2012. Вып. 2 (3). С. 3–8

МАТЕМАТИКА

УДК 512.554+517.530

ДВУМЕРНЫЕ ЭВОЛЮЦИОННЫЕ АЛГЕБРЫ: ИДЕМПОТЕНТНОСТЬ, РЕГУЛЯРНОСТЬ, АНАЛИТИЧНОСТЬ

С.Л. Блюмин

Аннотация

Идемпотентность, обратимость, регулярность элементов и аналитичность функций исследуются над общими, специальными и ненулевыми тривиальными двумерными эволюционными алгебрами при помощи элементарных методов, подходящих к использованию в учебном процессе.

Ключевые слова: двумерные эволюционные алгебры, идемпотентность, регулярность, аналитичность.

Широко известной двумерной алгеброй над полем действительных чисел **R**, используемой в математическом образовании, является алгебра – даже поле – комплексных чисел **C**. В различных областях прикладной математики используются алгебры двойных чисел **U** и дуальных чисел **V**, которые, как и **R, C**, коммутативны, ассоциативны, обладают нейтральными элементами, но не являются полями, так как не все их элементы обратимы, что приводит к вопросу об обобщённой обратимости, или регулярности по Дж. фон Нейману, тесно связанной с идемпотентностью. Эти алгебры рассмотрены с единых алгебраических позиций в [1] и с аналитических – в [2].

Определённым своеобразием в классе двумерных алгебр обладают эволюционные (генетические, менделевские) алгебры [3], возникшие (как видно из самого их названия) в связи с задачами эволюционной генетики (подробности об их возникновении и различных приложениях см. в [3]). Эти алгебры коммутативны, но, вообще говоря, не ассоциативны и не обладают нейтральными элементами. Следует отметить, что они не отражены в разделе справочника [4], посвященном неассоциативным кольцам и алгебрам.

Цель данной работы – исследовать идемпотентность, обратимость, регулярность элементов и аналитичность функций над общими, специальными и ненулевыми тривиальными двумерными эволюционными алгебрами при помощи элементарных методов, подходящих к использованию в учебном процессе образовательных учреждений различных статусов и уровней.

Ниже рассматриваются общие двумерные эволюционные алгебры $A(k,\ell,m,n)$ с параметрами k,ℓ,m,n и их частные случаи – специальные двумерные эволюционные алгебры $A(k,kt,m,mt)$ с параметрами *k, m, t* и ненулевые тривиальные двумерные эволюционные

алгебры $A(k,0,0,n)$ [3], определяемые соответствующими таблицами умножения их базисных элементов α и β:

$A(k,\ell,m,n)$		
	α	β
α	$k\alpha+\ell\beta$	0
β	0	$m\alpha+n\beta$

$A(k,kt,m,mt)$		
	α	β
α	$k(\alpha+t\beta)$	0
β	0	$m(\alpha+t\beta)$

$A(k,0,0,n)$		
	α	β
α	$k\alpha$	0
β	0	$n\beta$

Непосредственно проверяется, что эти алгебры коммутативны; в общем случае они не ассоциативны, но используемым в дальнейшем более слабым их свойством является эластичность [4]: $(ag)a=a(ga)$ для любых $a,g\in A$; ненулевые тривиальные двумерные эволюционные алгебры $A(k,0,0,n)$, и только они, обладают нейтральным элементом $\varepsilon=\frac{\alpha}{k}+\frac{\beta}{n}$. Специальные алгебры фактически являются одномерными, с единственным базисным элементом $\alpha+t\beta$.

Идемпотентность [4] элемента $a=x\alpha+y\beta\in A$, определяемая соотношением $a^2=a$, с учётом того, что

$$a^2=(x\alpha+y\beta)(x\alpha+y\beta)=x^2\alpha^2+y^2\beta^2=x^2(k\alpha+\ell\beta)+y^2(m\alpha+n\beta),$$

приводит к квадратичной системе уравнений относительно x, y

$$\begin{cases} kx^2+my^2=x \\ \ell x^2+ny^2=y \end{cases}.$$

Возведение второго уравнения в квадрат и исключение из него y с помощью первого уравнения

$$y^2=\frac{x-kx^2}{m}$$

приводит к уравнению относительно x:

$$x\left[\delta^2x^3-2n\delta x^2+(km+n^2)x-m\right]=0,$$

в котором использован детерминант алгебры $A(k,\ell,m,n)$

$$\delta=kn-\ell m.$$

Корень $x=0\Rightarrow y=0$ даёт стандартный идемпотент $a=0$, присущий и алгебрам **R, C, U, V,** тогда как второго стандартного идемпотента $a=1$ в произвольной алгебре $A(k,\ell,m,n)$ нет ввиду отсутствия в ней единицы.

Анализ остающегося кубического уравнения существенно упрощается в предположении

$$\delta=0\Rightarrow kn-\ell m=0\Rightarrow kn=\ell m\Rightarrow\frac{\ell}{k}=\frac{n}{m}=t\Rightarrow\ell=kt,n=mt,$$

что и приводит к специальной алгебре $A(k,kt,m,mt)$. В этой алгебре кубическое уравнение сводится к

$$(km+n^2)x-m=0 \Rightarrow x=\frac{m}{km+n^2} \Rightarrow y=\frac{n}{km+n^2},$$

что, при условии $km+n^2\neq 0$, определяет собственный идемпотент алгебры $A(k,kt,m,mt)$

$$a=\frac{m}{km+n^2}\alpha+\frac{n}{km+n^2}\beta=\frac{1}{k+mt^2}(\alpha+t\beta).$$

В ненулевой тривиальной алгебре $A(k,0,0,n)$, наряду со стандартными a=0 и $a=\varepsilon$, единицей этой алгебры, определяются ещё два собственных идемпотента:

$$\begin{cases} kx^2=x \\ ny^2=y \end{cases} \Rightarrow \begin{cases} x=0, x=\dfrac{1}{k} \\ y=0, y=\dfrac{1}{n} \end{cases} \Rightarrow a=0, a=\frac{\alpha}{k}, a=\frac{\beta}{n}, a=\varepsilon=\frac{\alpha}{k}+\frac{\beta}{n}.$$

Обратимость [4] элементов имеет смысл рассматривать только в алгебре $A(k,0,0,n)$, обладающей единицей, исходя из определения (здесь и далее $a=p\alpha+q\beta, a^{-1}=x\alpha+y\beta$)

$$aa^{-1}=\varepsilon \text{ или } (p\alpha+q\beta)(x\alpha+y\beta)=\frac{\alpha}{k}+\frac{\beta}{n}, kpx\alpha=\frac{\alpha}{k}, nqy\beta=\frac{\beta}{n},$$

откуда следует, что обратными

$$a^{-1}=\frac{\alpha}{pk^2}+\frac{\beta}{qn^2}$$

обладают те элементы $a\in A(k,0,0,n)$, у которых $p\neq 0$ и $q\neq 0$. Если же p=0 или q=0, то возникает вопрос о регулярности элементов.

Регулярность [4] элемента $a=p\alpha+q\beta\in A$, определяемая существованием элемента $g=a^{-}=x\alpha+y\beta\in A$, обобщённого обратного к a, такого, что $(ag)a=a(ga)=a$ (это определение в неассоциативной алгебре корректно благодаря её эластичности), в алгебре $A(k,0,0,n)$, в случае q=0, то есть $a=p\alpha$, приводит к уравнению

$$(p\alpha(x\alpha+y\beta))p\alpha=p\alpha, (kpx\alpha)p\alpha=p\alpha, k^2p^2x=p, x=\frac{1}{pk^2}, y \text{ любой},$$

так что $(p\alpha)^{-}=\frac{1}{pk^2}\alpha+c\beta$; аналогично $(q\beta)^{-}=c\alpha+\frac{1}{qn^2}\beta$; $0^{-}=c$.

В общей алгебре $A(k,\ell,m,n)$ условие регулярности, с учётом

$$ag=ga=(p\alpha+q\beta)(x\alpha+y\beta)=(pkx+qmy)\alpha+(p\ell x+qny)\beta,$$
$$(ag)a=a(ga)=(pkx+qmy)p(k\alpha+\ell\beta)+(p\ell x+qny)q(m\alpha+n\beta)=p\alpha+q\beta,$$

приводит к линейной системе уравнений относительно *x, y*

$$\begin{cases}(k^2p+\ell mq)px+(kp+nq)mqy=p\\(kp+nq)\ell px+(\ell mp+n^2q)qy=q\end{cases}.$$

Основной и вспомогательные определители этой системы

$$\Delta=(k^2p+\ell mq)(\ell mp+n^2q)pq-(kp+nq)^2\ell mpq=[pq(kn-\ell m)]^2=[pq\delta]^2,$$
$$\Delta_x=q[m(\ell p^2-nq^2)+pq(n^2-km)],\ \Delta_y=p[\ell(mq^2-kp^2)+pq(k^2-\ell n)]$$

позволяют определить a^- в случае $p, q, \delta\neq 0$. В случаях p=0 или q=0 обобщённые обратные определяются так же, как выше в алгебре $A(k,0,0,n)$.

В случае δ=0, то есть в алгебре $A(k,kt,m,mt)$, система уравнений

$$\begin{cases}(k^2p+ktmq)px+(kp+mtq)mqy=p\\(kp+mtq)\ell px+(ktmp+m^2t^2q)qy=q\end{cases}$$

при условии $q=tp$, то есть для элементов $a=p(\alpha+t\beta)$, сводится к одному уравнению $kpx+mpty=1/(k+mt^2)$, так что

$$((p(\alpha+t\beta))^-=c\alpha+\frac{1}{mt}\left(\frac{1}{p(k+mt^2)}-kc\right)\beta.$$

Абелева (строгая) регулярность [4] элемента $a=p\alpha+q\beta\in A$, определяемая существованием элемента $h=x\alpha+y\beta\in A$, такого, что $a^2h=a$, приводит к линейной системе уравнений относительно *x, y*

$$\begin{cases}k(kp^2+mq^2)x+m(\ell p^2+nq^2)y=p\\\ell(kp^2+mq^2)x+n(\ell p^2+nq^2)y=q\end{cases}.$$

Основной и вспомогательные определители этой системы

$$\Delta=(kp^2+mq^2)(\ell p^2+nq^2)\delta,$$
$$\Delta_x=(\ell p^2+nq^2)(np-mq),\ \Delta_y=(kp^2+mq^2)(kq-\ell p)$$

позволяют определить h в случае $p, q, \delta\neq 0$.

Случаи p=0 или q=0 отличаются от предыдущего тем, абелевы (строгие) обобщённые обратные определяются однозначно:

для $a=p\alpha$ $h=(n\alpha-k\beta)/kp\delta$; для $a=q\beta$ $h=(-n\alpha+k\beta)/nq\delta$.

В случае же δ=0, то есть в алгебре $A(k,kt,m,mt)$, абелевы обобщённые обратные определяются так же, как обобщённые обратные выше.

Следует отметить, что все полученные результаты допускают непосредственную проверку.

Аналитичность [5] функции $w = f(z)$, где $z = x\alpha + y\beta, w = u\alpha + v\beta$, представимой в виде $f(z) = u(x,y)\alpha + v(x,y)\beta = f(x,y)$, имеет место при условии существования её производной $f'_z(z) = r\alpha + s\beta$. Сравнение дифференциалов

$$df(x,y) = f'_x dx + f'_y dy = (u'_x\alpha + v'_x\beta)dx + (u'_y\alpha + v'_y\beta)dy =$$
$$= (u'_x dx + u'_y dy)\alpha + (v'_x dx + v'_y dy)\beta$$

и

$$df(z) = f'_z(z)dz = (r\alpha + s\beta)(dx\alpha + dy\beta) =$$
$$= (krdx + smdy)\alpha + (r\ell dx + sndy)\beta$$

приводит к соотношениям, которые служат аналогами условий Даламбера-Коши-Римана-Эйлера для двумерных эволюционных алгебр:

$$kr = u'_x, \quad \ell r = v'_x,$$
$$ms = u'_y, \quad ns = v'_y.$$

В случае $k, \ell, m, n \neq 0$ они записываются в виде, близком к стандартному:

$$\begin{cases} \dfrac{u'_x}{k} = \dfrac{v'_x}{\ell} \\ \dfrac{u'_y}{m} = \dfrac{v'_y}{n} \end{cases}.$$

Простейшей является функция $f(z) = z = x\alpha + y\beta$, для которой $u = x, v = y$, так что условия на величины *r* и *s*, $kr = u'_x = 1, \ell r = v'_x = 0$ и $ms = u'_y = 0, ns = v'_y = 1$ в общей алгебре $A(k,\ell,m,n)$ несовместны, то есть аргумент *z* не аналитичен; в алгебре же $A(k,0,0,n)$ эти условия дают $r = \dfrac{1}{k}, s = \dfrac{1}{n}$, так что $z' = \dfrac{\alpha}{k} + \dfrac{\beta}{n} = \varepsilon$ – единице этой алгебры, в согласии со стандартным свойством производной аргумента, тогда как в общей алгебре единицы нет.

Для функции $f(z) = z^2 = (x\alpha + y\beta)^2, u = kx^2 + my^2, v = \ell x^2 + ny^2$, условия выполняются в произвольной алгебре

$$kr = u'_x = 2kx, \ell r = v'_x = 2\ell x,$$
$$ms = u'_y = 2my, ns = v'_y = 2ny$$

и дают *r=2x, s=2y,* $(z^2)' = 2x\alpha + 2y\beta = 2z$ в согласии со стандартным свойством.

Для функции $f(z)=z^3=z^2z=((kx^2+my^2)\alpha+(\ell x^2+ny^2)\beta)(x\alpha+y\beta)$, для которой $u=k^2x^3+kmxy^2+\ell mx^2y+mny^3, v=k\ell x^3+\ell mxy^2+\ell nx^2y+n^2y^3$,

условия записываются в виде

$$kr=3k^2x^2+kmy^2+2\ell mxy, \ell r=3k\ell x^2+\ell my^2+2\ell nxy,$$
$$ms=2kmxy+\ell mx^2+3mny^2, ns=2\ell mxy+\ell nx^2+3n^2y^2$$

и оказываются совместными:

1) в алгебре $A(k,0,0,n)$, когда $r=3kx^2$, $s=3ny^2$, так что $(z^3)'=3kx^2\alpha+3ny^2\beta=3z^2$ и вообще в этой алгебре $(z^p)'=pz^{p-1}$;

2) в случае $k,\ell,m,n\neq 0$, когда они могут быть переписаны в виде

$$r=3kx^2+my^2+2\frac{\ell m}{k}xy, r=3kx^2+my^2+2nxy,$$
$$s=2kxy+\ell x^2+3ny^2, s=2\frac{\ell m}{n}xy+\ell x^2+3ny^2 \ -$$

при дополнительном условии $\ell m=kn$, то есть в алгебре $A(k,kt,m,mt)$, и при этом $(z^3)'=3z^2+2(tx-y)(my\alpha-kx\beta)=3z^2$ в случае $y=tx$, то есть $z=x\alpha+tx\beta=x(\alpha+t\beta)$, что и имеет место для элементов этой алгебры ввиду её фактической одномерности.

Таковы некоторые элементарные факты *теории функций эволюционного переменного*, которая может быть развита параллельно классической *теории функций комплексного переменного* [5] подобно тому, как в [2] представлены основы *теории функций квазикомплексного переменного*.

Благодарности.

Работа поддержана Российским фондом фундаментальных исследований, проект № 11-07-00580-а

Работа докладывалась на V Международной научно-практической конференции «Инновации и информационные технологии в образовании», апрель 2012 г., г. Липецк.

ЛИТЕРАТУРА

1. Блюмин, С.Л. Двумерные алгебры [Текст] / С.Л. Блюмин. – Липецк: ЛГТУ, 1995. – 18 с.

2. Блюмин, С.Л. Алгебраические основы комплексного анализа [Текст] / С.Л. Блюмин. – Липецк: ЛГТУ, 1996. – 18 с.

3. Tian, J. P. Evolution Algebras and their Applications [Text] / J. P. Tian. – NY: Springer, 2008. – 125 p.

4. Общая алгебра [Текст] / Под общей ред. Л.А. Скорнякова. – М.: Наука, 1990. – Т. 1. – 592 с.

5. Свешников, А.Г. Теория функций комплексного переменного [Текст] / А.Г. Свешников, А.Н. Тихонов. – М.: Наука, 2004. – 336 с.

2012. Вып. 2 (3). С. 9–13

УДК 519.5

A NOTE ON DIRECTLY COMPLETE ANALOGUE OF THE DE GROOT DUAL FOR GENERAL POSETS

Alena Chermikava, Martin Kovár

Abstract

In this paper we describe, for each general poset, a construction of a preframe (in the sense of B. Banaschewski) which is a certain analogue of the de Groot dual in topological spaces. Our dual satisfies a similar condition as $\tau^d \subseteq \tau^{ddd}$ *that was proved by the second author in 2001 for topological spaces.*

1 Introduction

In a chapter of *Open problems in Topology*, J. D. Lawson and M. Mislove stated a question whether a sequence of iterated de Groot duals of any topology terminates by two topologies, which are de Groot dual to each other [4]. The original formulation of the problem is motivated by some constructions of various semantic models in theoretical computer science and also by certain research in domain theory for which the dual and path topologies have a fundamental importance. The question remained unsolved for about a decade, when it was partially solved by the second author. To be more concrete, let us recall that for a topological space (X,τ), a topology τ^d, generated by the family of all compact saturated sets used as its closed base, is called the de Groot dual (or the co-compact topology) of the original topology τ. In 2001, the second author M. Kovбr proved that for any topology τ it holds $\tau^{dd} = \tau^{dddd}$ [2]. In 2004 he improved this result to its final and the most general form $\tau^d = (\tau \vee \tau^{dd})^d$ [3]. Note that from this last result it follows that $\tau^d \subseteq \tau^{ddd}$ for any topology τ, which finaly yields the equality $\tau^{dd} = \tau^{dddd}$.

However, the topological approach is not the only possible, and there are plenty of examples of mathematical and algebraic structures useful in computer science and domain theory, whose behavior and properties cannot be simply described by the classical topologies. Hence, it seems to be natural and interesting to study the original question of J. D. Lawson and M. Mislove also from a modified point of view. Beside the question of J. Lawson and M. Mislove, another source of inspiration for our considerations it is a notion introduced by B. Banaschewski in [1]. We say that a poset A is a *preframe* if A is closed under directed joins and finite meets (including the meet of the empty set), such that the binary meets distribute over the directed joins. By the usual definition, every directed set is non-empty, so the preframe need not have the least element $\bigvee\varnothing$ – the supremum of the empty set. On the other hand, a preframe always has the greatest element $\bigwedge\varnothing$. By $= \{\bot, \mathrm{T}\}$ we denote the *Sierpi*n′*ski frame*, consisting of the two elements, T and $\bot$. Let A be a set, then each mapping $f : A \to$ can be uniquely identified with its T-kernel, $\mathrm{Ker}_{\mathrm{T}} f = \{x \mid x \in A, f(x) = \mathrm{T}\}$. In this way, we can equip ^{A} with the partial order, given by the inclusion on the power set 2^A. By False and True we denote the constant functions on A identically equal to $\bot$ and T, respectively. Let A, B be posets. We say that a mapping $f : A \to B$ is a *morphism* if it preserves the (non-empty) directed joins and finite meets (including the meet of the empty set), whenever they exist. The set of all morphisms $f : A \to$ we denote by $\to$. We

consider it as a poset, naturally equipped with the order induced from the partially ordered set ^{A}. Throughout this paper, if not otherwise stated, by the dual of a poset A we mean the set $\to$ with this order.

2 Main Results

For our further considerations, we will need the following three lemmas:

Lemma 2.1 *Let A be a poset. Then the following conditions hold:*

1. $\to$ forms a preframe of all morphisms of A to .
2. For a directed set $Y \subseteq \to$, and $a \in A$, it holds $(\bigvee Y)(a) = \bigvee_{y \in Y} y(a)$.
3. For a non-empty finite set $Z \subseteq \to$, and $a \in A$, $(\bigwedge Z)(a) = \bigwedge_{z \in Z} z(a)$.
4. $\bigwedge \varnothing$, the top element of $\to$, is represented by the constant mapping $\mathrm{True}: A \to$, identically equal to T.

Proof. First, let us prove (ii). Let $Y \subseteq \to$ be non-empty and directed. Let $f(a) = \bigvee_{y \in Y} y(a)$ for every $a \in A$. We will show that $f = \bigvee Y$ in $\to$. First, we must prove that $f \in \to$. Let $B \subseteq A$ be non-empty and directed, such that $\bigvee B$ exists in A. Then $f(\bigvee B) = \bigvee_{y \in Y} y(\bigvee B) = \bigvee_{y \in Y} \bigvee_{b \in B} y(b) = \bigvee_{b \in B} \bigvee_{y \in Y} y(b) = \bigvee_{b \in B} f(b)$, so f preserves non-empty directed joins. Let $C \subseteq A$ be non-empty and finite. Suppose that $\bigwedge C$ exists in A. Then $f(\bigwedge C) = \bigvee_{y \in Y} y(\bigwedge C) = \bigvee_{y \in Y} \bigwedge_{c \in C} y(c) = \mathrm{T}$ implies that there exist some $y_1 \in Y$, such that for every $c \in C$ it follows $y_1(c) = \mathrm{T}$. Then $\mathrm{T} = \bigwedge_{c \in C} \bigvee_{y \in Y} y(c) = \bigwedge_{c \in C} f(c)$ which implies $f(\bigwedge C) \leq \bigwedge_{c \in C} f(c)$. Conversely, suppose that $\bigwedge_{c \in C} f(c) = \bigwedge_{c \in C} \bigvee_{y \in Y} y(c) = \mathrm{T}$. Then for every $c \in C$ there is some $y_c \in Y$ with $y_c(c) = \mathrm{T}$. Since Y is directed and C is finite, there exist some $y_1 \in Y$ such that $y_1 \geq y_c$ for every $c \in C$. Hence, for every $c \in C$ it follows $y_1(c) = \mathrm{T}$. Then $\mathrm{T} = \bigvee_{y \in Y} \bigwedge_{c \in C} y(c) = \bigvee_{y \in Y} y(\bigwedge C) = f(\bigwedge C)$ which implies that $f(\bigwedge C) \geq \bigwedge_{c \in C} f(c)$. Now we have $f(\bigwedge C) = \bigwedge_{c \in C} f(c)$, so f preserves also non-empty finite meets. It remains to check the preservation of the empty meet. Suppose that A has the greatest element $\bigwedge \varnothing \in A$. Then $f(\bigwedge \varnothing) = \bigvee_{y \in Y} y(\bigwedge \varnothing) = \bigvee_{y \in Y} \mathrm{T} = \mathrm{T}$. Hence, f is an element of $\to$, and, clearly, an upper bound of Y in $\to$. Now, let $u \in \to$ be another upper bound of Y. Then, for every $a \in A$ and every $y \in Y$ it follows that $u(a) \geq y(a)$, which gives $u(a) \geq \bigvee_{y \in Y} y(a) = f(a)$ and, consequently, $u \geq f$. So f is a correctly defined supremum of Y in $\to$.

Now, let us show (iii). Suppose that $Z \subseteq \to$ is non-empty and finite. Let $g(a) = \bigwedge_{z \in Z} z(a)$ for every $a \in A$. We will show that $g = \bigwedge Z$ in $\to$. First, we must show that $g \in \to$. Let $B \subseteq A$ be non-empty and directed, such that $\bigvee B$ exists in A. Then $g(\bigvee B) = \bigwedge_{z \in Z} z(\bigvee B) = \bigwedge_{z \in Z} \bigvee_{b \in B} z(b) = \mathrm{T}$ implies that for every $z \in Z$ there is $b_z \in B$ with $z(b_z) = \mathrm{T}$. Since Z is finite and B is directed, there is some $b_1 \in B$ such that $b_1 \geq b_z$ for every $z \in Z$. Then $z(b_1) = \mathrm{T}$ for every $z \in Z$, which implies that $\mathrm{T} = \bigvee_{b \in B} \bigwedge_{z \in Z} z(b) = \bigvee_{b \in B} g(b)$. Hence, $g(\bigvee B) \leq \bigvee_{b \in B} g(b)$. Conversely, suppose that $\bigvee_{b \in B} g(b) = \bigvee_{b \in B} \bigwedge_{z \in Z} z(b) = \mathrm{T}$. Then, there exists $b_1 \in B$, such that $z(b_1) = \mathrm{T}$ for every $z \in Z$. Then $\mathrm{T} = \bigwedge_{z \in Z} \bigvee_{b \in B} z(b) = \bigwedge_{z \in Z} z(\bigvee B) = g(\bigvee B)$. It follows that $g(\bigvee B) \geq \bigvee_{b \in B} g(b)$ and hence, together with the previously proved (converse)

inequality, we have $g(\bigvee B) = \bigvee_{b\in B} g(b)$. Now, let $C \subseteq A$ be non-empty and finite, having $\bigwedge C \in A$. Then $g(\bigwedge C) = \bigwedge_{z\in Z} z(\bigwedge C) = \bigwedge_{z\in Z}\bigwedge_{c\in C} z(c) = \bigwedge_{c\in C}\bigwedge_{z\in Z} z(c) = \bigwedge_{c\in C} g(c)$. Finally, suppose that A has the greatest element $\bigwedge\varnothing \in A$. Then $g(\bigwedge\varnothing) = \bigwedge_{z\in Z} z(\bigwedge\varnothing) = \bigwedge_{z\in Z} \mathrm{T} = \mathrm{T}$. It follows that g is an element of $\to$, and, clearly, a lower bound of Z in $\to$. Let $l \in \to$ be a lower bound of Z. Then, for every $a \in A$ and every $z \in Z$ we have $l(a) \le z(a)$, which gives $l(a) \le \bigwedge_{z\in Z} z(a) = g(a)$ and, consequently, $l \le g$. Therefore, g is a correctly defined infimum of Z in $\to$.

Regarding (iv), the mapping True, constantly equal to T, obviously preserves all non-empty directed joins and all finite meets, so $\to$ also has the greatest element $\wedge\varnothing$. Note that True does not preserve the empty join, but it is not required.

Finally, to show (i), it remains to check that binary meets distribute over directed joins in $\to$. Let $x \in \to$ and $Y \subseteq \to$ be directed. Then

$$(x \wedge (\bigvee Y))(a) = x(a) \wedge (\bigvee Y)(a) = x(a) \wedge (\bigvee_{y\in Y} y(a)) =$$
$$= \bigvee_{y\in Y} (x(a) \wedge y(a)) = \bigvee_{y\in Y} ((x \wedge y)(a)) = (\bigvee_{y\in Y} (x \wedge y))(a)$$

for every $a \in A$, which implies $x \wedge (\bigvee Y) = \bigvee_{y\in Y} (x \wedge y)$. By the definition, $\to$ is a preframe.

Let A be a poset. We denote by $h_A : A \to \to\to$ a mapping for which $(h_A(a))(x) = x(a)$ for every $x \in \to$. Further, for every $v \in \to\to\to$ and $a \in A$ we put $h_A^*(v) = v \circ h_A$. The announced lemma now follows:

Lemma 2.2 *Let A be a poset. Then $h_A : A \to \to\to$ is a morphism.*

Proof. Let $B \subseteq A$ be non-empty and directed and suppose that there exists $\bigvee B \in A$. Let $x \in \to$. Then $h_A(\bigvee B)(x) = x(\bigvee B) = \bigvee_{b\in B} x(b) = \bigvee_{b\in B} h_A(b)(x) = (\bigvee_{b\in B} h_A(b))(x)$, which implies that $h_A(\bigvee B) = \bigvee_{b\in B} h_A(b)$. Let $C \subseteq A$ be non-empty, finite and assume that there exists $\bigwedge C \in A$. Let $x \in \to$. It follows $h_A(\bigwedge C)(x) = x(\bigwedge C) = \bigwedge_{c\in C} x(c) = \bigwedge_{c\in C} h_A(c)(x) = (\bigwedge_{c\in C} h_A(c))(x)$, which implies that $h_A(\bigwedge C) = \bigwedge_{c\in C} h_A(c)$. Finally, suppose that there exists the greatest element $\bigwedge\varnothing \in A$. It follows that $h_A(\bigwedge\varnothing)(x) = x(\bigwedge\varnothing) = \mathrm{T}$ for every morphism $x \in \to$, so $h_A(\bigwedge\varnothing) = \mathrm{T}$. Now, since h_A preserves all non-empty directed joins and all finite meets, including the empty meet, it follows that h_A is a morphism.

Lemma 2.3 *Let A be a poset. Then $h_A^* : \to\to\to\to\to$ is a morphism.*

Proof. Let $v \in \to\to\to$. By the definition and the previous lemma, $h_A^*(v) = v \circ h_A$ is a morphism as a composition of two morphisms. So it is an element of $\to$, which means that h_A^* maps $\to\to\to$ into $\to$. It remains to show that it preserves the directed joins and the finite meets whenever they exist.

Let $V \subseteq \to\to\to$ be non-empty and directed. The poset $\to\to\to$ is a preframe by Lemma 2.1, so the supremum $\bigvee V$ exists. Denote $V' = \{v \circ h_A \mid v \in V\}$. The set V' is a non-empty directed subset of $\to$ and again, by Lemma 2.1, it follows that $\bigvee V'$ exists. Moreover, $\bigvee V' = \bigvee_{v\in V} v \circ h_A = \bigvee_{v\in V} h_A^*(v)$. We will show that $h_A^*(\bigvee V)$ and $\bigvee V'$ are the same maps. For any $a \in A$ it holds $h_A^*(\bigvee V)(a) = ((\bigvee V) \circ h_A)(a) = (\bigvee V)(h_A(a))$. Using the statement (ii) of Lemma 2.1,

we may continue: $(\bigvee V)(h_A(a)) = \bigvee_{v\in V} v(h_A(a)) = \bigvee_{v\in V}((v\circ h_A)(a)) = \bigvee_{w\in V'} w(a)$. Again, by the same statement (ii) of Lemma 2.1, we get $\bigvee_{w\in V'} w(a) = (\bigvee V')(a)$. Hence, it follows $h_A^*(\bigvee V)(a) = (\bigvee V')(a)$ for every $a\in A$ and so $h_A^*(\bigvee V) = \bigvee V' = \bigvee_{v\in V} h_A^*(v)$.

Let $C \subseteq \to\to\to$ be non-empty and finite. Since $\to..$ is a preframe, the infimum $\bigwedge C$ exists. Let $C' = \{c\circ h_A \mid c\in C\}$. The set C' is a finite subset of $\to$ and again, by Lemma 2.1, it follows that $\bigwedge C'$ exists. We have $\bigwedge C' = \bigwedge_{c\in C} c\circ h_A = \bigwedge_{c\in C} h_A^*(c)$. We will prove that $h_A^*(\bigwedge C)$ and $\bigwedge C'$ are the same maps. Let $a\in A$. Then $h_A^*(\bigwedge C)(a) = ((\bigwedge C)\circ h_A)(a) = (\bigwedge C)(h_A(a))$. Using the statement (iii) of Lemma 2.1, we may continue as follows: $(\bigwedge C)(h_A(a)) = \bigwedge_{c\in C} c(h_A(a)) = \bigwedge_{c\in C}((c\circ h_A)(a)) = \bigwedge_{d\in C'} d(a)$. Using the statement (iii) of Lemma 2.1 once more, we have $\bigwedge_{d\in C'} d(a) = (\bigwedge C')(a)$. Therefore, $h_A^*(\bigwedge C)(a) = (\bigwedge C')(a)$ for every $a\in A$ and so $h_A^*(\bigwedge C) = \bigwedge C' = \bigwedge_{c\in C} h_A^*(c)$.

To complete the proof, as the last step we need to prove the preservation of the empty meet. However, the empty meet $\bigwedge\varnothing$, considered as an element of $\to\to\to$, is a constant mapping, identically equal to T which takes arguments from $\to\to$. Composed with h_A, we get a mapping defined on A, which is constantly equal to T, that is, the top element True of $\to$.

As we will explain later, our main theorem is an analogue of the second author's result $\tau^d \subseteq \tau^{ddd}$, proven in 2001 for topological spaces.

Theorem 2.1 *Let A be a poset. Then*

$$h_A^* : \to\to\to\to\to$$

is a retraction.

Proof. We will show that $h_A^*\circ h_\to = \mathrm{id}\to$. Take $x\in\to$. Then
$(h_A^*\circ h_\to)(x) = h_A^*(h_\to(x)) = h_\to(x)\circ h_A$. Let $a\in A$. Then
$((h_A^*\circ h_\to)(x))(a) = (h_\to(x)\circ h_A)(a) = (h_\to(x))(h_A(a)) = (h_A(a))(x) = x(a)$. Now we have
$(h_A^*\circ h_\to)(x) = x$, which completes the proof.

Now, let us explain the analogy between Theorem 2.1 and the second author's result $\tau^d \subseteq \tau^{ddd}$, proven in 2001 for the de Groot dual in topological spaces. There are at least two common touch points with the classical problem formulated in topology.

Firstly, let (X,τ) be a topological space, $A\subseteq\tau$ a subbase for τ, ordered by the inclusion. A mapping $f : A\to$ is said to be a *compact function* if there exists a compact saturated set $K\subseteq X$ such that

$$f(U) = \begin{cases} \mathrm{T}, & \text{if} K\subseteq U \\ \bot, & \text{otherwise.} \end{cases}$$

The reader may easily check that each compact function $f:\tau\to$ is a morphism in our sense. Conversely, adjusting the usual Hofmann-Mislove Theorem, for a given preframe A, one can reconstruct (under some circumstances) the topological space such that each morphism $f : A\to$ is a

compact function. Thus the preframe $\to$, which is a dual of the poset A, corresponds with the de Groot dual.

Secondly, for a topological space (X,τ) with the family of all closed sets C, let us denote $f(S)=\mathrm{cl}^{ddd}(S)$, $g(S)=\mathrm{cl}^{d}(S)$ for every subset $S\in 2^X$. Then the closure operators $f:\mathsf{C}^d\to\mathsf{C}^{ddd}$, $g:\mathsf{C}^{ddd}\to\mathsf{C}^d$ satisfy $g\circ f=\mathrm{id}_{\mathsf{C}^d}$, so g is a retraction. The reader may easily check that this is equivalent to $\tau^d\subseteq\tau^{ddd}$ of [2] or [3]. Similarly, from Theorem 2.1 it follows that $h_{\to}:\to\to\to\to\to$ is a monomorphism, which makes the analogy with its topological counterpart $\tau^d\subseteq\tau^{ddd}$ obvious.

Работа была представлена на V Международной научно-практической конференции «Инновации и информационные технологии в образовании», апрель 2012 г., г. Липецк

REFERENCES

1. Banaschewski, B. *Another look at the localic Tychonoff theorem*, Comment. Math. Univ. Carolinae 29(4) (1988), 647-656.

2. Ковбг, M. M. *At most 4 topologies can arise from iterating the de Groot dual*, Topology and Appl. 130(1) (2003), 175-182.

3. Ковбг, M. M. *On iterated do Groot dualizations of topological spaces*, Topology and Appl. 146-147 (2005), 83-89.

4. Lawson J.D., Mislove M. *Problems in domain theory and topology*, Open problems in topology (van Mill J., Reed G. M. eds.), North-Holland, Amsterdam (1990), pp. 349-372.

2012. Вып. 2 (3). С. 14–21

УДК 519.5

REINFORCEMENT LEARNING VERSUS CLASSICAL MIXED-INTEGER LINEAR PROGRAMMING APPROACH FOR A REAL-LIFE BATCH PRODUCTION SCHEDULING PROBLEM

Yailen MARTINEZ, Dmitry BORODIN, Bert VAN VRECKEM, Wim DE BRUYN

Abstract

This paper provides a comparison of applying the reinforcement learning (RI) and classical Mixed Integer Linear Programming (MILP) approaches to the real-life complex hybrid flow-shop scheduling problem with limited-wait constraints. The case-study has been described in the literature, therefore, only the basic problem description and explanation of the used techniques are presented in the paper while the focus is on the results analysis. The results for the MILP formulation are derived using genetic algorithm (GA) and exact optimal solutions (though, not for the large problem instances) while the RI approach applies the Q-learning algorithm.

Introduction

In the paper we discuss two considerably different approaches for scheduling a chemical batch process, specifically the production of enzymes through fermentation, based on a real-life case [7]. Batch production scheduling problems have been actively studied: the recent reviews [17] and [19] provide various problem formulations and solution approaches, papers [14] and [21] give the overview of the recently developed optimization techniques for the short-term batch production scheduling and provide case studies.

The classical approach to solving scheduling problems consists of setting up a mathematical model, often based on a Mixed Integer Linear Programming formulation, and then applying a search algorithm like Branch and Bound to calculate the (guaranteed) optimal solution [14]. However, as the problem grows in the number of available resources, operations to be scheduled, and other constraints to be taken into account, this approach will no longer yield an optimal solution in a reasonable time span. In this situation, researchers often relax constraints that actually exist in the real-life problem. This makes the problem solvable, but it is no longer of practical use. As a consequence, only a fraction of published scheduling research deals with actual real-world problems [18].

As an alternative, approximative techniques are being explored where the guarantee of an optimal solution is abandoned in favor of quickly finding solutions that are “good enough”: heuristics, meta-heuristics, machine learning, etc. Although this may be frowned upon from the point of view of an optimization researcher, for a plant manager calculation speed and robustness to changes is of much greater importance than finding the theoretical optimum.

Often, the situation in a plant changes drastically (e.g. by incoming rush orders or machine breakdowns) so the previously calculated schedule is rendered useless and must be adapted. In this case, a suboptimal schedule with some slack usually needs fewer changes in order to make it feasible in the new situation.

The two approaches discussed in this paper are on the one hand a Genetic Algorithm (GA) [15] based on a classical MILP formulation of the scheduling problem, and on the other hand a Reinforcement Learning (RL) [12] algorithm applied on a light-weight simulation model of the plant.

The MILP problem formulation ideas used for the GA approach are similar to the ones explained in [2], [3], [4] and [20]. The idea of the problem reduction was also used in the mentioned papers.

Among various heuristic and meta-heuristic optimization techniques (e.g. [10]), GAs were widely and successfully used for solving scheduling problems, chemical batch production problems in particular [3], [4], [13], [21], [25]. The study of the proposed GA approach focuses on the GA itself, at the same time explaining the necessary problem formulation details and issues. Reinforcement Learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment [22].

It has been used in the solution of different scheduling problems, for example in [5], where the authors proposed the application of reinforcement learning techniques in the solution of the job shop scheduling, proving that interpreting it as a multi-agent learning problem is beneficial for obtaining near-optimal solutions, which can compete with alternative solution approaches.

The promising computational results help concluding that either of the proposed problem formulations and corresponding algorithm can be recommended for a generic class of problems similar to this case study; the choice of the approach depends on the goals of a certain problem. The GA provided solutions of the 0-20% better quality but it took 5-30 times longer comparing to the RI.

PRODUCTION PROCESS

The production unit under study is responsible for the production of various enzymes. The enzyme production must undergo the fermentation (3 stages: seed fermentation, main fermentation and broth preparation in the buffer tanks) and recovery (1 stage) processes.

The fermentation consists of multiplying an initially small amount of microorganisms and making them produce a desired enzyme. The first step is the seed fermentation: the microorganisms are placed in a small tank where they start growing and reproducing. After a limited time (typically 8 to 24 hours), the cells are transferred in a larger tank, a main fermenter, containing previously sterilized raw materials (nutrients) and water. During the main fermentation, the cells continue to grow and multiply, while producing the enzyme. The duration of this second step is significantly longer than the first one, typically 30 to 300h, depending on the type of enzyme considered.

To perform these operations, the case study factory possesses the following resources:

- three identical seed fermenters, $S = 2$;
- five identical main fermenters, $M = 5$;
- four buffer tanks: three identical small, $B^s = 3$, and one large, $B^l = 1$ ($B^s + B^l = 4$);
- one recovery line, $R = 1$.

A number of batches are to be processed over the planning horizon (which is in fact defined by customer demand). Several enzyme types are produced by the factory, denoted $p \in \{1 \dots P\}$. We have to produce α_p number of batches per each enzyme type, the size is given by q_p. Each batch is indexed by (p, j) consequently, where $p \in \{1 \dots P\}$ and $j \in \{1 \dots \alpha_p\}$.

Both seed and main fermentation phases include three operations: 1) the setup or preparation of the fermenters; 2) the fermentation itself (duration is denoted as π_p^S and π_p^M respectively); 3) the tear down or cleaning of the fermenters. So, the total time required for a batch to be processed either in a seed or in a main fermenter is given by sums of the times necessary to perform these three operations. Moreover, there are operational constraints: 1) a batch should be transferred without any delay to a main fermenter (which must be already setup or prepared) once the seed fermentation is finished (i.e. before starting the tear down or cleaning operations on the seed fermenter); 2) a batch should be transferred to the buffer tanks without any delay, once the main fermentation is finished (i.e. before starting the tear down or cleaning operations on the main fermenter); 3) the setup of a main fermenter requires a number of difficult manual operations, and because of the manpower restrictions at most one batch can be setup in main fermentation within a production shift.

When the fermentation is completed, the mixture of cells, nutrients and enzymes, called the broth, is transferred into a buffer tank where it is prepared for the recovery operations. Then the recovery takes place: it consists in separating the desired enzyme from the broth, using filtration and

purification. The duration of the recovery phase is denoted δ_p^R. During the recovery phase, a batch is progressively transferred from a buffer tank(s) to the filtration unit. That is why the buffer tank(s) in which the batch is stored remains occupied until the filtration of the entire batch is finished.

Figure 1 provides the graphical overview of the production stages.

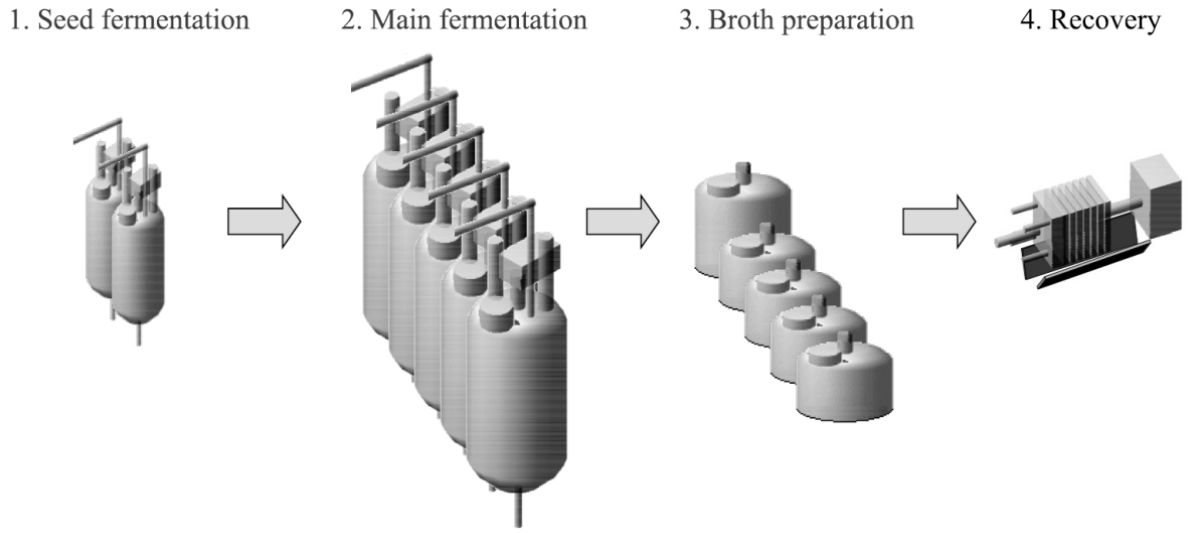

Figure 1. **Production process stages.**

GENETIC ALGORITHM AND REINFORCEMENT LEARNING

Genetic Algorithm

Genetic algorithm (GA) is a class of meta-heuristic algorithms, involving the population-based search technique with the "survival of the fittest" principle. After the initial set of solutions (called population) is generated (randomly or using a solution construction method), the series of steps (selection, crossover, mutation) is repeated consequently until a stopping criteria is satisfied [8]. The selection step is responsible for selecting some "good" solutions for the crossover and mutation. After the new solutions have been created, they are evaluated and selected to be included into the next population. A crossover operator generates new solutions from a number of already existing solutions (parent solutions or parents, usually only two parents are involved) hopefully taking good features from them, while mutation enhances the diversity and provides an opportunity to escape from the local optima.

GA is widely used for scheduling problems with both binary and permutation solution representation, providing good results (e.g. [3], [13], [17], [19], [21], [20], [23], [25]). GA was used for efficiently solving assignment problems as well. We propose the algorithm that combines the solution of permutation sequencing and two independent assignment problems but with one and the same objective function.

The implemented GA for two different solution encodings is described in [1] in details. Algorithm 1 proposes the GA. Both encodings SE_1 and SE_2 include subvectors (denoted as *SSV*) independent on each other.

Algorithm parameters:

- $SSV = (ssv_1; ssv_2; ssv_3; ssv_4; ssv_5)$ for encoding SE_1,
- $SSV = (ssv_1; ssv_2; ssv_3)$ for encoding SE_2,
- where $ssv_i = 1$ if subvector i is a affected on the iteration, 0 if its values are fixed,
- *Selection*, *Crossover* [11] and *Mutation* denote a selection strategy, crossover and mutation operators, respectively, chosen from the corresponding pools of the implemented genetic operators,
- *PM* is a probability of mutation to be applied on a particular solution,

- PP_i denotes a population of solutions on step i of size PP_{size},
- $Stop_L$ and $Stop_G$ denote local and global algorithm stopping conditions, respectively.

Algorithm 1. Genetic Algorithm.

Input:	b_{list}, $DD_(p,j)$
Step 0.	Choose SE_1 or SE_2.
	Set PP^{size}.
Step 1.	Generate initial population PP_0 of size PP^{size}.
Step 2_g.	Global iteration.
	REPEAT
	Set SSV.
Step $2_{g,l}$.	Local iteration.
	REPEAT
	Choose $Selection$, $Crossover$, $Mutation$.
	Perform Selection, Crossover, Mutation
	with probability PM.
	Set $PP_{g,l+1} = PP_{g,l}$, $l = l + 1$
	UNTIL $Stop_L = TRUE$
	Set $PP_{g+1} = PP_g$, $g = g + 1$
	UNTIL $Stop_G = TRUE$
Output:	A_0^M, A_0^B, $\sum_0^R$, $EF_{best}(A_0^M, A_0^B, \sum_0^R, b_{list}, DD_{p,j})$.

First, a solution encoding is chosen and a population size is specified. Second, an initial population is generated. Then, giving different values to the SSV vector we run global iterations until a global stopping condition is satisfied. Within the global iterations, for each value of SSV a number of local iterations is performed: they try various genetic operators for the chosen subvectors until a local stopping criterion is met. When a global stopping condition becomes true, the algorithm reports that if found a best possible solution and returns the corresponding complete schedule and the value of the objective function.

Reinforcement Learning Approach

Reinforcement Learning (RL) is a technique that allows an agent to learn how to maximize a numerical reward signal. The learner is not told which actions to take, but instead must discover which actions yield the most reward by trial-and-error. In the most interesting and challenging cases, actions may a affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These two characteristics, trial-and-error search and delayed reward, are the two most important distinguishing features of RL [22].

In the standard RL paradigm, an agent is connected to its environment via perception and action, as depicted in Figure 2. In each step of interaction, the agent senses the current state s of its environment, and then selects an action a which may change this state. The action generates a reinforcement signal r, which is received by the agent. The task of the agent is to learn a policy for choosing actions in each state so that the maximal long-run cumulative reward is received.

One important challenge in RL is the trade-off between exploration and exploitation. To obtain a high reward, a RL agent must prefer actions that it has tried in the past and found to be effective in producing reward. But in order to discover such actions, it has to try actions that it has not selected before. This means that the agent has to exploit what it already knows in order to obtain reward, but it also has to explore in order to make better action selections in the future. The dilemma is that neither exploration nor exploitation can be pursued exclusively without failing at the task. Therefore the agent must sample the available actions sufficiently and progressively favor those that appear to be best.

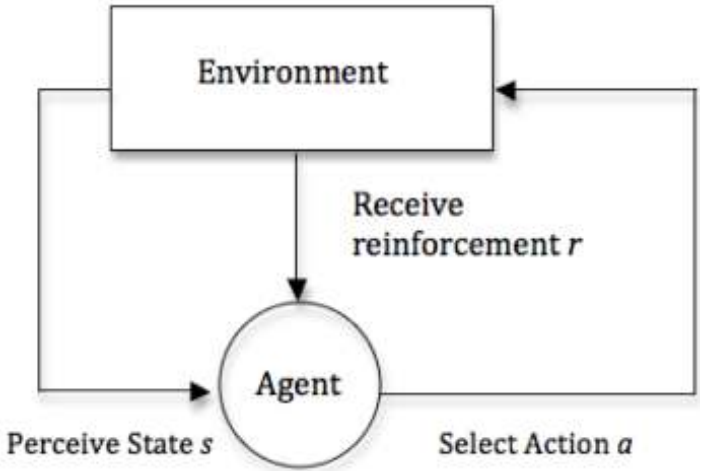

Figure 2. The Reinforcement Learning Paradigm.

Q-Learning

A well-known reinforcement learning algorithm is Q-Learning [24], which works by learning an action-value function that expresses the expected utility (i.e. cumulative reward) of taking a given action in a given state.

The core of the algorithm is a simple value iteration update, each state-action pair (*s, a*) has a Q-value associated. When action *a* is selected by the agent located in state *s*, the Q-value for that state-action pair is updated based on the reward received when selecting that action and the best Q-value for the subsequent state s_0. The update rule for the state action pair (*s, a*) is the following:

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma max_{a'}(Q(s',a')) - Q(s,a)]$$

In this expression, $\alpha \in \{0,1\}$ is the learning rate and *r* the reward or penalty resulting from taking action *a* in state *s*. The learning rate determines the degree by which the old value is updated. For example, if the learning rate is 0, then nothing is updated at all. If, on the other hand, $\alpha = 1$, then the old value is replaced by the new estimate. Usually a small value is chosen for the learning rate, for example, $\alpha = 0{,}1$. The discount factor (parameter γ) has a range value of 0 to 1 ($\gamma \in \{0,1\}$). If γ is closer to zero, the agent will tend to consider only immediate reward. If γ is closer to one, the agent will consider future reward with greater weight.

Dispatching Rules

Dispatching rules are among the more frequently applied heuristics in the solution of scheduling problems, mainly due to their ease of implementation and low time complexity. A dispatching rule is a sequencing strategy by which a priority is assigned to each job waiting to be executed on a specific machine.

Whenever a machine is available, a priority-based dispatching rule inspects the waiting jobs and the one with the highest priority is selected to be processed next [16]. Some of the most used dispatching rules are:

– Shortest Processing Time (SPT): The highest priority is given to the waiting operation with the shortest processing time.

– First In First Out (FIFO): The operation that arrived to the queue first receives the highest priority.

– Most Work Remaining (MWKR): Highest priority is given to the operation belonging to the job with the most total processing time remaining to be done.

– Earliest Due Date (EDD): The job due out first is processed first.

There are also some *composite* dispatching rules (CDR), which combine single dispatching rules and results have shown that a careful combination can perform better in terms of quality.

Learning Approach

The proposed learning approach is based on a Q-Learning algorithm. Each resource (i.e. fermenters, tanks and recovery line) has an agent associated, which will be responsible of deciding the next batch to process on its corresponding resource. This means that the set of actions is equivalent to the set of batches that are ready to be processed on the phase to which the agent belongs.

At the beginning of the process there is a list of orders that have to be executed, each of them with its corresponding due date. According to the characteristics of the learning algorithm, to choose an action, each agent takes into account the q-values associated to the set of possible actions it can execute. These q-values are arbitrarily initialized, and they are updated according to the last equation above.

The order in which the batches are selected during the initial phase is very important, because in that order they will gradually move to the next phases of the process, that is why, for example, we have to analyze if it is better to choose first the batch with the most work remaining, or the one with the earliest due date, etc. The rewards will be defined depending on the selected criteria. For example, if we decide that the best action is the batch with the earliest due date, then the reward will be the $\frac{1}{DueDate}$ of the batch.

It is also important to analyze whether it is better to use different criteria in the different phases or not. For example, if the broth preparation phase has several tanks, there can be several batches waiting for the recovery line at the same time, and maybe it will be better, in terms of tardiness, to choose the batch with the earliest due date, while in other phase it could be better to take into account another criterion.

According to the epsilon greedy policy, in order to balance exploration and exploitation, the agent has a small probability of selecting an action at random and a higher probability of selecting the best action, in this case the one with the highest q-value associated. Once a batch is finished on a specific resource, it becomes available for the next phase of the process.

NUMERICAL RESULTS

In this section, we present the numerical results of the computational experiments for 11 problem instances involving 10, 15, 20 and 30 batches. The results are compared with the optimal solutions (presented in [6], [9] and calculated using the CPLEX solver). The results of the proposed GA and RL approaches are compared and discussed.

The GA was coded in C# using the Microsoft Visual Studio 2005 .NET environment, using the same evaluation function program as in [6]. The computations were performed on a PC equipped with an Intel Core 2 duo 2.4 Mhz processor and 4 Gb of memory. The learning algorithm was implemented in Java, and the experiments were performed on a computer with the same characteristics that were mentioned before. The reason to choose Java was the existence of some previous work on the visualization and the interface.

Instance	CPLEX best	GA best	GA time	GA SE	QL best	QL time
N10_1	90	93	5	SE_1,SE_2	98	0.7
N10_2	30	30	17	SE_1,SE_2	37	1.5
N10_3	42	44	6	SE_2	44	0.9
N10_4	49	50	5	SE_2	50	1.3
N10_5	43	45	12	SE_2	58	1.5
N15_1	73	76	25	SE_2	84	3.3
N15_2	43	45	34	SE_2	56	3.6
N15_3	57	66	75	SE_2	75	3.4
N20_1	52	54	180	SE_2	61	5.4
N20_2	58	64	194	SE_2	75	5.4
N30_1	-	186	1560	SE_2	210	6.5

Table 1. Overall result comparison table.

Table 1 represents the overall comparison of the results. The *CPLEX best* column contains the optimal solutions, *GA best* – the GA results, computed in *GA time*. *GA SE* denotes the solution encoding used in the GA. Columns *QL best* and *QL time* show the results corresponding to the Q-Learning algorithm.

As the objective of both approaches is to minimize the tardiness, the columns showing the best solutions are reporting the weighted tardiness for each of the instances. The columns reporting the computational times show the time (in seconds) it took the algorithm to find the corresponding solution.

From table 1 it is possible to see that the GA outperforms the QL in most cases. Also it is obvious that the second solution representation SE_2 provides the solutions of better quality. It is important to notice that the solutions of the learning algorithm are close in quality to the best found, but the computational time is always significantly lower.

Conclusion

The paper presented two entirely different approaches for solving the same real-life case-study. Each approach showed its own benefits.

The GA represented a classical approach – a MILP model for the problem was formulated, decomposed into the IP and LP problems and solutions of high quality were obtained, at the same time providing tight schedules. A drawback of this classical approach is its low robustness – indeed, the schedule that corresponds to a high-quality solution is as tight as possible and not ready to be adapted in case an unexpected event occurs. In this case, a new schedule that takes into account the occurred events must be computed, which means that the problem will be solved from scratch.

The Reinforcement Learning approach comes with aid to handle this issue. Its main goal is to quickly compute a robust schedule. For instance, in real situations it often happens that a planner needs a schedule in a few seconds to report to the customer if their new order could be processed by this or that due-date.

The GA contributed with the hybrid use of the various genetic operators chosen in a random way. This concept is in line with the stochastic character of GAs. Besides, the proposed GA efficiently handled a combination of sequencing and assignment problems with one objective function, which is not typical for GAs.

Computational results proved that the GA is able to obtain very good solutions (comparing to the exact solutions), whereas the learning algorithm was able to obtain feasible solutions close to the optimal ones, and it reported significantly lower computational time. The RL approach also incorporated some extra features in order to deal with uncertainty and to provide more robust schedules.

In overall, this case study demonstrated that a real-life complex hybrid flow-shop scheduling problem with limited-wait constraints, which is difficult to solve by exact methods in the standard MILP formulation, can be efficiently solved by a meta-heuristic technique – on the one hand; and formulated in a lighter way to be suitable for obtaining more robust schedules faster and easier by an RL method – on the other.

REFERENCES

1. Borodin, D., Van Vreckem, B., De Bruyn, W.: Case study: an effective genetic algorithm for the chemical batch production scheduling. In: Proceedings of the 5th Multidisciplinary International Scheduling Conference: Theory & Applications (MISTA 2011), p. 14. Phoenix, Arizona (2011)

2. Chu, P., Beasley, J.: A genetic algorithm for the generalised assignment problem. Computers & Operations Research 24(1), pp. 17-23 (1997)

3. Davis, L.: Job-shop scheduling with genetic algorithms. In: Proceedings of the International Conference on Genetic Algorithms and Their Applications, pp. 136-140 (1985)

4. Engell, S., Мдrkert, A., Sand, G., Schultz, R.: Aggregated scheduling of a multiproduct batch plant by two-stage stochastic integer programming. Optimization and Engineering 5(3), 335-359 (2004)

5. Gabel, T., Riedmiller, M.: On a successful application of multi-agent reinforcement learning to operations research benchmarks. IEEE Transactions on Systems, Man, and Cybernetics (2009)

6. Gicquel, C., Hege, L., Minoux, M.: Mathematical models and solution approaches for the chemical batch scheduling problem (2010)

7. Gicquel, C., Hege, L., Minoux, M., Van Canneyt, W.: A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints. Computers & Operations Research 39, 629-636 (2012)

8. Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, MA (1989)

9. Hege, L., Gicquel, C., Minoux, M.: MILP models and solution approaches for scheduling a chemical batch process. In: Proceedings of 24th European Conference on Operational Research, p. 297 (2010)

10. Jans, R., Degraeve, Z.: Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. European Journal of Operational Research 177(3), 1855-1875 (2007)

11. Jong, K.A., Spears, W.: A formal analysis of the role of multi-point crossover in genetic algorithms. Annals of Mathematics and Artificial Intelligence 5(1), 1-26 (1992)

12. Kaelbling, L.P., Littman, M., Moore, A.W.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 237-285 (1996)

13. Kimms, A.: A genetic algorithm for multi-level, multi-machine lot sizing and scheduling. Computers & Operations Research 26(8), 829-848 (1999)

14. Mendez, C., Cerda, J., Grossmann, I., Harjunkoski, I., Fahl, M.: State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers & Chemical Engineering 30(6-7), 913-946 (2006)

15. Moraglio, A.: Towards a Geometric Unification of Evolutionary Algorithms. Ph.D. thesis, Department of Computer Science, University of Essex (2007)

16. Nhu Binh, H., Joc Cing, T.: Evolving Dispatching Rules for solving the Flexible Job-Shop Problem. In: IEEE Congress on Evolutionary Computation (CEC2005), vol. 3, pp. 2848-2855 (2005)

17. Ribas, I., Leisten, R., Framiтan, J.M.: Review and classification of hybrid flow-shop scheduling problems from a production system and a solutions procedure perspective. Computers & Operations Research 37(8), 1439-1454 (2010)

18. Ruiz, R., Юerifoplu, F., Urlings, T.: Modeling realistic hybrid flexible flowshop scheduling problems. Computers & Operations Research 35(4), 1151-1175 (2008)

19. Ruiz, R., Vбzquez-Rodrнguez, J.A.: The hybrid flow shop scheduling problem. European Journal of Operational Research 205(1), 1-18 (2010)

20. Sand, G., Engell, S.: Modeling and solving real-time scheduling problems by stochastic integer programming. Computers & Chemical Engineering 28(6-7), 1087-1103 (2004)

21. Sand, G., Till, J., Tometzki, T., Urselmann, M., Engell, S., Emmerich, M.: Engineered versus standard evolutionary algorithms: A case study in batch scheduling with recourse. Computers & Chemical Engineering 32(11), 2706-2722 (2008)

22. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press (1998)

23. Wang, L., Zhang, L., Zheng, D.: An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. Computers & Operations Research 33(10), 2960-2971 (2006)

24. Watkins, C., Dayan, P.: Technical note – q-learning. Machine learning 8(3), 279-292 (1992)

25. Xie, J.: Heuristic genetic algorithms for general capacitated lot-sizing problems. Computers & Mathematics with Applications 44(1-2), 263-276 (2002)

Работа была представлена на V Международной научно-практической конференции «Инновации и информационные технологии в образовании», апрель 2012 г., г. Липецк

2012. Вып. 2 (3). С. 22–25

УДК 517.968

ОБ УРАВНЕНИЯХ ВОЛЬТЕРРА С ЧАСТНЫМИ ИНТЕГРАЛАМИ В ПРОСТРАНСТВАХ СИММЕТРИЧНЫХ И КОСОСИММЕТРИЧНЫХ ФУНКЦИЙ

А.С. Калитвин, В.А. Калитвин

Аннотация

Получены условия однозначной разрешимости линейных уравнений Вольтерра с частными интегралами в пространствах симметричных и кососимметричных на квадрате функций.

Введение

Данная заметка является естественным продолжением работы [1] и содержит условия однозначной разрешимости линейных уравнений Вольтерра с частными интегралами в пространствах симметричных и кососимметричных на квадрате $D=[a,b]\times[a,b]$ функций; здесь симметричность и кососимметричность определенной на D функции $x(t,s)$ означает, что $x(t,s)=x(s,t)$ и $x(t,s)=-x(s,t)$ соответственно.

Очевидно, что определенная на D функция единственным образом представима в виде суммы симметричной и кососимметричной на D функций и справедливо равенство $x(t,s)=x_+(t,s)+x_-(t,s),$ где

$$x_+(t,s)=\frac{x(t,s)+x(s,t)}{2},\ x_-(t,s)=\frac{x(t,s)-x(s,t)}{2}$$

являются симметричной и, соответственно, кососимметричной функциями. Поэтому множество функций, определенных на квадрате $D,$ может быть представлено в виде прямой суммы $X_+\oplus X_-$, где X_+ — множество симметричных на D функций из X, а X_- — множество кососимметричных на D функций из X.

Пусть $X=X(D)$ — банахово пространство функций, определенных на D. Непосредственно проверяется, что $X_+=X_+(D)$ и $X_-=X_-(D)$ являются банаховыми пространствами относительно нормы, индуцируемой в X_+ и X_- нормой пространства X.

Далее в качестве X будем выбирать пространство $C(D)$-непрерывных на D функций и пространство $L^p=L^p(D)(1\le p\le\infty)$.

Через K обозначим линейный оператор Вольтерра с частными интегралами вида

$$(Kx)(t,s)=\int_a^t l(t,s,\tau)x(\tau,s)d\tau+\int_a^s m(t,s,\sigma)x(t,\sigma)d\sigma+\int_a^t\int_a^s n(t,s,\tau,\sigma)x(\tau,\sigma)d\tau d\sigma, \tag{1}$$

где l,m,n — заданные измеримые на $D\times[a,b],D\times[a,b],D\times D$ соответственно функции, а интегралы понимаются в смысле Лебега.

1. $X = C(D)$

В этом случае $X_+ = X_+(D)$ — пространство непрерывных и симметричных на D функций, $X_- = X_-(D)$ — пространство непрерывных и кососимметричных на D функций. Оба эти пространства банаховы относительно нормы, индуцируемой нормой пространства $X = C(D)$:

$$\|x\|_{X_+} = \|x\|_X = \max_{(t,s)\in D} |x(t,s)| \,(x \in X_+), \|x\|_{X_-} = \|x\|_X = \max_{(t,s)\in D} |x(t,s)| \,(x \in X_-),$$

причем $X = X_+ \oplus X_-$.

С применением методов из [1-5] доказывается

Теорема 1. Если оператор K действует в X_+ (в X_-) или действует из X_+ (из X_-) в X, то он непрерывен, а его норма вычисляется по формуле

$$\|K\| = \sup_D \left(\int_a^t (|l(t,s,\tau)| + |m(t,s,\tau)|)d\tau + \int_a^t\int_a^s |n(t,s,\tau,\sigma)|\, d\tau d\sigma\right). \qquad (2)$$

В книгах [4, 6] содержатся критерии действия оператора K в $X = C(D)$. Критерии действия оператора K в $X_+ = C_+(D)$, в $X_- = C_-(D)$, из $X_+ = C_+(D)$ и из $X_- = C_-(D)$ в $X = C(D)$ неизвестны.

Достаточные условия действия оператора K в X_+ и в X_- содержит приводимая ниже теорема 2, доказательство которой аналогично доказательству теоремы 2 из [1].

Через $C(L^1(\Omega))$, где $\Omega = [a,b]$ или $\Omega = D$, обозначим множество непрерывных на D вектор-функций со значениями в $L^1(\Omega)$.

Теорема 2. Пусть $l, m \in C(L^1([a,b])), n \in C(L^1(D))$ и $l(t,s,\tau) = m(s,t,\tau)$.

Тогда справедливы утверждения:

а) если $n(t,s,\tau,\sigma) = n(s,t,\tau,\sigma)$, то оператор K действует в X_+;

б) если $n(t,s,\tau,\sigma) = -n(s,t,\tau,\sigma)$, то оператор K действует в X_-.

Через $r(K), r_+(K), r_-(K)$ обозначим спектральный радиус оператора K, рассматриваемого в пространствах X, X_+, X_- соответственно.

В условии теоремы 2 $r(K) = 0$ [2, 4, 6]. Отсюда, формулы Гельфанда для спектрального радиуса ограниченного в банаховом пространстве линейного оператора и равенства (2) вытекает

Теорема 3. Пусть $l, m \in C(L^1([a,b])), n \in C(L^1(D))$ и $l(t,s,\tau) = m(s,t,\tau)$.

Тогда справедливы утверждения:

а) если $n(t,s,\tau,\sigma) = n(s,t,\tau,\sigma)$, то $r_+(K) = 0$;

б) если $n(t,s,\tau,\sigma) = -n(s,t,\tau,\sigma)$, то $r_-(K) = 0$.

Непосредственным следствием теоремы 3 является однозначная разрешимость в X_+ (в X_-) следующего интегрального уравнения Вольтерра с частными интегралами:

$$x = Kx + f,\ f \in X_+ \ (f \in X_-). \qquad (3)$$

Теорема 4. Пусть $l, m \in C(L^1([a,b])), n \in C(L^1(D))$ и $l(t,s,\tau) = m(s,t,\tau)$.

Тогда справедливы утверждения:

а) если $n(t,s,\tau,\sigma) = n(s,t,\tau,\sigma)$, то уравнение (3) имеет единственное решение в X_+ при любой функции $f \in X_+$;

б) если $n(t,s,\tau,\sigma) = -n(s,t,\tau,\sigma)$, то уравнение (3) имеет единственное решение в X_- при любой функции $f \in X_-$.

Разнообразные условия выполнения равенства $r(K) = 0$ содержатся в [4, 6]. Этими условиями можно заменить в теоремах 3 и 4 включения $l, m \in C(L^1([a,b])), n \in C(L^1(D))$, причем утверждения этих теорем не изменятся. В частности, утверждения теорем 3 и 4 останутся в силе, если эти включения заменить предположением о действии в X операторов

$$(Lx)(t,s) = \int_a^t l(t,s,\tau)x(\tau,s)d\tau, (Mx)(t,s) = \int_a^s m(t,s,\sigma)x(t,\sigma)d\sigma,$$
$$(Nx)(t,s) = \int_a^t \int_a^s n(t,s,\tau,\sigma)x(\tau,\sigma)d\tau d\sigma \qquad (4)$$

и выполнением условий

$$\int_a^t |l(t,s,\tau)|\,d\tau \to 0, \int_a^s |m(t,s,\sigma)|\,d\sigma \to 0, \int_a^t \int_a^s |n(t,s,\tau,\sigma)|\,d\tau d\sigma \to 0$$

равномерно по (t,s) при $mesT \to 0, mesS \to 0$, где $T \subset [a,b]$ и $S \subset [a,b]$ — отрезки.

2. $X = L^p(D)\ (1 \le p \le \infty)$

С применением методов из [2, 3, 5] доказывается

Теорема 5. Если оператор K действует в X_+ (в X_-) или действует из X_+ (из X_-) в $X = L^p(D)$ $(1 \le p \le \infty)$, то он непрерывен.

Отметим, что в общем случае оператора K критерии его действия в X_+ (в X_-), из X_+ (из X_-) в $X = L^p(D)$ $(1 \le p \le \infty)$ неизвестны, неизвестны они и в случае пространства $X = L^p(D)$ $(1 < p < \infty)$.

Различные достаточные условия действия оператора K в X_+ (в X_-) получаются объединением условий [2, 3, 5], обеспечивающих действие в $X = L^p(D)$ $(1 \le p \le \infty)$ операторов (4), с равенствами

$$l(t,s,\tau) = m(s,t,\tau), n(t,s,\tau,\sigma) = n(s,t,\tau,\sigma)(n(t,s,\tau,\sigma) = -n(s,t,\tau,\sigma)).$$

Действительно, условия действия оператора K в $X = L^p(D)$ $(1 \le p \le \infty)$ являются и условиями действия оператора K из X_+ и из X_- в X, а выполнение приведенных равенств обеспечивает симметричность функции $(Kx)(t,s)$ при $x \in X_+$ и кососимметричность при $x \in X_-$.

В [2] показано, что $r(K)=0,$ если оператор K обладает введенным первым автором свойством Андо:

$$\lim_{mesT+mesS\to 0}\left\|P_{T\times S}KP_{T\times S}\right\|=0,$$

где $T,S\subset[a,b]$ — измеримые множества, а $P_{T\times S}$ — оператор умножения на характеристическую функцию множества $T\times S.$

Имеет место

Теорема 6. Пусть операторы L,M,N действуют в пространстве X, оператор K обладает свойством Андо и $l(t,s,\tau)=m(s,t,\tau).$

Тогда справедливы утверждения:

а) если $n(t,s,\tau,\sigma)=n(s,t,\tau,\sigma),$ то $r_{+}(K)=0$ и уравнение (3) имеет единственное решение в X_{+} при любой функции $f\in X_{+}$;

б) если $n(t,s,\tau,\sigma)=-n(s,t,\tau,\sigma),$ то $r_{-}(K)=0$ и уравнение (3) имеет единственное решение в X_{-} при любой функции $f\in X_{-}.$

Отметим, что в обоих случаях теорем 4 и 6 решение уравнения (3) может быть получено методом последовательных приближений:

$$x_{n+1}=Kx_{n}+f\ (n=0,1,2,...)$$

при любых $f,x_{0}\in X_{+}$ в случае а) и любых $f,x_{0}\in X_{-}$ — в случае б).

Отметим, что разработанные вторым автором алгоритмы численного решения уравнений Вольтерра с частными интегралами можно найти в [6].

Благодарности. Работа поддержана Минобрнауки РФ, проект 1.4407.2011.

ЛИТЕРАТУРА

1. Калитвин, А.С. О линейных операторах с частными интегралами в пространствах симметричных и кососимметричных функций [Текст] / А.С. Калитвин // Вестник Липецкого государственного педагогического университета. Серия МИФЕ: математика, информационные технологии, физика, естествознание.—Липецк:ЛГПУ, 2012. — 1 (2). — С. 9–13.

2. Калитвин, А.С. Линейные операторы с частными интегралами [Текст] / А.С. Калитвин. — Воронеж: ЦЧКИ, 2000. — 252 с.

3. Appell, J.M. Partial Integral Operators and Integro-Differential Equations [Text] / J.M. Appell, A.S. Kalitvin, P.P. Zabrejko. — New York-Basel: Marcel Dekker, 2000. — 560 p.

4. Калитвин, А.С. Линейные уравнения с частными интегралами. С-теория [Текст] / А.С. Калитвин, Е.В. Фролова. — Липецк: ЛГПУ, 2004. — 195 с.

5. Калитвин, А.С. Нелинейные операторы с частными интегралами [Текст] / А.С. Калитвин. — Липецк: ЛГПУ, 2002. — 208 с.

6. Калитвин, А.С. Интегральные уравнения Вольтерра и Вольтерра – Фредгольма с частными интегралами [Текст] / А.С. Калитвин, В.А. Калитвин. — Липецк: ЛГПУ, 2006. — 177 с.

2012. Вып. 2 (3). С. 26–31

УДК 517.968
519.8

О МАТЕМАТИЧЕСКИХ МОДЕЛЯХ МНОГОСВЯЗНЫХ ЦЕПЕЙ МАРКОВА В L^p

А.С. Калитвин, В.А. Калитвин

Аннотация

Рассматривается задача многосвязных цепей Маркова, математической моделью которой является многомерное линейное интегральное уравнение Романовского. Приводятся условия фредгольмовости этого уравнения в L^p.

1. Вывод многомерного интегрального уравнения Романовского, моделирующего многосвязную цепь Маркова

Пусть даны непрерывная случайная переменная X со значениями на спрямляемой кривой T на плоскости или в трехмерном пространстве и бесконечный ряд опытов с этой переменной, пронумерованных числами 1, 2, 3,.... Предполагается, что при естественной параметризации кривой длина ее произвольной дуги $(t, t+\Delta t)$ обозначается через dt, а параметр изменяется на конечном отрезке. Опытам с переменной X присваиваются номера 1,2,.... Опыты $1,2,...,n$ образуют начальное звено 0, опыты $2,3,...,n+1$ — звено 1 и так далее. Бесконечная последовательность этих звеньев составляет многосвязную цепь Маркова, если выполняются следующие условия.

Значения $x_1, x_2, ..., x_n$ переменной X в начальном звене удовлетворяют дифференциальному закону теории вероятностей $p_0(x_1, x_2, ..., x_n)$ — начальному закону. Дифференциальный закон равенства $X = x_n$ в каком-нибудь опыте дается функцией $\varphi(t, x_1, ..., x_{n-1}, x_n)$ при условии, что в n предыдущих опытах $X = t, X = x_1, ..., X = x_{n-1}$. При этом должно выполняться равенство

$$\int_T \varphi(t, x_1, ..., x_n)\, dx_n = 1.$$

Пусть $p_k(x_1, ..., x_n)$ — дифференциальный закон вероятностей того, что в звене k $X = x_1, ..., X = x_n$, а о результатах других опытов ничего не известно. Тогда

$$p_{k+1}(x_1, ..., x_n) = \int_T p_k(t, x_1, x_2, ..., x_{n-1})\varphi(t, x_1, ..., x_n)\, dt. \tag{1}$$

Если теперь $p_k(x)$ — дифференциальный закон вероятностей равенства $X = x$ в k-ом опыте при условии, что результаты других опытов неизвестны, то

$$p_k(x) = \int_T ... \int_T p_{k-n-1}(t_1, ..., t_n)\varphi(t_1, ..., t_n, x)\, dt_1 ... dt_n.$$

Полагая в (1) $p_k(x_1,...,x_n) = \lambda^{-k} u(x_1,...,x_n)$, получим, что функция $u(x_1,...,x_n)$ удовлетворяет многомерному однородному интегральному уравнению

$$u(x_1,...,x_n) = \lambda \int_T \varphi(t,x_1,...,x_n) u(t,x_1,...,x_{n-1}) dt. \tag{2}$$

Уравнение (2) и неоднородное интегральное уравнение

$$u(x_1,...,x_n) = \lambda \int_T \varphi(t,x_1,...,x_n) u(t,x_1,...,x_{n-1}) dt + f(x_1,...,x_n) \tag{3}$$

изучались В.И. Романовским при $T = [a,b]$ и при условии непрерывности заданных функций $\varphi(t,x_1,...,x_n)$, $f(x_1,...,x_n)$. При этом они рассматривались в пространстве непрерывных функций [1, с. 265-389]. В данной работе уравнения (2) и (3) изучаются при более общих предположениях в пространствах Лебега L^p $(1 \le p \le \infty)$.

2. Условия фредгольмовости многомерного интегрального уравнения Романовского в L^p

Пусть $D = T^n$, $L^p = L^p(D)$ $(1 \le p \le \infty)$. Оператор Π определим равенством $\Pi u(x_1,...,x_{n-1},t) = u(t,x_1,...,x_{n-1})$. Оператор Π, очевидно, действует и непрерывен в L^p, причем его норма равна единице.

Через L обозначим линейный частично интегральный оператор, определяемый равенством

$$Lu(x_1,...,x_n) = \int_T \varphi(t,x_1,...,x_n) u(x_1,...,x_{n-1},t) dt. \tag{4}$$

Действие оператора L в L^p равносильно его непрерывности L^p [2, с. 28-29], [3, с. 215-217]. Тогда действие в L^p оператора $\Phi = L\Pi$ равносильно действию в L^p оператора L. Следовательно, действующий в L^p оператор Φ непрерывен.

Напомним [4, с. 30], что действующий в L^p линейный оператор называется регулярным, если он представим в виде разности двух действующих в L^p положительных операторов, т.е. преобразующих конус неотрицательных функций из L^p в себя.

В [5] (см. также [2, с. 30], [3, p. 218]) показано, что регулярность в L^p оператора L равносильна действию в L^p оператора

$$]L[u(x_1,...,x_n) = \int_T |\varphi(t,x_1,...,x_n)| u(x_1,...,x_{n-1},t) dt. \tag{5}$$

В силу представления $\Phi = L\Pi$ регулярность в L^p оператора Φ совпадает с регулярностью в L^p оператора L, т.е. с действием в L^p частично интегрального оператора (5).

Пусть L — регулярный оператор в L^p. В силу теоремы Фубини в L^p действует оператор $]\Phi[^n$, где оператор $]\Phi[$ определяется равенством

$$]\Phi[u(x_1,...,x_n) = \int_T |\varphi(t,x_1,...,x_n)| u(t,x_1,...,x_{n-1}) dt. \tag{6}$$

Следовательно, Φ^n — регулярный интегральный оператор в L^p с ядром

$$k(x_1,...,x_n,t_1,...,t_n)=\varphi(t_n,x_1,...,x_n)\varphi(t_{n-1},t_n,x_1,...,x_{n-1})...\varphi(t_1,...,t_n,x_1). \tag{7}$$

В виду (7) оператор $\Phi^n = K$, где

$$Ku(x_1,...,x_n)=\int_T...\int_T k(x_1,...,x_n,t_1,...,t_n)u(t_1,...,t_n)dt_1...dt_n. \tag{8}$$

Если теперь регулярный интегральный оператор K компактен в L^p, то для рассматриваемого в L^p уравнения $u = Ku + f \equiv \Phi^n u + f$, где $f \in L^p$, справедлива альтернатива Фредгольма. Так как альтернатива Фредгольма справедлива для уравнения $u = \Phi^n u + f$ в L^p, то в силу [6, с. 511] она справедлива и для уравнения (3) в L^p.

Отметим, что условия компактности регулярных интегральных операторов в L^p хорошо известны [4, с. 91-92, 97, 111-120]. Различные классы интегральных операторов с ядрами, удовлетворяющими условиям типа Л.В. Канторовича, и операторов типа потенциала содержатся в множестве линейных регулярных и компактных в L^p интегральных операторов [4, с. 124-172].

Через P_G обозначим оператор умножения на характеристическую функцию множества G:

$$P_G u(s)=\begin{cases} u(s), \text{ если } s \in G, \\ 0, \text{ если } s \notin G. \end{cases}$$

По теореме 5.6 из [4, с. 92] действующий в L^p $(1 < p < \infty)$линейный регулярный интегральный оператор K компактен точно тогда, когда

$$\lim_{r\to 0}\left\|P_E K P_F\right\| = 0, \tag{9}$$

где E и F — измеримые множества из D, а $r = mesE + mesF$.

Интегральный оператор (8) будет компактным в L^p $(1 \le p < \infty)$, если он имеет компактную мажоранту, т.е.

$$|k(x_1,...,x_n,t_1,...,t_n)| \le k_0(x_1,...,x_n,t_1,...,t_n) \tag{10}$$

и интегральный оператор

$$K_0u(x_1,...,x_n)=\int_T...\int_T k_0(x_1,...,x_n,t_1,...,t_n)u(t_1,...,t_n)dt_1...dt_n. \tag{11}$$

действует в L^p и компактен [4, с. 97].

По теореме 5.4 из [4, с. 91] регулярный в L^p оператор (8) является компактным оператором в L^p, если он действует из L^{p_0} в L^{q_0}, где $1 \le p_0 \le p \le q_0 \le \infty$, причем $p \ne p_0$ или $p \ne q_0$.

Отсюда и из [5, с. 511] вытекает

Теорема 1. *Пусть оператор (5) действует в* L^p $(1 \le p \le \infty)$ *и выполнено одно из условий:*

а) оператор (8) удовлетворяет условию (9) и $1 < p < \infty$;

б) выполнено неравенство (10), оператор (11) компактен в L^p *и* $1 \le p < \infty$;

в) оператор (5) действует из L^{p_0} *в* L^{q_0}, *где* $1 \le p_0 \le p \le q_0 \le \infty$, *причем* $p \ne p_0$ *или* $p \ne q_0$.

Тогда для уравнения (3) в L^p *справедлива альтернатива Фредгольма.*

Компактность действующего в L^p $(1 < p < \infty)$ интегрального оператора (8) имеет место [4, с. 119, 120], если выполнено одно из включений:

$$\left\| k(x_1,...,x_n,\cdot,...,\cdot) \right\|_{L^q} \in L^p, \left\| k(\cdot,...,\cdot,t_1,...,t_n) \right\|_{L^p} \in L^q, \tag{12}$$

где $1/p + 1/q = 1$.

Условия компактности оператора (8) в L^∞ и L^1 вытекают из теорем 6.4 и 6.6 работы [4, с. 113, 116].

Напомним [4, с. 113], что вектор-функцию $\omega(x_1,...,x_n)$ со значениями в некотором L^p $(1 \le p \le \infty)$ называют компактной в существенном, если найдется подмножество $D_0 \subset D$ с мерой, равной мере множества D, на котором значения функции ω принадлежат компактному в L^p множеству.

По теореме 6.4 из [4, с. 113] действующий в L^∞ оператор (8) компактен точно тогда, когда вектор-функция

$$\omega(x_1,...,x_n) = k(x_1,...,x_n,t_1,...,t_n), (x_1,...,x_n) \in D, \tag{13}$$

со значениями в пространстве L^1 компактна в существенном. Так как $D = T^n$ — ограниченное замкнутое множество в R^n, то компактность интегрального оператора (8) вытекает из непрерывности вектор-функции (13). В частности, этот оператор компактен в L^∞, если его ядро принадлежит пространству $C(L^1)$ непрерывных относительно $(x_1,...,x_n) \in D$ вектор-функций со значениями в L^1.

Аналогично, оператор (8) действует в L^1 и компактен тогда и только тогда, когда

$$\left\| k(x_1,...,x_n,\cdot,...,\cdot) \right\|_{L^\infty} \in L^1 \tag{14}$$

и вектор-функция

$$\tilde{\omega}(t_1,...,t_n) = k(x_1,...,x_n,t_1,...,t_n), (t_1,...,t_n) \in D, \tag{15}$$

со значениями в L^1 компактна в существенном [4, с. 116].

В силу [5, с. 511] отсюда вытекает

Теорема 2. *Пусть оператор (5) действует в* L^p $(1 \le p \le \infty)$ *и выполнено одно из условий:*

а) $1 < p < \infty$ *и справедливо хотя бы одно из включений (12);*

б) $p=\infty$ *и вектор-функция (13) со значениями в* L^1 *компактна в существенном;*

в) $p=1$, *имеет место включение (14) и вектор-функция (15) со значениями в* L^1 *компактна в существенном.*

Тогда для уравнения (3) в L^p *справедлива альтернатива Фредгольма.*

Отметим, что непрерывная функция φ принадлежит $C(L_1)$. Поэтому утверждение теоремы 2 справедливо для интегральных уравнений (3) с непрерывными ядрами, которые являются математическими моделями многосвязных цепей Маркова. Утверждение теоремы 2 справедливо, если φ — ограниченная измеримая функция, имеющая разрывы только вдоль конечного числа поверхностей $t=\gamma(x_1,...,x_n)$ с непрерывной функцией γ.

Через Y обозначим пространство $L^p[L^\infty[L^q]]$ $(1\le p\le\infty, 1/p+1/q=1)$ со смешанной нормой [7, с. 9]. Оно состоит из измеримых функций $\varphi(t,x_1,...,x_{n-1},x_n)$, для которых конечна норма

$$\|\varphi\|_Y=\left\|\left\|\left\|\varphi\right\|_{q,t}\right\|_{\infty,x_1,...,x_{n-1}}\right\|_{p,x_n}<\infty.$$

Аналогично, пространство $Z=L^q[L^\infty[L^p]]$ $(1\le p\le\infty, 1/p+1/q=1)$ со смешанной нормой состоит из измеримых функций $\varphi(t,x_1,...,x_{n-1},x_n)$, для которых конечна норма

$$\|\varphi\|_Z=\left\|\left\|\left\|\varphi\right\|_{p,x_n}\right\|_{\infty,x_1,...,x_{n-1}}\right\|_{q,t}<\infty.$$

Непосредственными оценками показывается, что если $\varphi\in Y$ $(\varphi\in Z)$, то выполнено первое (второе) включение (12). Отсюда и из теоремы 2 вытекает

Теорема 3. *Если* $\varphi\in Y$ *или* $\varphi\in Z$, *то для уравнения (3) в* L^p $(1<p<\infty)$ *справедлива альтернатива Фредгольма.*

В заключение отметим, что в [1, с. 387-389] приведены условия, при которых в случае непрерывной функции $\varphi(t_1,...,t_n,x)$ законы $p_k(x)$ и $p_k(x_1,...,x_n)$ стремятся при $k\to\infty$ к конечным законам, т.е.

$$\lim_{k\to\infty}p_k(x_1,...,x_n)=p(x_1,...,x_n)$$

и

$$\lim_{k\to\infty}p_k(x)=\int_D p(t_1,...,t_n)\varphi(t_1,...,t_n,x)dt_1...dt_n,$$

каков бы ни был непрерывный начальный закон $p_0(x_1,...,x_n)$. Последние два равенства выражают важнейшие свойства математических моделей многосвязных цепей Маркова.

Благодарности. Работа поддержана Минобрнауки РФ, проект 1.4407.2011.

ЛИТЕРАТУРА

1. Романовский, В.И. Избранные труды. Т.2: Теория вероятностей, статистика и анализ [Текст] / В.И. Романовский. – Ташкент: Наука, 1964. – 390 с.

2. Калитвин, А.С. Линейные операторы с частными интегралами [Текст] / А.С. Калитвин. – Воронеж: ЦЧКИ, 2000. – 252 с.

3. Appell, J.M. Partial Integral Operators and Integro-Differential Equations [Text] / J.M. Appell, A.S. Kalitvin, P.P. Zabrejko. - New York-Basel: Marcel Dekker, 2000. - 560 p.

4. Красносельский, М.А. и др. Интегральные операторы в пространствах суммируемых функций. – М.: Наука, 1966. – 500 с.

5. Kalitvin, A.S. On the theory of partial integral operators [Text] / A.S. Kalitvin, P.P. Zabrejko // J. Integral Equ. Applications. 1991. – V. 3. – 3. – P. 351-382.

6. Канторович, Л.В. Функциональный анализ [Текст] / Л.В. Канторович, Г.П. Акилов. – М.: Наука,1984. – 752 с.

7. Бесов, О.В. Интегральные представления функций и теоремы вложения [Текст] / О.В. Бесов, В.П. Ильин, С.М. Никольский. – М.: Наука, 1975. – 480 с.