УДК 519.8

Ю.П. ЛАПТИН, А.П. ЛИХОВИД, Н.Н. СТРЮКОВА

РЕШЕНИЕ НЕКОТОРЫХ ЗАДАЧ ПЛАНИРОВАНИЯ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ

Рассматриваются задачи большой размерности, возникающие при использовании сценарного подхода для решения задач многоэтапного стохастического программирования. Такие задачи формулируются в форме вложенных квазиблочных задач линейного программирования со связывающими переменными. Для решения используется схема декомпозиции, основанная на построении линейных аппроксимаций и поиске ε*-оптимальных решений подзадач. Приводятся результаты вычислительных экспериментов.*

В работе рассматриваются задачи большой размерности, возникающие при использовании сценарного подхода для решения задач многоэтапного стохастического программирования [1]. Такой подход приводит к естественному обобщению квазиблочных задач линейного программирования со связывающими переменными [2], в котором каждый блок квазиблочной задачи в свою очередь является квазиблочной задачей. Совокупность блоков-подзадач описывается деревом подзадач и формируется при построении дерева возможных сценариев для исходной задачи стохастического программирования.

Для решения рассматриваемых задач в настоящее время разработаны различные алгоритмы. Большая часть этих алгоритмов является обобщением схемы декомпозиции Бендерса.

В настоящей работе исследуются возможности алгоритма, предложенного в [3]. Содержанием алгоритма является построение и последовательное уточнение аппроксимаций для подзадач сформированного дерева. Для уточнения аппроксимаций используется приближенное решение вспомогательных квазиблочных задач линейного программирования. Особенностью рассматриваемого алгоритма является возможность согласования точности аппроксимаций для различных подзадач и точности решения квазиблочных задач.

Возможности разработанных программных средств исследуются на примере стохастической задачи планирования инвестиций в электроэнергетике, являющейся модификацией модели из [5]. Приводятся результаты вычислительных экспериментов по сравнению разработанных и существующих программных средств.

1. Вложенные квазиблочные задачи линейного программирования

1.1. При использовании сценарного подхода для решения задач многоэтапного линейного стохастического программирования формируется детерминированный аналог задачи, который может быть описан следующим образом. Задано некоторое дерево (V,E). Каждой вершине $q \in V$ поставлена в соответствие оптимизационная задача, зависящая от параметров. Вектор параметров обозначим x^q, вектор переменных, по которым осуществляется оптимизация – y^q. Задачи, соответствующие разным вершинам дерева (V,E), являются связанными. Связь заключается в том, что если вершина q есть непосредственный потомок вершины p, то

$$x^q = (x^p, y^p), \tag{1}$$

т. е. совокупность параметров и переменных родительской задачи – параметры дочерней задачи. Для корневой вершины r предполагается, что $x^r = 0$.

Пусть $f^q(x^q)$ – некоторая функция параметров x^q вершины q. В дальнейшем будем считать эквивалентными записи $f^q(x^q)$ и $f^q(x^p, y^p)$, где q – непосредственный потомок вершины p.

Обозначим $S(q)$ множество непосредственных потомков вершины q.

Оптимизационная задача для вершины $q \in V$ имеет вид: найти

$$\varphi^q(x^q) = \min_{y^q} \left\{ c^q y^q + \sum_{s \in S(q)} \varphi^s(x^q, y^q) \right\}, \tag{2}$$

при ограничениях

$$A^q y^q \le d^q - B^q x^q. \tag{3}$$

Матрицы A^q, B^q, вектор-строка c^q и вектор d^q считаются заданными для каждой вершины $q \in V$.

Для висячих вершин дерева (V,E) задачи (2)–(3) являются задачами линейного программирования, для остальных вершин задачи (2)–(3) определяются рекуррентно. Известно (см., например, [2]), что функции $\varphi^q(x^q)$, $q \in V$, являются выпуклыми. Для корневой вершины оптимизационная задача от параметров не зависит (или, что эквивалентно, значения параметров зафиксированы).

Совокупность задач (2)–(3) для всех вершин дерева (V,E) может быть представлена одной задачей линейного программирования большой размерности, обладающей вложенной структурой. В случае, когда дерево состоит из корня и набора висячих вершин, эта задача является квазиблочной задачей со связывающими переменными [2] и с дополнительными ограничениями (3).

1.2. Функцию $H(x, y) = \max_i \left\{ (h_x^i, x) + (h_y^i, y) + \eta_i,\ i = 1, ..., I \right\}$ будем называть кусочно-линейной нижней аппроксимацией для выпуклой функции $F(x, y)$, если $H(x, y) \le F(x, y)$.

Пусть задана некоторая кусочно-линейная нижняя аппроксимация $H^q(x^q, y^q)$ для функции $\sum_{s \in S(q)} \varphi^s(x^q, y^q)$ в выражении (2) для каждого $q \in V$. Для висячих вершин $H^q(x^q, y^q) \equiv 0$.

Будем называть l-аппроксимацией задачи (2)–(3) следующую задачу: найти

$$\psi^q(x^q) = \min_{y^q} \left\{ c^q y^q + H^q(x^q, y^q) \right\}, \tag{4}$$

при ограничениях

$$A^q y^q \le d^q - B^q x^q, \tag{5}$$

ld-аппроксимацией задачи (2)–(3) будем называть задачу: найти

$$\xi^q(x^q) = \min_{y^q} \left\{ c^q y^q + \sum_{s \in S(q)} \psi^s(x^q, y^q) \right\}, \tag{6}$$

при ограничениях

$$A^q y^q \le d^q - B^q x^q. \tag{7}$$

Заметим, что ld-аппроксимация для каждой невисячей вершины есть квазиблочная задача линейного программирования со связывающими переменными с дополнительными ограничениями (7).

Аппроксимации $H^q(x^q, y^q),\ q \in V$ назовем согласованными, если $\psi^q(x^q) \le \xi^q(x^q),\ q \in V$.

Обозначим x совокупность всех параметров и переменных вершин дерева (V, E). Пусть некоторым образом определены значения компонент вектора $\bar{x}$. Тем самым определены значения вектора $(\bar{x}^q, \bar{y}^q)$ для каждой вершины $q \in V$.

Вектор $\bar{x}$ будем называть допустимым, если для каждой вершины выполняются ограничения (3). Обозначим V^* множество висячих вершин дерева (V, E). Пусть задан набор неотрицательных чисел $\varepsilon = (\varepsilon^q,\ q \in V \setminus V^*)$.

Согласованные аппроксимации $H^q(x^q, y^q),\ q \in V \setminus V^*$ будем называть ε-сбалансированными в допустимой точке $\bar{x}$, если

$$0 \le c^q \bar{y}^q + \sum_{s \in S(q)} \psi^s(\bar{x}^q, \bar{y}^q) - \psi^q(\bar{x}^q) \le \varepsilon^q,\ q \in V \setminus V^*. \tag{8}$$

Заметим, что соотношение (8) для фиксированного q является условием ε^q-оптимальности решения $\bar{y}^q$ квазиблочной задачи (6)–(7) при $x^q = \bar{x}^q$.

Лемма [3]. Пусть аппроксимации $H^q(x^q, y^q),\ q \in V \setminus V^*$ являются ε-сбалансированными в точке $\bar{x}$. Тогда $\bar{x}$ есть σ-оптимальное решение совокупности задач (2)–(3), поставленных в соответствие вершинам дерева (V, E), где $\sigma = \sum_{q \in V \setminus V^*} \varepsilon^q$.

1.3. В дальнейшем предполагается, что задачи (2)–(3) имеют решение для любых значений параметров. Для обеспечения таких свойств в [2] предложена специальная регуляризация исходной задачи. Обозначим $\xi^q(x^q, y^q)$ значение целевой функции задачи (6)–(7) в допустимой точке y^q

$$\xi^q(x^q, y^q) = c^q y^q + \sum_{s \in S(q)} \psi^s(x^q, y^q). \tag{9}$$

$P(\varepsilon^q)$**-процедурой** решения ld-аппроксимации квазиблочной задачи (6)–(7) для заданного $q \in V$ и фиксированного x^q назовем процедуру поиска ε^q-оптимального решения y^q этой задачи и построения (уточнения) аппроксимации $H^q(x^q, y^q)$, для которых выполняется

$$0 \le \xi^q(x^q, y^q) - \psi^q(x^q) \le \varepsilon^q,\ q \in V. \tag{10}$$

Для реализации $P(\varepsilon^q)$-процедуры могут использоваться различные подходы – методы негладкой оптимизации [2], методы секущих плоскостей [1] и др. При этом для вычисления субградиентов (отсекающих гиперплоскостей) необходимо находить оптимальные значения двойственных переменных для подзадач квазиблочной задачи. Для программной реализации более простыми являются схемы, основанные на методах секущих плоскостей и современных программных средствах линейного программирования. В дальнейшем будем предполагать, что в нашем распоряжении имеется некоторая реализация $P(\varepsilon^q)$-процедуры.

Пусть задан вектор $\varepsilon = (\varepsilon^q,\ q \in V)$. Рассмотрим $A(\varepsilon)$ **-алгоритм** поиска допустимой точки x и построения ε-сбалансированных аппроксимаций $H^q(x^q, y^q),\ q \in V$ в этой точке. Будем предполагать заданными некоторые начальные аппроксимации $H_0^q(x^q, y^q)$, $q \in V$. Пусть для каждой вершины $q \in V$ задана некоторая текущая аппроксимация $H^q(x^q, y^q)$. Рассмотрим связное поддерево $B = (V_b, E_b)$ дерева (V, E), содержащее корневую вершину r. Обозначим $x(B)$ совокупность параметров и переменных задач поддерева B.

Поддерево B будем называть ε-сбалансированным для аппроксимаций $H^q(x^q, y^q)$, $q \in V_b$ в допустимой точке $x(B)$, если условие (10) выполняется для всех $q \in V_b$. На каждой итерации рассматриваемого $A(\varepsilon)$-алгоритма анализируется и модифицируется некоторое текущее ε-сбалансированное поддерево B (если $V_b = \varnothing$, то поддерево B будем называть пустым), текущие точка $x(B)$, аппроксимация $H^q(x^q, y^q)$ для всех $q \in V$. Считается заданным параметр алгоритма γ, $0 < \gamma < 1$, некоторая начальная точка $\tilde{x}(V, E)$.

Приведем описание $A(\varepsilon)$ **-алгоритма.**

Шаг 0. Положить $V_b = \varnothing$ (B – пустое поддерево).

Шаг 1. Выбрать вершину $q \in V \setminus V_b$, для непосредственного предка p которой выполняется $p \in V_b$. Если $V_b = \varnothing$, выбрать корневую вершину дерева.

Шаг 2. Если для текущей точки (x^q, y^q) выполняется условие (10), добавить вершину q в поддерево B и перейти на шаг 5.

Шаг 3. Применить $P(\gamma\varepsilon^q)$-процедуру к вершине q, пусть y^q – полученное ε^q-оптимальное решение.

Шаг 4. Исключить непосредственного предка p и всех его потомков из поддерева B.

Шаг 5. Если поддерево B совпадает с исходным деревом (V, E), закончить процесс вычислений (искомая точка x и ε-сбалансированные аппроксимации построены), иначе перейти на шаг 1.

Приведем некоторые пояснения шагов 3 и 4. Применение $P(\gamma\varepsilon^q)$-процедуры к вершине q приводит к уточнению аппроксимации $H^q(x^q, y^q)$, что в свою очередь приводит к увеличению разности $\xi^p(x^p, y^p) - \psi^p(x^p)$ для непосредственного предка p вершины q. При нарушении условия (10) для вершины p необходимо повторно уточнять аппроксимацию для этой вершины и искать уточненное решение y^p, т. е. применять $P(\gamma\varepsilon^p)$-процедуру. В случае $0 < \gamma < 1$ при малом увеличении разности $\xi^p(x^p, y^p) - \psi^p(x^p)$ не приходится повторно применять $P(\gamma\varepsilon^p)$-процедуру.

В качестве начальных аппроксимаций выбираются функции $H_0^q(x^q, y^q) \equiv -\bar{H}$, $q \in V$, где $\bar{H}$ – достаточно большое положительное число.

Алгоритм решения совокупности задач (2)–(3) для всех вершин дерева (V, E) состоит в итеративном применении $A(\varepsilon)$-алгоритма. На первой итерации определяется начальное значение вектора $\varepsilon = (\varepsilon^q, q \in V)$. На каждой итерации уточняются (уменьшаются) значения компонент вектора ε и применяется $A(\varepsilon)$-алгоритм. Процесс прекращается при достижении требуемой точности.

Характер поведения $A(\varepsilon)$-алгоритма зависит от выбора вектора ε и правил его уточнения на каждой итерации. Нетрудно видеть, что при относительно малых ε^q для вершин дерева близких к корню (по сравнению с удаленными от корня вершинами) соответствующие ld-аппроксимационные задачи решаются часто и с большой точностью, при этом используются грубые аппроксимации $H^s(x^s, y^s)$ для их потомков. В случае, когда выбираются относительно малые ε^q для вершин дерева близких к висячим, чаще решаются ld-аппроксимационные задачи для этих вершин и используются более точные аппроксимации $H^s(x^s, y^s)$ для их потомков. В последнем случае точки, относительно которых строятся точные аппроксимации, могут быть достаточно далеки от оптимальных.

1.4. Описанный алгоритм был реализован средствами языка программирования C++ с использованием библиотеки классов линейного программирования Soplex [4]. Для реализация $P(\varepsilon^q)$-процедуры был выбран алгоритм Келли. Начальные значения вектора $\varepsilon = (\varepsilon^q, q \in V)$ задавались следующим образом: $\varepsilon^q = \varepsilon_0 * (Pr_q)^s$, где ε_0, s – параметры алгоритма, Pr_q – вероятность реализации вершины q дерева (V, E). При итеративном применении $A(\varepsilon)$-алгоритма величины ε^q изменялись следующим образом $\varepsilon^q = t * \varepsilon^q$, где $0 < t < 1$ – еще один параметр алгоритма. Результаты вычислительных экспериментов для задач, описаных далее, приведены на рисунке.

2. Стохастическая модель планирования инвестиций в электроэнергетике

В качестве основы для рассматриваемой модели взята постановка задачи из [5]. На каждом этапе планирования t интервала планирования $[0, ..., T]$ осуществляются инвестиции в n различных технологий. С каждой технологией i связаны соответствующие значения случайных удельных капитальных затрат α^{it}, эксплуатационных затрат (затрат на обслуживание) γ^{it}.

Обозначим:

g_i^t – динамика изменения мощностей технологии i, которые существовали в начале процесса планирования;

X^{it} – новые мощности технологии i, которые планируется создать в момент времени t;

Δ_i – период введения новой мощности технологии i в эксплуатацию.

Выбор значений X^{it}, $t = 0,..., T$ осуществляется в начале интервала планирования – составляется план инвестиций. В дальнейшем, в начале каждого этапа планирования t, когда известна информация о реализации случайных величин предыдущего этапа, принятых решениях на предыдущих этапах и появляется дополнительная информация о реализации случайных величин текущего этапа, осуществляется корректировка плана. Такие корректировки приводят к определенным потерям, связанным с расторжением инвестиционных контрактов, заключением новых контрактов на менее выгодных условиях. Обозначим:

W^{it} – корректировка (уменьшение) плана инвестиций для технологии i в момент времени t, $W^{it} \le 0$;

U^{it} – корректировка (увеличение) плана инвестиций для технологии i в момент времени t, $U^{it} \ge 0$;

β^{it} – случайное удельное сокращение средств при уменьшении (корректировке) плана инвестиций для технологии i в момент времени t (уменьшение затрат равно $\beta^{it} W^{it}$);

χ^{it} – случайные удельные затраты при увеличении (корректировке) плана инвестиций для технологии i в момент времени t (увеличение затрат равно $\chi^{it} U^{it}$).

После введения в эксплуатацию новых мощностей их величина не изменяется. Обозначим G^{it} новые наличные мощности, доступные в момент времени t, тогда

$$G^{it} = \sum_{\tau=0}^{t-\Delta_i} (X^{i\tau} + W^{i\tau} + U^{i\tau}) = G^{i(t-1)} + X^{i(t-\Delta_i)} + W^{i(t-\Delta_i)} + U^{i(t-\Delta_i)},$$

$$t \ge \Delta_i$$

$$G^{it} = 0,\ t < \Delta_i .$$

Случайные запросы на электроэнергию могут приходить от различных потребителей, и удовлетворение этих запросов должно быть полностью выполнено на каждом временном этапе. В случае если наличных мощностей не хватает, имеется возможность закупки электроэнергии на внешнем рынке (по более высоким ценам). Обозначим: d^t – запрос на электроэнергию на этапе t, Z^t – величина закупки электроэнергии на внешнем рынке, Y^{it} – количество генерируемой электроэнергии по технологии $i=1,\ldots,n$, получим

$$\sum_{i=1}^{n} Y^{it} + Z^t \geq d^t, \quad t = 0,\ldots,T.$$

Должны также выполняться ограничения на генерируемые мощности:

$$Y^{it} \leq G^{it} + g_i^t, \quad i = 1,\ldots,n,\ t = 0,\ldots,T.$$

В задаче необходимо найти значения переменных, которые минимизируют математическое ожидание затрат, представленных в виде суммы капитальных и эксплуатационных затрат. Случайными параметрами являются запросы на электроэнергию d^t, значения удельных капитальных и эксплуатационных затрат α^{it}, γ^{it}, удельных затрат на корректировку плана β^{it}, χ^{it}, $i=1,\ldots,n$, а также цена λ^t электроэнергии, закупаемой на внешнем рынке. Обозначим ξ^t векторную случайную переменную с элементами $\left\{\alpha^{it}, \beta^{it}, \chi^{it}, \gamma^{it}, \lambda^t, d^t, \forall i,t\right\}$. Рассматриваемая задача имеет вид: найти

$$\varphi^* = \min_{x,y,z,w,u} E_\xi \left[\sum_{t=1}^{T} \left(\sum_{i=1}^{n} \left(\alpha^{it} X^{it} + \beta^{it} W^{it} + \chi^{it} U^{it} + \gamma^{it} Y^{it} \right) + \lambda^t Z^t \right) \right],$$

при ограничениях

$$\sum_{i=1}^{n} Y^{it} + Z^t \geq d^t, \quad t = 0,\ldots,T,$$

$$G^{it} = G^{i(t-1)} + X^{i(t-\Delta_i)} + W^{i(t-\Delta_i)} + U^{i(t-\Delta_i)}, \quad t \geq \Delta_i,\ i = 1,\ldots,n$$

$$G^{it} = 0,\ t < \Delta_i.$$

$$Y^{it} \leq g_i^t + G^{it}, \quad i = 1,\ldots,n,\ t = 0,\ldots,T,$$

$$X^{it} + W^{it} + U^{it} \geq 0, \quad i = 1,\ldots,n,\ t = 0,\ldots,T,$$

$$X^{it} \geq 0,\ Y^{it} \geq 0,\ i = 0,\ldots,n,\ Z^t \geq 0,\ t = 0,\ldots,T,$$

$$W^{it} \leq 0,\ U^{it} \geq 0,\ i = 0,\ldots,n,\ t = 0,\ldots,T.$$

Для описанной модели формировались задачи двух-, трех-, четырех- и пятиэтапного планирования. Каждая невисячая вершина дерева сценариев имела десять потомков (в некоторых случаях – 20). Случайными величинами считались потребности в электроэнергии. Предполагалось, что эти величины независимы для каждого этапа и дискретно равномерно распределены на отрезке.

Исходные данные для экземпляров задач формировались в форматах MPS и SMPS с использованием системы моделирования SLP-IOR [1]. Это позволило для вычислительных экспериментов использовать следующие существующие программные средства – Soplex, MSLiP [6]. Поскольку для рассматриваемой задачи в MSLiP выявились программные ошибки, то сравнение проводилось только с пакетом Soplex.

С увеличением размерности увеличивалась разница времени решения задач разработанными средствами и системой Soplex. Для максимальной по размерности задачи пятиэтапного планирования (20 000 сценариев, число ограничений – 211 104, число переменных – 337 773) результаты следующие:

- общее время решения Soplex-ом – 925.91 сек. (процессорное время – 651.02 сек.), оптимальное значение – 3.4887500e+03;

- общее время решения разработанными программными средствами ~ 65 сек. (варьируется в зависимости от выбора параметров алгоритма), по ходу вычислительного процесса генерируются приближенные решения, точность которых увеличивается.

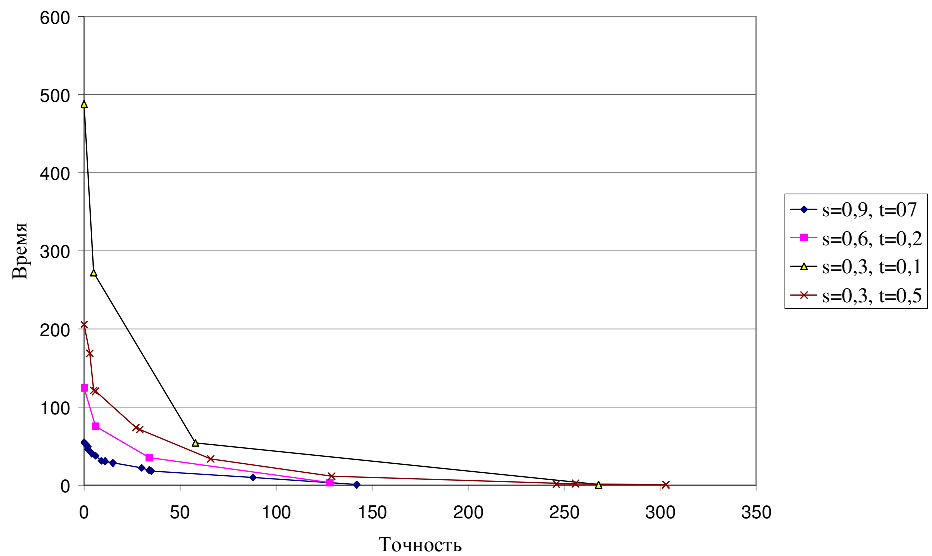

РИСУНОК. Зависимость времени (сек.) поиска приближенного решения от точности

Поскольку по ходу решения исходной задачи используются и уточняются кусочно-линейные аппроксимации задач (2)–(3), важной характеристикой алгоритма является число строк, добавленных к исходной задаче. При этом добавленные строки, которые длительное время остаются неактивными в решениях задач (4)–(5), удаляются из задачи. Финальное число строк равно 233 286 (исходное – 211 104) при использовании параметров алгоритма p = 0.9, q = 0.7. Такое относительно небольшое количество добавленных строк объясняется тем, что такие строки добавляются только к задачам, соответствующим невисячим вершинам дерева задач. Динамика вычислительного процесса в зависимости от выбора параметров алгоритма показана на рисунке и подтверждает предположение о существенной зависимости эффективности алгоритма от согласования точности решения квазиблочных задач (6)–(7) в ходе вычислений.

Ю.П. Лаптін, О.П. Лиховид, Н.М. Стрюкова

РОЗВ'ЯЗАННЯ ДЕЯКИХ ЗАДАЧ ПЛАНУВАННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ

Розглядаються задачі великої розмірності, що виникають при використанні сценарного підходу для розв'язання задач багатоетапного стохастичного програмування. Такі задачі формулюються у вигляді вкладених квазіблочних задач лінійного програмування зі зв'язуючими змінними. Для розв'язання використовується схема декомпозиції, заснована на побудові лінійних апроксимацій та пошуку ε-оптимальних рішень підзадач. Наводяться результати обчислювальних експериментів.

Yu.P. Laptin, O.P. Lykhovyd, N.N. Strukova

SOLUTION OF SOME PLANNING PROBLEMS UNDER UNCERTAINTY

Large-scale problems, arising from application of scenario approach for solving multistage stochastic programming problems, are considered. Such problems are formulated in the form of quasi-block linear programming problems with linking variables. Decomposition scheme based on construction of linear approximations and searching ε-optimal solutions for subproblems is used. The results of numerical experiments are presented.

1. *Mayer J.* Stochastic Linear Programming Algorithms: a Comparison Based on a Model Management System. – Amsterdam: Overseas PublishersAssociation, 1998. – 153 p.
2. *Shor N. Z.* Nondifferentiable Optimization and Polynomial Problems. – London: Kluwer Academic Publishers, 1998. – 381 p.
3. *Лаптин Ю.П., Лиховид А.П.* Один подход к решению оптимизационных задач с вложенной структурой // Теорія оптимальних рішень. – 2007. – № 6. – С. 90–99.
4. *Wunderling R.* Paralleler und Objektorientierter Simplex-Algorithmus, ZIB technical report TR 96-09, Berlin 1996. (http://www.zib.de/Optimization/Software/Soplex/soplex.php)
5. *Louveaux F.V. and Smeers Y.* Optimal investments for electricity generation: A stochastic model and a test-problem // In R.J-B. Wets and Y.Ermoliev, editors, Numerical Techniques for Stochastic Optimization. Springer-Verlag, 1988. – **24.** – P. 445–453.

6. *Gassmann H.I.* MSLiP: A computer code for the multistage stochastic linear programming problem // Math. Prog. – 1990. – **47.** – P. 407–423.

Получено 29.04.2010