только один потомок, который оценивается и занимает место наименее приспособленной особи. После этого снова случайным образом выбираются две особи, и их потомок занимает место особи с самой низкой приспособленностью. Таким образом, на каждом шаге в популяции обновляется только одна особь. В работе [10] утверждается, что при помощи Genitor-модели поиск гиперплоскостей происходит лучше, а сходимость быстрее, чем у классического ГА.

В СНС-модели (*Cross generational elitist selection, Heterogeneous recombination, Cataclysmic mutation*), характеризующейся отсутствием мутаций, используется популяция небольшого размера, отбор особей в следующее поколение ведется и между родительскими особями, и между их потомками. После нахождения некоторого решения алгоритм перезапускается, и лучшая особь копируется в новую популяцию, а оставшиеся особи являются сильной мутацией (мутирует примерно треть битов в хромосоме) существующих, и поиск повторяется. Другой специфичной чертой является стратегия скрещивания: все особи разбиваются на пары, причем скрещиваются только те пары, в которых хромосомы особей существенно различны. При скрещивании используется так называемый HUX-оператор (*Half Uniform Crossover*) — разновидность однородного кроссовера, но в нем к каждому потомку попадает ровно половина битов хромосомы от каждого родителя.

Использование гибридного алгоритма (*Hybrid Algorithms*) позволяет объединить преимущества ГА и классических методов. На каждом поколении каждый полученный потомок оптимизируется этим выбранным классическим методом, после чего продолжают обычные для ГА действия. Стоит отметить, что такой метод ухудшает способность алгоритма к поиску решения с помощью отбора гиперплоскостей, однако на практике гибридные алгоритмы показывают успешные результаты [11]. Это связано с тем, что обычно вели-

ка вероятность того, что одна из особей попадет в область глобального максимума и после проведенной оптимизации окажется решением задачи.

Параллельные ГА предназначены для снижения преждевременной сходимости к локальному оптимуму, стимуляции разнообразия и поиска альтернативных решений той же проблемы. Они основаны на разбиении популяции на несколько отдельных подпопуляций, каждая из которых будет обрабатываться ГА независимо от других. Кроме того, разнообразные миграции индивидов порождают обмен генетическим материалом среди популяций, которые обычно улучшают точность и эффективность алгоритма. Наиболее распространенными из параллельных ГА являются островная модель (*Island model*) и ячеистый алгоритм (*Cellular Genetic Algorithm*).

Несмотря на очевидные преимущества параллельных ГА, для решения поставленной задачи будем использовать классический алгоритм, так как в рамках данной работы необходимо проверить адекватность разработанной математической модели распределения ресурсов при выполнении АР, которая в свою очередь заключается в наличии сходимости целевой функции. Получение приемлемых результатов с помощью алгоритма Холланда позволяет применять и другие модели ГА для решения задачи распределения ресурсов АР.

Этапы и пример решения задачи распределения ресурсов при выполнении АР

Теперь рассмотрим каждый из этапов решения задачи. Вначале случайным образом выбираются N ($N \geq b$) значений вектора x (начальная популяция из N особей), каждое из которых соответствует системе неравенств (9) и условию $\alpha_k > 1$. Далее генерируется промежуточная популяция — набор особей, получивших право размножаться. Для генерации промежуточной популяции используется принцип пропорционального отбора, заключающийся в том, что каждая особь попадает в популяцию с вероятностью, пропорциональной ее приспособленности. Для данной задачи чем меньше значение $D(\alpha)$, тем больше вероятность попадания в промежуточную популяцию:

$$P_p = 1 - \frac{D_p(\alpha)}{D_1(\alpha) + \dots + D_N(\alpha)}. \quad (11)$$

Далее особи в случайном порядке разбиваются на пары, и производится скрещивание (обмен случайными отсеченными частями):

$$x_p[01011.01\dots101] \leftrightarrow x_{p+1}[11001.11\dots001] = x'_p[11001.01\dots101].$$

К полученному в результате отбора и скрещивания новому поколению применяется оператор мутации, который инвертирует каждый бит популяции с вероятностью $1/N$. Далее из полученных особей выбираются только те, что соответствуют системе неравенств (9) и условию $\alpha_k > 1$, после чего цикл повторяется снова. Процесс эволюции (цикл отбор—скрещивание—мутация) может продолжаться бесконечное число шагов, поэтому критерием останова является получение сходимости целевой функции за n число шагов. При этом оптимальному значению вектора x будет соответствовать наиболее приспособленная особь из последнего поколения:

$$x = \arg \min \left((D_1(\alpha), \dots, D_N(\alpha))_n \right). \quad (12)$$

В случае если ни на одном из шагов не обеспечивается выполнение любого из условий, то распределение ресурсов в системе невозможно ввиду их недостаточности, и решение данной проблемы возможно только путем перераспределения функциональных обязанностей, расширяя при этом группу S_o .

Приведем пример работы ГА с использованием скрипта «ga.m», входящего в пакет среды MatLab, для распределения семи задач между пятью субъектами-исполнителями. В качестве исходных данных будем использовать значения λ_k , $t(z_k)$, $T_{k \max}$, удовлетворяющие условию $l/\mu < 1$:

$$\begin{aligned} \lambda_k &= [1; 2; 1; 0.5; 0.4; 2; 1]; \\ t(z_k) &= [0.2; 0.4; 0.1; 0.5; 0.05; 0.3; 0.8]; \\ T_{k \max} &= [0.8; 3; 1.2; 5.6; 1.9; 3.3; 3.5]. \end{aligned}$$

Для подсчета дисперсии величины α_k для среды MatLab написан специальный скрипт «ga_example.m», который также проверяет полученные промежуточные значения T_k на соответствие условию $\alpha_k > 1$, и в случае отсутствия такого соответствия присваивает целевой функции значение 100, в противном случае выводит значение дисперсии $D(\alpha)$.

В качестве начальной популяции использовалось значение вектора, при котором целевая функция отлична от 100:

$$\begin{aligned} x &= [10; 30; 30; 18; 30; 12; 7]; \\ D(\alpha) &= 0.082326984378338. \end{aligned}$$

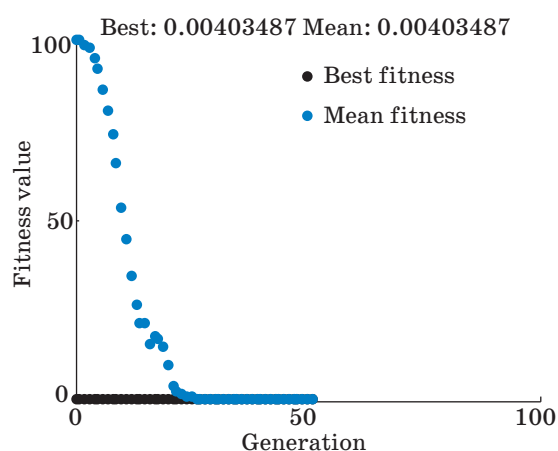
В результате расчета получены следующие численные значения:

Optimization running.
Objective function value: 0.004034869320015893
 $x = [10 \ 19 \ 2 \ 25 \ 2 \ 18 \ 7]$.

Полученное значение вектора x соответствует матрице (табл. 2).

■ **Таблица 2.** Результат распределения ресурсов в системе

Z_i	S_o				
	s_1	s_2	s_3	s_4	s_5
z_1 1, 0.2, 0.8	0	1	0	1	0
z_2 2, 0.4, 3	1	0	0	1	1
z_3 1, 0.1, 1.2	0	0	0	1	0
z_4 0.5, 0.5, 5.6	1	1	0	0	1
z_5 0.4, 0.05, 1.9	0	0	0	1	0
z_6 2, 0.3, 3.3	1	0	0	1	0
z_7 1, 0.8, 3.5	0	0	1	1	1



■ **Рис. 2.** Процесс оптимизации целевой функции при помощи ГА

Для наблюдения схождения целевой функции в среде MatLab использовался инструмент psearch-tool (рис. 2).

Из графика видно, что процесс сходится к установившемуся значению

$$D(\alpha) = 0.004034869320015893$$

приблизительно за 30 итераций.

Заключение

Внедрение данного метода в виде программного решения в автоматизированные системы исполнения государственных функций и предоставления государственных услуг позволит вывести систему государственного управления на качественно новый уровень путем непрерывного решения задач выполнимости AP.

Литература

1. **Федеральный закон** «Об организации предоставления государственных и муниципальных услуг» № 210-ФЗ от 27 июля 2010 года. <http://www.rg.ru/2010/07/30/gosusl-dok.html> (дата обращения: 01.03.2013).
2. **Постановление** Правительства РФ от 16.05.2011 № 373 «О разработке и утверждении административных регламентов исполнения государственных функций и административных регламентов предоставления государственных услуг». <http://www.rg.ru/2011/05/31/gosuslugi-site-dok.html> (дата обращения: 01.03.2013).
3. **Региональное электронное правительство: стратегия создания, архитектура, типовые решения / под ред. В. И. Дрожжинова, А. А. Лучина.** — М.: Экотрендз, 2004. — 288 с.
4. **Копытов В. В., Наumenко В. В., Минин В. А., Зайцев А. А.** Анализ проблем обеспечения бесконфликтного выполнения электронных административных регламентов // Сб. науч. ст. / Ставропольский филиал ИГУТИ. Ставрополь, 2012. Вып. XII. С. 72–78.
5. **Клейнрок Л.** Теория массового обслуживания / пер. с англ. И. И. Грушко; ред. В. И. Нейман. — М.: Машиностроение, 1979. — 432 с.
6. **Жерновков В. А., Дмитриенко Д.В.** Распределение ресурсов на основе алгоритмов открытого поиска // Журнал университета водных коммуникаций. 2009. № 3. С. 153–156.
7. **Давидюк Н. В.** Разработка системы поддержки принятия решений для обеспечения физической безопасности объектов [электронный ресурс]: дис. ... канд. техн. наук: 05.13.01, 05.13.19 — Астрахань: РГБ, 2010.
8. **Holland J. H.** Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. — Massachusetts Institute of Technology, 1992. — 328 p.
9. **Уитли Д.** Учебник по генетическим алгоритмам // Статистика и компьютеринг. 1994. № 4. С. 65–85.
10. **Syswerda G.** Schedule optimization using genetic algorithms // Handbook of genetic algorithms / ed. By L. Davis. — N. Y.: Van Nostrand Reinhold, 1991. P. 332–349.
11. **Газизов Р. К., Гагарин А. В.** Гибридный генетический нейросетевой алгоритм в задаче идентификации параметров цифровых моделей // Вестник УГАТУ. Уфа, 2009. Т. 13. № 2. С. 246–256.