УДК 519.853.4

© А.В. Орлов

ГИБРИДНЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ГЛОБАЛЬНОГО ПОИСКА ОПТИМИСТИЧЕСКИХ РЕШЕНИЙ В ЗАДАЧАХ ДВУХУРОВНЕВОЙ ОПТИМИЗАЦИИ[1]

Рассматривается разработка гибридного подхода к решению квадратично-линейных задач двухуровневой оптимизации в оптимистической постановке. Эффективность предложенного подхода демонстрируется вычислительным экспериментом.

Ключевые слова: двухуровневая оптимизация, оптимистическое решение, теория глобального поиска, локальный поиск, построение аппроксимации поверхности уровня, генетический алгоритм.

© A.V. Orlov

HYBRID GENETIC ALGORITHM OF GLOBAL SEARCH FOR OPTIMISTIC SOLUTIONS IN BILEVEL OPTIMIZATION PROBLEMS

The article is devoted to elaboration of a hybrid approach to solving quadratic-linear bilevel optimization problems in optimistic formulation. The computational experiment shows the efficiency of the proposed approach.

Keywords: bilevel optimization, optimistic solution, global search theory, local search, constructing of the level surface approximation, genetic algorithm.

Введение

Как известно, разработка эффективных методов поиска решений в задачах с иерархической структурой представляет собой актуальную задачу современной математической оптимизации [1, 2], что обусловлено в первую очередь широким полем приложений иерархических задач. Аппарат двухуровневой оптимизации весьма удобен при моделировании конфликтов с неравноправным положением сторон, возникающих в практических задачах управления, экономики, транспорта и других областей [3, 4].

Например, при исследовании транспортно-логистических систем с помощью двухуровневой оптимизации моделируются задача назначения тарифов (Toll Setting Problem) [5] и задача развития транспортной сети (Highway Network Design Problem) [6]. На железнодорожном транспорте двухуровневой структурой обладает задача формирования поездов с целью перевозки грузов (Train Set Organization) [7] и ряд других.

Отметим, что практические приложения зачастую требуют решения задач достаточно высокой размерности. Ранее в нашей группе были разработаны методы локального и глобального поиска оптимистических решений в линейных [8, 9] и квадратично-линейных [8, 10, 11] задачах двухуровневой оптимизации, базирующиеся на теории глобального поиска в невыпуклых задачах с (d.c.) функциями А.Д. Александрова (т.е. представимыми в виде разности двух выпуклых функций) [12, 13]. Численное тестирование разработанных методов на большом спектре тестовых задач различной сложности и размерности продемонстрировало эффективность предлагаемого подхода по решению линейных задач с суммарной размерностью переменных на обоих уровнях до 1000 [9] и по решению квадратично-линейных задач – с размерностью до 300 [8, 11].

Тем не менее проблема разработки новых алгоритмов двухуровневой оптимизации, позволяющих оперировать с задачами еще большей размерности, в настоящее время остается актуальной. В данной работе с этой целью на примере квадратично-линейной двухуровневой задачи предлагается гибридный подход к построению метода глобального поиска, который, с одной стороны, базируется на теории глобального поиска [12], а с другой стороны – для реализации одного из этапов глобального поиска использует блоки генетических алгоритмов, такие как кроссовер и мутация [14].

[1] Работа выполнена при финансовой поддержке РФФИ, проекты 11-01-00270-а, 12-07-33045-мол_а_вед, 12-07-13116-офи_м_РЖД.

При этом разрабатываемый гибридный алгоритм включает в себя специальные методы локального поиска, использующие билинейную структуру исследуемой задачи [13].

1. Элементы теории глобального поиска

При решении невыпуклых задач математической оптимизации (математического программирования) с помощью подхода, основанного на теории глобального поиска [12], одним из ключевых этапов является построение аппроксимации поверхности уровня выпуклой функции, задающей базовую невыпуклость в исследуемой задаче [13]. Успешное построение аппроксимации позволяет «выскочить» из стационарной (критической) точки, полученной с помощью локального поиска, что, как известно, является одной из главных целей глобального поиска.

Напомним кратко этапы глобального поиска в задаче d.c. минимизации [12]:

$$\Phi(x) \stackrel{\Delta}{=} g(x) - f(x) \downarrow \min, \quad x \in D, \qquad (DC)$$

где $g(\cdot)$, $f(\cdot)$ – выпуклые функции, D – выпуклое множество.

Пусть известна некоторая приближенно критическая (стационарная) в задаче (DC) точка z^k со значением целевой функции $\zeta_k := \Phi(z^k)$, построенная с помощью какого-либо метода локального поиска. Тогда производится следующая цепочка операций:

1) выбирается число $\gamma \in [\gamma_-, \gamma_+]$, где $\gamma_- := \inf(g, D)$, $\gamma_+ := \sup(g, D)$ и строится некоторая конечная аппроксимация

$$A_k = \{v^1, ..., v^N \mid f(v^i) = \gamma - \zeta_k, i = 1, ..., N\}$$

поверхности уровня $U(\zeta_k) = \{y \mid f(y) = \gamma - \zeta_k\}$ выпуклой функции $f(\cdot)$;

2) для всех точек аппроксимации A_k проверяется неравенство $g(v^i) \le \gamma$, $i = 1, 2, ..., N$, следующее из условий глобальной оптимальности для задачи (DC) [12];

3) исходя из точек v^i, выбранных на втором этапе, осуществляется локальный поиск, доставляющий приближенно критические точки $\hat{x}^i$, $i \in \{1, ..., N\}$ в задаче (DC);

4) значение целевой функции в каждой точке $\hat{x}^i$ сравнивается со значением целевой функции в текущей критической точке z. Если какая-то из точек $\hat{x}^i$ оказывается лучше текущей, происходит обновление последней, и весь процесс повторяется до исчерпания аппроксимации поверхности уровня.

Отметим, что построенная на этапе 1 аппроксимация должна быть достаточно репрезентативной, чтобы можно было судить о том, является ли текущая критическая точка z^k глобальным решением [12].

В настоящее время не разработано методов построения репрезентативной аппроксимации для задачи (DC) в общем виде. При решении конкретных невыпуклых задач эта проблема решается на основе накопленного опыта построения аппроксимаций [8, 9, 11-13]. Только для нескольких простейших невыпуклых задач (например, для задачи максимизации нормы на параллелепипеде) удалось построить аппроксимации, которые гарантируют достижение глобального решения [12], поэтому разработка новых подходов к решению этой проблемы является актуальной задачей невыпуклой оптимизации.

В данной работе эту задачу предлагается решать с использованием элементов генетического алгоритма.

2. Основные этапы генетического алгоритма

Напомним далее основные этапы генетического алгоритма [14], представив их для простоты на примере самой общей задачи оптимизации следующего вида:

$$F(x) \downarrow \min, \quad x \in D. \qquad (P_0)$$

Сначала каким-либо образом, например, случайно, задается набор допустимых в задаче (P_0) точек, называемый популяцией:

$$Pop = \{x^i \mid x^i \in D, \quad i = 1,...,N\}.$$

Затем выбирается функция пригодности, с помощью которой можно оценить эти точки (чаще всего для этого используется целевая функция задачи $F(\cdot)$).

Далее, с помощью двух случайно выбранных точек (особей) из популяции, которые называются родителями, определенным образом строятся две новые особи (потомки): $x^{i_1}, x^{i_2} \in Pop \Rightarrow y^1 \in D, y^2 \in D.$ Эта процедура носит название кроссовер (кроссинговер, скрещивание). Существующие варианты осуществления кроссовера весьма разнообразны. Наиболее популярными являются одноточечный, двухточечный и однородный кроссовер [14]. Главной сложностью здесь является необходимость сохранения допустимости в исходной задаче построенных потомков.

Следующий этап – случайная мутация некоторых компонент построенных потомков, осуществляемая с определенной вероятностью: $y^1 \Rightarrow w^1 \in D$, $y^2 \Rightarrow w^2 \in D$. Здесь также необходимо следить за допустимостью получающихся особей [14].

После этого две получившиеся особи сравниваются друг с другом, и если лучшая из них с точки зрения функции пригодности оказывается лучше худшей особи из популяции, производится обновление последней: $w := arg\min\{F(w^1), F(w^2)\}$, $j : F(x^j) \geq F(x^i) \ \forall x^i \in Pop$; если $F(w) < F(x^j)$, то $x^j := w$.

Процесс обычно заканчивается, когда произведено определенное заранее заданное количество итераций (поколений) описанной процедуры [14].

Разумеется, здесь описан только один из многих вариантов генетических алгоритмов. Количество публикаций по этой теме для различного сорта задач очень и очень велико (см., например, литературу в [14]). При этом неизменным остается главная идея подобного сорта подходов – аналогия с эволюцией живых организмов в ходе естественного отбора.

Далее, на примере квадратично-линейной двухуровневой задачи представим один из способов «встраивания» схемы генетического алгоритма в процедуру глобального поиска, описанную в первом разделе статьи.

3. Базовый алгоритм глобального поиска в квадратично-линейной двухуровневой задаче

Рассмотрим следующую квадратично-линейную задачу двухуровневой оптимизации в оптимистической постановке (на верхнем уровне производится минимизация по переменным обоих уровней):

$$\left.\begin{aligned} F(x,y) \stackrel{\Delta}{=} \frac{1}{2}\langle x, Cx\rangle + \langle c, x\rangle + \frac{1}{2}\langle y, C_1 y\rangle + \langle c_1, y\rangle \downarrow \min_{x,y}, \\ (x,y) \in X \stackrel{\Delta}{=} \{(x,y) \in R^m \times R^n \mid Ax + By \leq a, \ x \geq 0\}, \\ y \in Y_*(x) \stackrel{\Delta}{=} \operatorname{Arg}\min_y \{\langle d, y\rangle \mid y \in Y(x)\}, \\ Y(x) \stackrel{\Delta}{=} \{y \in R^n \mid A_1 x + B_1 y \leq b, \ y \geq 0\}, \end{aligned}\right\} (BP)$$

где $A \in R^{p\times m}$, $B \in R^{p\times n}$, $A_1 \in R^{q\times m}$, $B_1 \in R^{q\times n}$, $C \in R^{m\times m}$, $C_1 \in R^{n\times n}$, $c \in R^m$, $c_1, d \in R^n$, $a \in R^p$, $b \in R^q$ кроме того, $C = C^T \geq 0, C_1 = C_1^T \geq 0$.

Отметим, что целевая функция верхнего уровня $F(x,y)$ является выпуклой квадратичной с разделенными переменными. Целевая функция нижнего уровня является линейной. Пусть функция $F(x,y)$ ограничена снизу на множестве X, а функция $\langle d, y\rangle$ ограничена снизу на множестве $Y(x)$ для всех $x \in Pr(X) \stackrel{\Delta}{=} \{x \in R^m \mid \exists y : (x,y) \in X\}$, так что

$$\inf_x \inf_y \{\langle d, y\rangle \mid y \in Y(x), x \in Pr(X)\} > -\infty.$$

Обозначим через $h(x,y,v) := \langle d,y\rangle - \langle A_1x - b, v\rangle$ невязку двойственности для задачи нижнего уровня, где $v \in R^q$ – вектор множителей Лагранжа в задаче нижнего уровня. В [8, 10] была осуществлена редукция задачи (BP) к следующему μ -параметрическому семейству задач:

$$\left.\begin{array}{l} \Phi(x,y,v) = F(x,y) + \mu h(x,y,v) \downarrow \min\limits_{x,y,v}, \\ (x,y,v) \in D = \{(x,y,v) \mid Ax + By \le a, \quad A_1x + B_1y \le b, \\ d + vB_1 \ge 0, \quad x \ge 0, \quad y \ge 0, \quad v \ge 0\}, \end{array}\right\} \qquad (P(\mu))$$

где $\mu > 0$ – штрафной множитель.

Заметим, что ограничения в задаче ($P(\mu)$) накладываются на пару (x,y) и на переменную v раздельно. Основываясь на этом факте, для осуществления локального поиска в задаче ($P(\mu)$) в [8, 10] были предложены две процедуры (XY-процедура и V-процедура), доставляющие точки, которые являются частично глобальными решениями исходной задачи по несвязанным группам переменных (критические точки). Эти процедуры заключаются в последовательном решении вспомогательных задач выпуклого квадратичного и линейного программирования (ЛП).

Для построения процедуры глобального поиска, согласно методологии из [12], прежде всего необходимо получить явное d.c. разложение целевой функции задачи. Нетрудно видеть, что целевая функция $\Phi(x,y,v)$ задачи ($P(\mu)$) может быть представлена в виде разности двух выпуклых функций, например, следующим образом:

$$\Phi(x,y,v) = g(x,y,v) - f(x,y,v), \qquad (1)$$

где

$$g(x,y,v) = \frac{1}{2}\langle x, Cx\rangle + \langle c,x\rangle + \frac{1}{2}\langle y, C_1y\rangle + \langle c_1,y\rangle + \mu(\langle v,b\rangle + \frac{1}{4}\mathrm{P}v - A_1x\mathrm{P}^2),$$

$f(x,y,v) = \mu(\frac{1}{4}\mathrm{P}v + A_1x\mathrm{P}^2 - \langle y,d\rangle)$ – выпуклые функции.

Представим далее алгоритм глобального поиска в задаче ($P(\mu)$), использующий d.c. разложение (1). Пусть дана некоторая точка $(x_0,y_0,v_0) \in R^{m+n+q}$, числовые последовательности $\{\tau_k\},\{\delta_k\}$, $\tau_k,\delta_k > 0, k = 0,1,..., \tau_k \downarrow 0, \delta_k \downarrow 0\ (k \to \infty)$, множество направлений

$$Dir = \{(a^l,b^l,c^l) \in R^{m+n+q} \mid (a^l,b^l,c^l) \ne 0, l = 0,1,...,N\},$$

числа $\gamma_- := \inf\limits_{(x,y,v)} (g,D)$ и $\gamma_+ := \sup\limits_{(x,y,v)} (g,D)$ и параметры алгоритма M и ν .

Базовый алгоритм глобального поиска

Шаг 0. Положить $k := 0, \quad (\overline{x}^k, \overline{y}^k, \overline{v}^k) := (x_0,y_0,v_0)$, $l := 1$, $\gamma := \gamma_-, \quad \Delta\gamma := (\gamma_+ - \gamma_-)/M$.

Шаг 1. Начиная с точки $(\overline{x}^k, \overline{y}^k, \overline{v}^k)$, с помощью XY- или V-процедуры локального поиска построить τ_k – критическую точку $(x^k,y^k,v^k) \in D$ в задаче ($P(\mu)$).

Положить $\zeta_k := \Phi(x^k,y^k,v^k)$.

Шаг 2. По точке $(a^l,b^l,c^l) \in Dir$ построить точку (z^l,u^l,w^l) из аппроксимации A_k поверхности уровня функции $f(\cdot)$:

$$A_k = \{(z^l,u^l,w^l) \mid f(z^l,u^l,w^l) = \gamma - \zeta_k\}.$$

Шаг 3. Если $g(z^l,u^l,w^l) > \gamma + \nu\gamma$, то положить $l := l+1$ и перейти на шаг 2.

Шаг 4. Начиная с точки (z^l,u^l,w^l) , с помощью процедуры локального поиска построить δ_k – критическую точку $(\hat{x}^l,\hat{y}^l,\hat{v}^l) \in D$ в задаче ($P(\mu)$).

Шаг 5. Если $\Phi(\hat{x}^l,\hat{y}^l,\hat{v}^l) < \Phi(x^k,y^k,v^k)$, то положить

$(\bar{x}^{k+1}, \bar{y}^{k+1}, \bar{v}^{k+1}) := (\hat{x}^l, \hat{y}^l, \hat{v}^l)$, $k := k+1$, $l := 1$, $\gamma := \gamma_-$ и перейти на шаг 1.

Шаг 6. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$, $l < N$, то положить $l := l+1$ и вернуться на шаг 2.

Шаг 7. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$, $l = N$ и $\gamma < \gamma_+$, то положить $\gamma := \gamma + \Delta\gamma$, $l := 1$ и вернуться на шаг 2.

Шаг 8. Если $l = N$, $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$ $\forall \gamma \in [\gamma_-, \gamma_+]$ (т.е. одномерный поиск по γ на отрезке $[\gamma_-, \gamma_+]$ закончен), то стоп; (x^k, y^k, v^k) – полученное решение задачи.

Шаги 1-8 представляют собой алгоритмизованную запись этапов глобального поиска, приведенных в разделе 1, которая осуществлена в терминах задачи ($P(\mu)$). При этом отметим, что на шаге 3 алгоритма производится проверка точки на пригодность к дальнейшим исследованиям с помощью неравенства, вытекающего из условий оптимальности для задач d.c. минимизации [12]. При численной реализации необходимо использовать параметр ν, варьированием которого можно изменять точность выполнения этого неравенства (с целью уменьшить влияние машинных ошибок округления [8, 11, 13]). Различные же значения параметра M отвечают за разбиение отрезка $[\gamma_-, \gamma_+]$ на соответствующее количество частей для реализации пассивного одномерного поиска по γ.

Обратим также внимание на шаг 2, где аппроксимация поверхности уровня функции f, которая задает базовую невыпуклость в исследуемой задаче, строится с помощью заданного множества направлений Dir. Это множество было выбрано экспериментально для задач с билинейной структурой (к которым относится и задача ($P(\mu)$)) [8, 11, 13] и включает в себя стандартные векторы евклидового базиса и компоненты текущей критической точки. Недостатком такого подхода к построению аппроксимации является, в частности, то, что оно не меняется от итерации к итерации алгоритма и содержит достаточно большое количество точек (в некоторых случаях порядка $(m+n)q$). Последний факт в значительной степени влияет на эффективность работы алгоритма глобального поиска при повышении размерности задачи.

Гибридный алгоритм глобального поиска, описанию которого посвящен следующий раздел, лишен указанных недостатков.

4. Гибридный алгоритм глобального поиска и его тестирование

Прежде всего опишем выбранные способы реализации генетических блоков для гибридного алгоритма решения задачи ($P(\mu)$).

В качестве популяции на каждой итерации глобального поиска выбираются точки, составляющие текущую аппроксимацию поверхности уровня:

$$Pop_k = \{(r^s, z^s, w^s) \mid (r^s, z^s, w^s) \in A_k,\ s = 1,...,N\}.$$

Функция пригодности $Loc(\cdot)$, оценивающая точки аппроксимации, представляет собой значение целевой функции в критической точке, построенной исходя из соответствующей точки аппроксимации: $Loc(r^s, z^s, w^s) := \Phi(\hat{x}^s, \hat{y}^s, \hat{v}^s)$. Здесь отметим, что поскольку точки аппроксимации могут быть недопустимыми, важным представляется свойство используемых процедур локального поиска, сохраняющих работоспособность при старте с недопустимой точки [8, 10].

Для осуществления скрещивания была реализована процедура так называемого однородного кроссовера [14]: выберем два произвольных индекса $s_1, s_2 \in \{1,...N\}$ $(s_1 \neq s_2)$ и положим $l := Random[0,1]$.

Далее $\forall j = 1,...,m+n+q$, если $l < 0.5$, то $(\breve{r}, \breve{z}, \breve{w})_j := (r^{s_1}, z^{s_1}, w^{s_1})_j$, $(\tilde{r}, \tilde{z}, \tilde{w})_j := (r^{s_2}, z^{s_2}, w^{s_2})_j$, иначе $(\tilde{r}, \tilde{z}, \tilde{w})_j := (r^{s_1}, z^{s_1}, w^{s_1})_j$, $(\breve{r}, \breve{z}, \breve{w})_j := (r^{s_2}, z^{s_2}, w^{s_2})_j$. Таким образом, каждая компонента одного из векторов-потомков с вероятностью $1/2$ представляет собой компоненту одного из родителей.

В качестве процедуры мутации была выбрана следующая: пусть $P_m := 0.01$, $K = const$. Положим $l_1 := Random[0,1]$. Если $l_1 < P_m$, то $(\breve{r}, \breve{z}, \breve{w})_j := Random[-K, K]$ $\forall j = 1,...,m+n+q.$ Положим

$l_2 := Random[0,1]$. Если $l_2 < P_m$, то $(\tilde{r},\tilde{z},\tilde{w})_j := Random[-K,K] \quad \forall j = 1,...,m+n+q$. Подобного сорта процедура называется равномерной мутацией [14].

В результате был построен следующий гибридный алгоритм глобального поиска, сочетающий этапы базового алгоритма и блоки генетического алгоритма.

Пусть даны точка $(x_0, y_0, v_0) \in D$, множество направлений $Dir = \{(a^1,b^1,c^1),...,(a^N,b^N,c^N) \in R^{m+n+q} \mid (a^s,b^s,c^s) \neq 0, \ s = 1,...,N\}$, числа $\gamma_- := \inf(g, D)$ и $\gamma_+ := \sup(g, D)$, вероятность мутации P_m и количество поколений G_{max}.

Гибридный алгоритм глобального поиска

Шаг 0. Положить $k := 0, \quad (\bar{x}^k, \bar{y}^k, \bar{v}^k) := (x^0, y^0, v^0), \quad s := 1, \quad \gamma := \gamma_-, \quad \Delta\gamma = (\gamma_+ - \gamma_-)/N$.

Шаг 1. Начиная с точки $(x_0, y_0, v_0) \in D$ с помощью *XY*- или *V*-процедуры локального поиска найти приближенно критическую точку $(\hat{x}^0, \hat{y}^0, \hat{v}^0) \in D$ в задаче $(P(\mu))$. Положить $\zeta_s := \Phi(\hat{x}^0, \hat{y}^0, \hat{v}^0)$.

Шаг 2. По точкам $(a^s,b^s,c^s) \in Dir$ построить точки (r^s, z^s, w^s) аппроксимации поверхности уровня функции f, $s = 1,...,N$, такие что $f(r^s, z^s, w^s) = \gamma + \Delta\gamma * (s-1) - \zeta_s$, т.е. построить популяцию точек поверхности уровня Pop_k. Для каждой точки популяции вычислить значение функции пригодности: $\zeta_s := Loc(r^s, z^s, w^s), \ s = 1,...,N$.

Шаг 3. $s_1 := Random\{1,...,N\}$, $s_2 := Random\{1,...,N\}$. С помощью процедуры кроссовера по точкам $(r^{s_1}, z^{s_1}, w^{s_1})$ и $(r^{s_2}, z^{s_2}, w^{s_2})$ построить точки $(\tilde{r},\tilde{z},\tilde{w})$ и $(\breve{r},\breve{z},\breve{w})$.

Шаг 4. С вероятностью P_m реализовать процедуру мутации для точек $(\tilde{r},\tilde{z},\tilde{w})$ и $(\breve{r},\breve{z},\breve{w})$.

Шаг 5. Вычислить $\tilde{\zeta} := \Phi(\tilde{r},\tilde{z},\tilde{w})$, $\breve{\zeta} := \Phi(\breve{r},\breve{z},\breve{w})$, $\tilde{\gamma} := g(\tilde{r},\tilde{z},\tilde{w})$, $\breve{\gamma} := g(\breve{r},\breve{z},\breve{w})$ и построить две новые точки лежащие на поверхностях уровня: $f(\bar{r}^{s_1}, \bar{z}^{s_1}, \bar{w}^{s_1}) = \tilde{\gamma} - \tilde{\zeta}$ и $f(\bar{r}^{s_2}, \bar{z}^{s_2}, \bar{w}^{s_2}) = \breve{\gamma} - \breve{\zeta}$.

Шаг 6. Вычислить значения $Loc(\bar{r}^{s_1}, \bar{z}^{s_1}, \bar{w}^{s_1})$ и $Loc(\bar{r}^{s_2}, \bar{z}^{s_2}, \bar{w}^{s_2})$. Пусть $(\bar{r}^s, \bar{z}^s, \bar{w}^s) := \arg\min\{Loc(\bar{r}^{s_1}, \bar{z}^{s_1}, \bar{w}^{s_1}), \ Loc(\bar{r}^{s_2}, \bar{z}^{s_2}, \bar{w}^{s_2})\}$.

Шаг 7. Вычислить номер $j : Loc(r^j, z^j, w^j) \geq Loc(r^s, z^s, w^s) \quad \forall (r^s, z^s, w^s) \in Pop_k$; если $Loc(\bar{r}^s, \bar{z}^s, \bar{w}^s) < Loc(r^j, z^j, w^j)$, то $(r^j, z^j, w^j) := (\bar{r}^s, \bar{z}^s, \bar{w}^s)$.

Шаг 8. Если $k < G_{max}$, то $k := k+1$ и перейти на шаг 3, иначе стоп. $(x^*, y^*, v^*) := \arg\min\{Loc(r^s, z^s, w^s) \mid (r^s, z^s, w^s) \in Pop_k, \ s = \overline{1,N}\}$ – полученное решение задачи.

Отметим, что процедуру построения точек, лежащих на поверхности уровня, на шаге 5 алгоритма можно реализовать для любой точки пространства (кроме 0), так что проблем с обеспечением допустимости точек, полученных в результате применения блоков генетического алгоритма, здесь не возникает.

Для сравнения эффективности базового и гибридного алгоритмов глобального поиска использовались тестовые задачи, сгенерированные с помощью специального метода, предложенного в [15], который основан на построении задач произвольной размерности с известными локальными и глобальными решениями из задач-ядер небольшой размерности (см. также [8-11, 13]). При этом для простоты полагаем $C = 0$ и $C_1 = 0$.

Результаты сравнения приведены в таблице со следующими обозначениями: *Name* – имя тестового примера в формате $n+m\_N$, где n – число переменных на верхнем уровне, m – число переменных на нижнем уровне, N – номер примера данной размерности; LP_b и LP_g – количество вспомогательных задач ЛП, которое потребовалось решить для достижения глобального решения базовым и гибридным алгоритмами соответственно; T_b и T_g – время решения примера в формате мин:сек.доли с помощью базового и гибридного алгоритма соответственно; *Pop* и *MaxG* –

параметры гибридного алгоритма, обозначающие размер популяции и количество поколений соответственно.

Таблица

Сравнение базового и гибридного алгоритмов глобального поиска

Name	LP_b	T_b	*Pop*	*MaxG*	LP_g	T_g
1+1_1	118	0:00.02	3	5	56	0:00.00
3+3_1	118	0:00.02	3	5	55	0:00.00
5+5_1	**18**	**0:00.02**	**3**	**5**	**60**	**0:00.02**
5+5_2	512	0:00.09	3	5	62	0:00.02
10+10_1	854	0:00.72	3	5	62	0:00.05
15+15_1	1210	0:02.92	3	10	104	0:00.26
20+20_1	1610	0:08.67	3	20	193	0:01.09
25+25_1	2010	0:20.33	3	20	200	0:02.26
30+30_1	2410	0:39.27	5	20	208	0:04.00
35+35_1	2810	1:10.64	5	20	208	0:06.36
35+35_2	2810	1:11.12	5	20	208	0:06.38
40+40_1	3728	2:21.38	5	50	470	0:21.03
40+40_2	3210	2:02.36	5	50	469	0:21.00
50+50_1	4650	5:39.44	10	50	492	0:42.03
50+50_2	4010	4:53.03	10	50	492	0:43.44
75+75_1	6970	32:14.69	10	50	497	2:53.38
75+75_2	6982	32:42.48	10	50	497	2:54.31
100+100_1	9290	102:09.97	10	50	519	7:21.72

По результатам, представленным в таблице, можно видеть, что во всех случаях, кроме одного (для примера 5+5_1, который отмечен в таблице жирным шрифтом), удалось подобрать такой размер популяции и количество поколений, что гибридный алгоритм оказался существенно эффективнее и по времени, и по количеству вспомогательных задач ЛП, которые потребовалось решить для достижения глобального решения. Последнюю величину можно считать некоторой мерой сложности решаемой задачи для используемого алгоритма. В некоторых случаях превосходство гибридного алгоритма над базовым составило 10-15 раз.

Заключение

В работе предложен новый гибридный подход к разработке алгоритмов поиска оптимистических решений в квадратично-линейных двухуровневых задачах оптимизации, сочетающий использование теории глобального поиска [12] и элементы генетических алгоритмов [14].

На основании проведенного вычислительного эксперимента по сравнению базового и гибридного алгоритмов глобального поиска можно сделать вывод о том, что использование для одного из самых сложных этапов глобального поиска – построения аппроксимации поверхности уровня – блоков генетического алгоритма оказалось оправданным. Имеющийся численный и алгоритмический опыт по решению различного сорта задач исследования операций с билинейной структурой [8-11, 13] позволяет также утверждать, что предложенный гибридный подход будет эффективен и при решении более сложных двухуровневых задач, в том числе, возникающих на автомобильном и железнодорожном транспорте [5-7].

Литература

1. Pang J.-S. Three modelling paradigms in mathematical programming // Mathematical programming, Ser.B. 2010. V. 125, No. 2. P. 297-323.

2. Dempe S. Foundations of Bilevel Programming. – Dordrecht: Kluwer Academic Publishers, 2002.

3. Bard J.F. Practical Bilevel Optimization. – Dordrecht: Kluwer Academic Publishers, 1998.

4. Colson B., Marcotte P., Savard G. An overview of bilevel optimization // Annals of operations research. 2007. V. 153. P. 235-256.

5. Brotcorne L., Labbe M., Marcotte P., Savard G. A Bilevel Model for Toll Optimization on a Multicommodity Transportation Network // Transportation Science. 2001. V. 35, No. 4. P. 345–358.

6. Ben-Ayed O., Blair C.E., Boyce D.E., LeBlanc L.J. Construction of a real-world bilevel linear programming model of the highway network design problem // Annals of Operations Research. 1992. V. 34, No. 1-4. P. 219-254.

7. Gao Y., Lu J., Zhang G., Gao S. A Bilevel Model for Railway Train Set Organizing Optimization // Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2007) (October 15-16, 2007, Chengdu, China). Atlantis Press, 2007, doi:10.2991/iske.2007.239.

8. Strekalovsky A.S., Orlov A.V., Malyshev A.V. On computational search for optimistic solution in bilevel problems // Journal of Global Optimization. 2010. V. 48, No. 1. P. 159-172.

9. Груздева Т.В., Петрова Е.Г. Численное решение линейной двухуровневой задачи // Журнал вычислительной математики и математической физики. 2010. Т. 50, № 10. С. 1715-1726.

10. Стрекаловский А.С., Орлов А.В., Малышев А.В. Локальный поиск в квадратично-линейной задаче двухуровневого программирования // Сиб. журн. вычисл. математики. 2010. Т. 13, № 1. С. 75-88.

11. Стрекаловский А.С., Орлов А.В., Малышев А.В. Численное решение одного класса задач двухуровневого программирования // Сиб. журн. вычисл. математики. 2010. Т. 13, № 2. С. 201-212.

12. Стрекаловский А.С. Элементы невыпуклой оптимизации. Новосибирск: Наука, 2003.

13. Стрекаловский А.С., Орлов А.В. Биматричные игры и билинейное программирование. М.: ФИЗМАТЛИТ, 2007.

14. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. New York: Springer-Verlag, 1994.

15. Calamai P., Vicente L. Generating Linear and Linear-Quadratic Bilevel Programming Problems // SIAM Journal on Scientific Computing. 1993. V. 14, No. 4. P. 770-782.

Орлов Андрей Васильевич, кандидат физико-математических наук, доцент, старший научный сотрудник Федерального государственного бюджетного учреждения науки Институт динамики систем и теории управления Сибирского отделения Российской академии наук, 664033, Иркутск, ул. Лермонтова, 134, тел. +7(3952) 45-30-63, e-mail: anor@icc.ru.

Orlov Andrey Vasilievich, candidate of physical and mathematical sciences, associate professor, senior researcher, Institute for System Dynamics and Control Theory SB RAS, 664033, Irkutsk, Lermontov str., 134, ph. +7(3952) 45-30-63, e-mail: anor@icc.ru.