7. **Filippova A. S., Valiakhmetova J. I.** *Optimal use of resources: cutting-packing problems*. Advances in Economics and Optimization: Collected Scientific Studies Dedicated to the Memory of L. V. Kantorovich. — NY: Nova Science Publishers, 2014, pp. 35—48.
8. **Kantorovich L. V., Zalgaller V. A.** *Racional'nyj raskroj promyshlennyh materialov* (Industrial materials rational cutting). SPb.: Nevskij Dialekt, 2012. 304 p.
9. **URL:** http://paginas.fe.up.pt/~esicup/tiki-index.php (access date: 30.09.15).
10. **Filippova A. S., Kuznetsov V. Yu.** Zadachi o minimal'nom pokrytii ortogonal'nyh mnogougol'nikov s zapretnymi uchastkami (Problems concerning minimal covering of orthogonal polygons containing prohibited zones), *Informacionnye tehnologii*, 2008, no. 9, pp. 60—65.
11. **Zak E.** A counterpart of one-dimensional stock cutting: skiving stock problem, *Proceedings of the sixteenth triennial conference of the International Federation of Operational Research Societies*, 2002, p. 121.
12. **Zavelin S. L., Zhegolko K. V., Frolovskij V. D.** Razrabotka i issledovanie geneticheskogo algoritma dlja avtomatizacii proektnyh procedur optimizacii geometricheskogo pokrytija (Development and research of a genetic algorithm for automation of geometrical covering optimization procedures), *Vestnik Tambovskogo gosudarstvennogo tehnicheskogo universiteta*, 2015, vol. 21, no, 2, pp. 257—265.
13. **Krivulja A. V., Zlotnik M. V., Romanova T. E.** Sredstva matematicheskogo modelirovanija v zadachah prjamougol'nogo pokrytija proizvol'nyh mnogougol'nyh oblastei (Mathematical modelling means in problems of rectangular covering of random polygon areas), *Radiojelektronika i informatika*. 2007, no. 4, pp. 34—40.
14. **Stojan Yu. G., Patsuk V. N.** Pokrytie mnogougol'noi oblasti minimal'nym kolichestvom odinakovyh krugov zadannogo radiusa (Polygon area covering with a minimal number of similar circles of given radius), *Dop. NAN Ukraini*, 2006, no. 3, pp. 74—77.
15. **Bortfeldt A.** A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. *European Journal of Operational Research*, 2006, vol. 172 (3), pp. 814—837.
16. **Muhacheva A. S., Chiglintsev A. V., Smagin M. A., Muhacheva E. A.** Zadachi dvumernoj upakovki: razvitie geneticheskih algoritmov na baze smeshannyh procedur lokal'nogo poiska optimal'nogo reshenija (Two-dimensional packing problems: genetic algorithms development based on mixed procedures of optimal solving local search), *Informacionnye tehnologii*, 2001, no. 9, Prilozhenie, 28 p.
17. **Kenmochi M., Imamichi T., Nonobe K., Yagiura M., Nagamochi H.** Extract algorithms for the two-dimensional strip packing problem with and without rotations, *European Journal of Operational Research*, 2009, vol. 198 (1), pp. 73—83.
18. **Lesh N., Mitzenmacher M.** Bubble Search: A simple heuristic for improving priority-based greedy algorithms, *Information Processing Letters*, 2006, vol. 97 (4), pp. 161—169.
19. **Muhacheva E. A., Valiahmetova Yu. I., Telitskij S. V., Hasanova E. I.** Proektirovanie razmeshhenija ortogonal'nyh ob''ektov na poligonah s prepjatstvijami (Designing of orthogonal objects location on polygons with some obstacles), *Informacionnye tehnologii*, 2010, no. 10, pp. 16—22.
20. **Filippova A. S., Telitskij S. V., Porechnyj S. S.** *Optimizacija komgleksnogo processa geometricheskogo pokrytija i raskroja* (Optimization of the geometrical cutting-packing complex process) / Saarbrucken, Germany: LAP LAMBERT Academic Publishing GmbH & Co. KG, 2013. 175 p.
21. **Telitskij S. V., Valiahmetova Yu. I., Hasanova E. I.** Gibridnyj algoritm na osnove posledovatel'nogo utochnenija ocenok dlja zadach maksimal'nogo ortogonal'nogo pokrytija, *Vestnik Bashkirskogo universiteta*, 2012, vol. 17, no. 1 (I), pp. 421—425.
22. **Muhacheva E. A., Filippova A. S., Chiglintsev A. V.** Zadacha prjamougol'noj upakovki v polubeskonechnuju polosu: chislennyj jeksperiment s bezothodnymi zadachami E. Hopper na baze algoritmov blochnoj struktury (The problem of rectangular packing into a semi-endless strip: numerical experiment with the wasteless problems), *Informacionnye tehnologii*, 2005, no. 7, pp. 23—32.

УДК 519.857

**В. И. Струченков,** д-р техн. наук, проф., e-mail: str1942@mail.ru
Московский государственный университет
информационных технологий, радиотехники и электроники (МИРЭА)

# Динамическое программирование с использованием множеств Парето в задачах планирования реализации возобновляемых ресурсов

*Настоящая статья является продолжением статьи, опубликованной в № 2 за 2016*[1]*. В ней рассматривается задача составления оптимального поэтапного плана использования возобновляемого однородного ресурса в течение заданного времени. В данной статье предложен эффективный алгоритм динамического программирования с использованием множеств Парето.*

***Ключевые слова:*** *ресурс, целевая функция, множество состояний, динамическое программирование, множества Парето*

## Введение

В статье [1] рассмотрены две математические модели задачи планирования реализации возобновляемого ресурса. В рамках этих моделей применительно к использованию метода динамического программирования получены рекуррентные соотношения, позволяющие строить оптимальный план без анализа пошаговых вариантов. В вычислительном плане это во много раз эффективнее, чем классический алгоритм динамического программирования, так как практически поиск решения сводится к вычислению трех величин на каждом шаге процесса по простым формулам. Однако эти модели не претендуют на универсальность, так

[1] Струченков В. И. Динамическое программирование в задачах планирования реализации частично возобновляемых ресурсов // Информационные технологии. 2016. № 2.

как возможна различная постановка рассматриваемой задачи. В общем случае модели, рассмотренные в [1], могут потребовать уточнений, которые не позволят получить рекуррентные формулы для построения оптималького плана. Поэтому рассматриваемая задача продолжает оставаться актуальной.

Цель настоящей статьи состоит в анализе задачи построения оптимального плана реализации возобновляемого ресурса в общем случае и разработке алгоритма ее решения по методу динамического программирования с использованием множеств Парето для повышения вычислительной эффективности алгоритмов [2, 3].

## Постановка задачи

Задача составления поэтапного плана реализации возобновляемого ресурса состоит в следующем [1, 4]. Заданы:

- период планирования $T$ (например, $T = 10$ лет);
- начальное количество возобновляемого ресурса $b_1$;
- коэффициент воспроизводства $p$;
- функция годового дохода $d(x)$ при реализации $x$ единиц ресурса;
- функция сопутствующих затрат в течение года $c(x, b)$, где $b$ — количество ресурса на начало года.

Требуется построить такой план реализации ресурса, чтобы в течение $T$ лет суммарный количественный показатель качества плана (целевая функция) принял максимальное значение.

Будем считать, что годы пронумерованы, начиная с единицы, так что последний год имеет номер $T$.

Если в начале любого года $t$ ($t = 1, 2, ..., T$) имеется $b_t$ единиц ресурса и в течение этого года реализуется $x_t$ его единиц, то на начало следующего года будет $b_{t+1} = p(b_t - x_t)$ единиц ресурса. Коэффициент воспроизводства $p$ не изменяется в течение рассматриваемого периода планирования.

Если целевая функция — это суммарная прибыль, то задача формализуется в следующем виде: найти вектор $\mathbf{x}(x_1, x_2, ..., x_T)$, при котором достигает максимума

$$\sum_{t=1}^{T} (d(x_t) - c(x_t, b_t)) \text{ и } 0 \leqslant x_t \leqslant b_t;\ b_{t+1} = p(b_t - x_t).$$

Выражение $d(x_t) - c(x_t, b_t)$ только условно можно назвать годовой прибылью, так как реально могут быть еще и сопутствующие затраты, не зависящие от количества реализуемого ресурса. Теоретически наличие в целевой функции постоянного слагаемого несущественно, так как оно не влияет на точку экстремума, т. е. на искомое решение. При анализе математических моделей и построении рекуррентных соотношений для расчета оптимального плана в работе [1] наличие постоянной составляющей затрат на реализацию ресурса не учитывалось. Однако формальный оптимум может соответствовать решению, при котором в каком-либо году $d(x_t) - c(x_t, b_t))$ меньше постоянной составляющей годовых затрат на реализацию ресурса, поэтому годовая прибыль отрицательна, хотя суммарно за все годы прибыль максимальна. Если отрицательная годовая прибыль недопустима, то на каждом этапе появляется дополнительное ограничение на минимальную прибыль и тем самым на минимальное количество реализации ресурса. Это ограничение усложняет алгоритм поиска.

## Классическая схема динамического программирования

Для решения задачи применительно к построению оптимального плана вылова форели в работе [4] предлагается классическая схема динамического программирования. Ключевое для метода динамического программирования понятие "состояние системы" формализуется как количество имеющегося ресурса. Соответственно "траектория" (или "путь") — это последовательность состояний, т. е. значений имеющегося ресурса $b_t$ в начале каждого года с номером $t$ ($t = 1, 2, ..., T$). В классическом алгоритме динамического программирования [5, 6] процесс рассматривается от "конца к началу", т. е. начиная с множества состояний $b_T = b_1p^{T-1} + 1$ в начале последнего года. При реальных $b_1$, $p$ и $T$ значение $b_T$ может быть велико, поэтому усовершенствование классического алгоритма динамического программирования актуально с точки зрения объема вычислений.

Отметим, что в данной задаче при ее численном решении удобно и более эффективно двигаться от начала к концу. Конкретный вид функций $d(x)$ — доход и $c(x, b)$ — затраты существенного значения не имеет, если есть возможность их вычисления при соответствующих дискретных значениях $x_t$ и $b_t$.

Далее будем рассматривать задачу в общем виде, двигаясь от начального состояния $b_1$. Дополнительное ограничение на минимальную годовую прибыль означает, что к последнему году многие из состояний последнего года, о которых речь шла выше (от 0 до $b_1p^{T-1} + 1$), оказываются недостижимыми, но заранее вычислить, какие именно состояния недостижимы, сложно. Дело в том, что сопутствующие затраты могут зависеть не только от количества ресурса $x_t$, реализуемого в году $t$, но и от количества имеющегося в начале года ресурса $b_t$. Действительно, например, применительно к разведению форели выловить 100 форелей легче, если их много (скажем 10 000), чем если их всего 200 и должно остаться не менее 100.

Однако это не единственное преимущество выбора направления построения оптимального пути от начального состояния в конечные, которых окажется много меньше, чем $b_1p^{T-1} + 1$.

Алгоритм при движении от начала к концу состоит в следующем.

1. Начиная с первого года, имея $b_1$ единиц ресурса, рассматриваем все возможные количества его реализации $0 \leqslant x_1 \leqslant b_1$. Вместо нуля может быть заданное минимальное количество реализуемого ресурса $x_{\min}$, что сокращает объем вычислений. Для каждого из возможных значений $x_1$ вычисляем и запоминаем соответствующие значения целевой функции и состояние с $b_2 = (b_1 - x_1)p$. Переходы с недопустимо малой прибылью отбраковываются.

Тем самым завершается формирование множества состояний после первого года (этапа).

2. Аналогично формируем состояния каждого из последующих этапов, вычисляя суммарные значения целевой функции и оставшиеся количества ресурса. Во избежание полного перебора при достижении одного состояния разными путями в соответствии с принципом оптимальности Р. Беллмана [5, 6] оставляем только путь, по которому это состояние достигается с бóльшим значением целевой функции, и запоминаем соответствующее ему состояние предыдущего этапа (связь).

3. Завершив последний этап, получаем оптимальное значение целевой функции и обратным разворотом по цепочке связей, начиная с оптимального конечного состояния ($b_T + 1$), восстанавливаем оптимальную последовательность состояний и соответствующие величины реализации ресурса на каждом этапе, т. е. в каждом году.

При наличии дополнительных требований, например, не оставлять ресурса меньше заданного числа, легко скорректировать построение множества состояний на каждом этапе.

## Использование множеств Парето

В данной задаче возможно существенное сокращение объема вычислений при использовании более совершенной реализации этого классического алгоритма динамического программирования за счет использования множеств Парето [2, 3].

Предположим, что зависимости $d(x)$ и $c(x, b)$ заданы более сложными формулами, чем линейная и квадратичная модели, рассмотренные ранее [1], таблично или еще каким-то образом. Важно, что мы можем вычислить прибыль за год при известных $x$ и $b$, т. е. для любого заданного состояния ($b$) и перехода к новому состоянию. Зададимся вопросом, может ли при различных планах реализации ресурса через несколько лет от начала планирования получиться так, что для одного состояния ресурса осталось больше, чем для второго, и к тому же суммарная прибыль получена больше, чем для второго? Если это так, то очевидно, что второе состояние бесперспективно, и его вообще не надо рассматривать при переходе к следующему этапу, так как наличие лишнего ресурса никак не может помешать принятию тех же решений при дальнейших переходах из первого состояния, что и при дальнейшем продвижении из второго состояния. Получается, что второе состояние "отстало навсегда".

Речь идет о наличии доминируемых (непаретовских) точек в множестве пар чисел (координат точек на плоскости), из которых первое число — это количество оставшегося ресурса, а второе число — это полученная за все предыдущие годы суммарная прибыль [2, 3]. Множество недоминируемых точек называется множеством Парето [2]. Именно эти точки и соответствующие им состояния имеет смысл оставлять на каждом этапе для дальнейшего рассмотрения, исключая все доминируемые точки (состояния) как бесперспективные. Характерно, что для различных функций $d(x)$ и $c(x, b)$ доминируемые точки могут появляться уже после первого этапа, когда из начального состояния ($b_1$) исходят несколько путей и никакой отбраковки по принципу Р. Беллмана путей, приводящих в одну точку, еще нет.

Будем считать, что при заданном $b_1$ принимаемое на первом шаге решение $x_1'$ доминирует над другим таким решением $x_1''$, если соответствующая ему точка с координатами $(b_2', z_1(x_1'))$ доминирует над $(b_2'', z_1(x_1''))$, т. е. $b_2' \geqslant b_2''$ и $z_1(x_1') \geqslant z_1(x_1'')$). Здесь $b_2'$ — количество ресурса на начало второго года, $z_1(x_1')$ — соответственно прибыль, полученная при реализации $x_1'$ единиц ресурса. Аналогичный смысл имеют величины, отмеченные двумя штрихами для другого значения $x_1$ (рис. 1).

Если прибыль $z(x) = d(x) - c(x, b)$ не является монотонно неубывающей функцией $x$, то точки, соответствующие участкам убывания ($x_1 > x_1^*$ на рис. 1) являются доминируемыми, и соответствующие значения $x$ можно не рассматривать ни на одном из этапов процесса построения оптимального плана.

Однако и для монотонно возрастающей функции $z(x) = d(x) - c(x, b)$ появление доминируемых

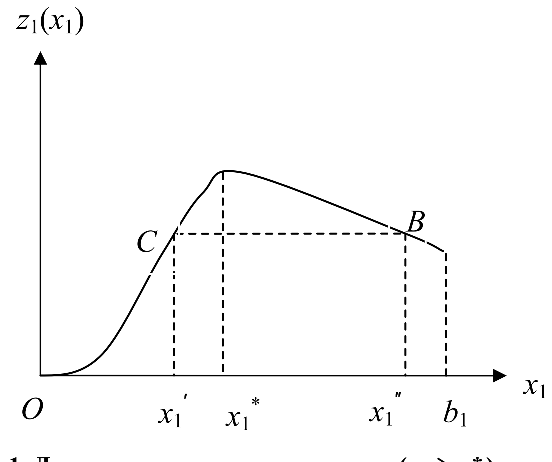


**Рис. 1. Доминируемые точки первого шага ($x_1 > x_1^*$)**

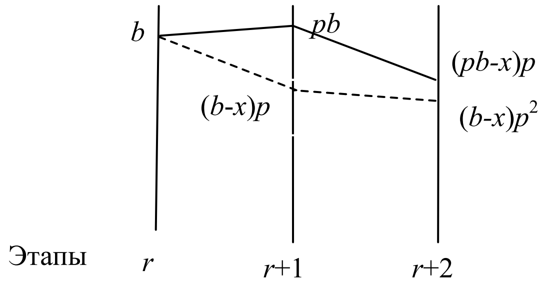


**Рис. 2. Пример появления доминируемых состояний**

точек (соответственно состояний на различных этапах) также вполне обычно.

В общем случае можно предположить, что $c(x, b)$ является возрастающей функцией $x$ при фиксированном $b$ и убывающей функцией $b$ при фиксированном $x$. Рассмотрим некоторое состояние $b$ на произвольном этапе $r$ и переходы на два следующих этапа из этого состояния, но по разным вариантам. В каждом варианте на двух переходах реализуется суммарно одно и то же количество ресурса $x$, но в первом варианте при переходе на $(r + 1)$-й этап нет реализации ресурса (сплошная линия на рис. 2), а во втором варианте нет реализации ресурса при переходе от $(r + 1)$-го к $(r + 2)$-му этапу (штриховая линия на рис. 2).

Очевидно, что $(pb - x)p > (b - x)p^2$ при $p > 1$.

Далее, доход $d(x)$ по этим двум вариантам один и тот же, так как он зависит только от количества реализованного ресурса. Поэтому надо сравнить суммарные за два этапа затраты. Полагая, что при отсутствии реализации ресурса затраты равны нулю, получаем по первому варианту $c_1 = c(x, pb)$, а по второму варианту $c_2 = c(x, b)$, так как по первому варианту реализация осуществляется при наличии ресурса $pb$, а по второму — при наличии ресурса $b$. Далее, $pb > b$ и поэтому $c_2 > c_1$, так как функция затрат $c(x, b)$ является убывающей по $b$ при заданном $x$.

Получается, что состоянию, достигаемому по первому варианту, соответствует бóльший ресурс и бóльшая прибыль, чем состоянию, достигаемому по второму варианту. Другими словами, во втором варианте получаем доминируемое состояние, которое следует исключить из дальнейшего рассмотрения.

Этот вывод остается в силе и при наличии постоянной составляющей затрат, не зависящей от x.

В реальных задачах возможны также дополнительные условия:

- нельзя допускать падение ресурса ниже заданного уровня;
- реализация ресурса планируется количествами, существенно бóльшими, чем единица.

При этих дополнительных условиях появление непаретовских состояний также возможно. Пусть, например, начальное состояние — 1000 единиц ресурса, коэффициент воспроизводства — 1,2, минимальный уровень — 300 единиц ресурса, дискрет использования ресурса — 100 единиц. Первый план состоит в том, что в первый и второй год ресурс не используется, а в третий год из имеющихся к этому году 1440 используются 1000 единиц и 440 остаются. Второй план состоит в том, что во второй год используются 900 единиц (больше нельзя) и далее в конце третьего года остается только 432 единицы ресурса. Это состояние (432) может далее не рассматриваться, так как состояние 440 (по первому плану) ему ни в чем не уступает, если получение прибыли от реализации 1000 единиц ресурса в третьем году предпочтительнее прибыли от реализации 900 единиц ресурса во втором году.

Для реализации динамического программирования с использованием множеств Парето нужно отказаться от ложного стереотипа необходимости разбиения регулярной сетки и решать задачу при движении от начального состояния в конечное. При анализе каждого перехода из каждого состояния нужно вычислить соответствующий ресурс, получаемый в результате такого перехода, и оценку нового состояния (суммарную прибыль от начала до нового состояния). Если в новое состояние приводит несколько путей, то надо оставить лучший из них (как в классической реализации метода Р. Беллмана), иначе нужно проверить, не существует ли таких состояний, которые и по ресурсу и по целевой функции не хуже полученного. Если такие состояния есть, то рассматриваемый переход исключается, если нет, то, наоборот, нужно проверить, не позволяет ли новое состояние исключить из рассмотрения уже имеющиеся. Другими словами, на каждом шаге должны оставаться только такие состояния, которые соответствуют множеству Парето для двухкритериальной задачи: максимум прибыли при максимуме оставшегося ресурса. Такие паретовские множества легко формировать, если на каждом шаге упорядочивать состояния по одному из критериев, например по оставшемуся ресурсу. Фактически, переходя к паретовским множествам, мы используем метод "погружения", т. е. замены исходной однокритериальной задачи более общей двухкритериальной задачей, для решения которой удается найти более эффективный алгоритм, который в итоге приводит и к решению исходной однокритериальной задачи. Действительно, после завершения последнего этапа мы имеем множество Парето, в котором точка с наибольшим значением первого критерия (прибыль) является решением исходной задачи. Однако наличие такого паретовского множества позволяет рассмотреть и субоптимальные решения, если требуется в итоге оставить не менее заданного количества ресурса и субоптимальное решение незначительно уступает оптимальному по целевой функции, но ему соответствует существенно больший остающийся ресурс

Расчеты показали, что такое динамическое программирование с использованием множеств Парето существенно эффективнее и по памяти и по объему вычислений, чем классическая реализация метода динамического программирования.

### Заключение

Для практического использования предложенных алгоритмов необходимо получение реальных данных из практики. Речь идет прежде всего о функциях дохода $d(x)$ и сопутствующих затрат $c(x, b)$, а также коэффициенте воспроизводства $p$, т. е. о величинах, которые в рассматриваемой постановке задачи являются исходными данными. Расчеты могут быть выполнены при различных значениях количества ресурса на начало периода планирования ($b_1$) для сопоставления различных планов реализации ресурса. Если функции $d(x)$ и $c(x, b)$ являются аддитивными, то возможность применения динамического программирования для составления оптимальных планов реализации возобновляемых ресурсов сомнений не вызывает. Эффективные в вычислительном плане реализации динамического программирования могут быть получены при использовании множеств Парето на каждом этапе построения оптимального плана.

Заметим, однако, что при выборе в качестве целевой функции экономической эффективности, вычисляемой как отношение суммарной прибыли к суммарным затратам, динамическое программирование применить не удается, так как эта функция не является аддитивной, т. е. не может быть представлена в виде суммы слагаемых, каждое из которых относится к соответствующему этапу. План, оптимальный в смысле экономической эффективности, вообще говоря, не совпадает с планом, дающим максимальную суммарную прибыль. Целесообразность такого плана с меньшей суммарной прибылью не очевидна. В любом случае решение, полученное по методу динамического программирования из условия максимума суммарной прибыли, может служить надежной оценкой в практических расчетах.

#### Список литературы


1. **Струченков В. И.** Динамическое программирование в задачах планирования реализации частично возобновляемых ресурсов // Информационные технологии. 2016. № 2.
2. **Струченков В. И.** Устаревшие стереотипы и новые алгоритмы решения прикладных задач дискретной оптимизации // Информационные технологии. 2012, № 5. С. 20—29.
3. **Struchenkov V. I.** Combined Algorithms of Optimal Resource Allocation // Applied Mathematics. Scientific Research. 2012. Vol. 3, no. 1.
4. **Косоруков О. А., Мищенко А. В.** Исследование операций: учебник для вузов. М.: Экзамен, 2003, 270 с.
5. **Беллман Р.** Динамическое программирование. М.: ИЛ, 1960.
6. **Беллман Р., Дрейфус С.** Прикладные задачи динамического программирования. М.: Наука, 1965. 458 с.



**V. I. Struchenkov,** Professor, e-mail: str1942@mail,ru,
Moscow State University of Information Technology, Radio Engineering and Electronics, Moscow, Russia.


## Dynamic Programming with Pareto Sets for Planning of the Renewable Resources Implementation


*Under study is the problem of optimal planning of the renewable resources implementation, such as the commercial breeding of fish, animals and so on.*

*The aim of this article is to study the oportunity of the optimal plan calculation using dynamic programming.*

*The predetermined scheduling period is divided into a number of stages, such as months or years.*

*At each step of the process of constructing an optimal plan the key concept of the Dynamic Programming "state of the system" is formalized as the number of available resources. Process goes from a given initial state to the final one. In accordance with the principle of optimality R. Bellman in achieving some state by different ways all of them are rejected, except the way that corresponds maximum profit. In addition, the state is rejected if on the same step there is a state with more resources and more profit. As a result, at each step Pareto set is formed. Such algorithm is more efficient than classic algorithm of Dynamic Programming.*



***Keywords:*** *resource, the objective function, the set of states, dynamic programming, the optimal path, Pareto sets*


#### References


1. **Struchenkov V. I.** Dinamicheskoe programmirovanie v zadachah planirovanija realizacii chastichno vozoblovljaenyh resursov, *Informacionnye tehnologii*, 2016, no 2, pp. 94—99 (in Russian).
2. **Struchenkov V. I.** Ustarevshie stereotipy i novye algoritmy reshenija prikladnyh zadach diskretnoi optimizacii, *Informacionnye tehnologii.* 2012, no. 5, pp. 20—29 (in Russian).
3. **Struchenkov V. I.** Combined Algorithms of Optimal Resource Allocation, *Applied Mathematics. Scientific Research*, 2012, vol. 3, no. 1.
4. **Kosorukov O. A., Mishenko A. V.** *Issledovanie operacij. Uchebnik dlja studentov vuzov*, Moscow: Examen, 2003, 270 p. (in Russian).
5. **Bellman R.** *Dinamicheskoe programmirovanie* (Dynamic Programming), Moscow: Foreign Literature, 1960, 234 p. (in Russian).
6. **Bellman R., Drejfus S.** *Prikladnye zadachi dinamicheskogo programmirovanija* (Applied Problems of Dynamic Programming), Moscow: Science, 1965, 458 p. in Russian).