

Научная статья

УДК 519.853.6

DOI: 10.18101/2304-5728-2023-2-14-29

ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ ГИБРИДНОГО АЛГОРИТМА ГЛОБАЛЬНОГО ПОИСКА В ГЕКСАМАТРИЧНЫХ ИГРАХ

© Орлов Андрей Васильевич

кандидат физико-математических наук, доцент,
старший научный сотрудник, заведующий лабораторией,
Институт динамики систем и теории управления
имени В. М. Матросова СО РАН
Россия, 664033, г. Иркутск, ул. Лермонтова, 134
anor@icc.ru

Аннотация. В статье представлен гибридный подход к разработке методов отыскания ситуаций равновесия по Нэшу в полиматричных играх трех лиц (гексаматричных играх). С одной стороны, он базируется на теории глобального поиска в невыпуклых задачах оптимизации с d.c. функциями (представимыми в виде разности двух выпуклых функций), созданной А. С. Стрекаловским, с другой — для реализации одного из ключевых этапов глобального поиска (построения аппроксимации поверхности уровня) используются операторы генетических алгоритмов. После описания гибридного подхода подробно рассказывается об организации и проведении вычислительного эксперимента по сравнению гибридного алгоритма с «базовым» алгоритмом глобального поиска, разработанным ранее. Приведены результаты эксперимента на сериях случайно сгенерированных задач, свидетельствующие об эффективности предложенного гибридного подхода к решению гексаматричных игр.

Ключевые слова: полиматричные игры трех лиц, гексаматричные игры, равновесие Нэша, теория глобального поиска, локальный поиск, алгоритм глобального поиска, аппроксимация поверхности уровня, генетические операторы, гибридный алгоритм, вычислительный эксперимент.

Благодарности

Работа выполнена в рамках базового проекта фундаментальных исследований Минобрнауки РФ «Теоретические основы, методы и высокопроизводительные алгоритмы непрерывной и дискретной оптимизации для поддержки междисциплинарных научных исследований» (Номер гос. регистрации: 121041300065-9, код проекта FWEW-2021-0003).

Для цитирования

Орлов А. В. Численное исследование гибридного алгоритма глобального поиска в гексаматричных играх // Вестник Бурятского государственного университета. Математика, информатика. 2023. № 2. С. 14–29.

Введение

Численное нахождение равновесий в теории игр [1] является одной из актуальных проблем современной математической оптимизации [2], когда поиск равновесия может быть сведен к экстремальной задаче. Известно, что классическая матричная игра эквивалентна двум двойственным задачам линейного программирования (ЛП) [1]. Это означает, что такая игра имеет выпуклую структуру и принципиальных трудностей с ее решением не возникает. Первым расширением матричной игры является биматричная игра, которая уже представляет собой невыпуклую (билинейную) структуру [1, 3, 4]. То же самое можно сказать и о полиматричных играх [5]. Например, поиск равновесия по Нэшу в полиматричной игре трех лиц оказывается эквивалентным специальной задаче невыпуклой оптимизации с тройной билинейной структурой в целевой функции [5–8].

В нашей группе разработан оптимизационный подход к численному нахождению равновесий Нэша в таких играх, основанный на этой эквивалентности [4, 5]. При этом получающиеся оптимизационные задачи решаются с помощью теории глобального поиска (ТГП), созданной А. С. Стрекаловским для невыпуклых задач с d.c. функциями, представимыми в виде разности двух выпуклых функций [9, 10]. Одной из особенностей ТГП является возможность применения внутри алгоритмов достижений современной выпуклой оптимизации (см., например [11, 12]).

Одним из важнейших этапов при использовании ТГП [9, 10] является построение аппроксимации поверхности уровня выпуклой функции, задающей базовую невыпуклость в исследуемой задаче (см. также [4, 13]). Например, в ходе решения следующей задачи d.c. максимизации:

$$\Phi(x) \triangleq h(x) - g(x) \uparrow \max, \quad x \in S, \quad (\mathcal{DC})$$

где $g(\cdot)$, $h(\cdot)$ — выпуклые функции, S — выпуклое множество, необходимо построить конечную аппроксимацию

$$\mathcal{A}(\zeta, \xi) = \{v^1, \dots, v^N \mid h(v^i) = \xi + \zeta, r = 1, \dots, N\}, \quad (1)$$

где $\inf(g, S) \leq \xi \leq \sup(g, S)$ фиксировано, $\zeta \triangleq \Phi(z)$ — значение целевой функции задачи (\mathcal{DC}) в текущей стационарной (критической) точке z . При этом аппроксимация (1) должна быть достаточно репрезентативной, чтобы можно было определить, является ли текущая точка z глобальным решением. Подробнее см. в [4, 9, 10, 13].

В настоящее время не существует общих подходов к построению репрезентативных аппроксимаций для задачи (\mathcal{DC}) . При решении конкретных невыпуклых задач решение этого вопроса основывается на

накопленном численном опыте [4, 9, 13–15]. Поэтому разработка новых подходов к построению аппроксимаций является актуальной проблемой невыпуклой оптимизации. При этом алгоритмы глобального поиска (АГП), основанные на ТГП, оказываются весьма эффективными для биматричных игр высокой размерности (с плотными матрицами до 1000 стратегий у одного игрока) [4], в то же время решение гексаматричных игр с помощью АГП ограничивается разреженными матрицами размера (100×100) [8].

Поиск ситуации равновесия в гексаматричных играх исследовался и другими авторами. Например, в основе подхода из [16] лежит редукция подобных игр к задачам смешанной непрерывно-целочисленной оптимизации (были решены игры до размерности 13 стратегий у каждого из игроков). А в [17] предложена специальная модификация классического метода Лемке-Хаусона (см., например, [4]) в сочетании с так называемым ЗЛР-методом, родственным методу локального поиска, который используется в настоящей работе (решены игры вплоть до 500 стратегий у одного игрока).

В данной работе проводится численное исследование гибридного алгоритма глобального поиска (ГАГП) в гексаматричных играх, разработанного в [18], где для построения аппроксимаций поверхности уровня используются операторы генетических алгоритмов (ГА) [19, 20]. Впервые подобная идея была применена для решения простейших двухуровневых задач в [21]. При этом как в [21], так и в [18] было решено только несколько простейших тестовых задач, а вопрос об общей эффективности подхода при решении большого количества задач различной сложности и размерности оставался открытым. Здесь мы представляем детализацию гибридного алгоритма и результаты его тестирования на широком спектре случайно сгенерированных гексаматричных игр.

1 Постановка задачи и ее редукция

Кратко напомним постановку гексаматричной игры в смешанных стратегиях [5–8]:

$$\left. \begin{aligned} F_1(x, y, z) &\triangleq \langle x, A_1 y + A_2 z \rangle \uparrow \max_x, & x \in S_m \subset \mathbb{R}^m, \\ F_2(x, y, z) &\triangleq \langle y, B_1 x + B_2 z \rangle \uparrow \max_y, & y \in S_n \subset \mathbb{R}^n, \\ F_3(x, y, z) &\triangleq \langle z, C_1 x + C_2 y \rangle \uparrow \max_z, & z \in S_l \subset \mathbb{R}^l, \end{aligned} \right\}$$

где x, y, z — стратегии 1, 2 и 3 игроков соответственно, $A_1, A_2, B_1, B_2, C_1, C_2$ — матрицы, а $S_p = \{u = (u_1, \dots, u_p)^T \in \mathbb{R}^p \mid u_i \geq 0, \sum_{i=1}^p u_i = 1\}$, $p = m, n, l$ — канонические симплексы соответствующей размерности.

Цель состоит в отыскании ситуации равновесия по Нэшу в игре $\Gamma_3 = \Gamma(A, B, C)$ ($A = (A_1, A_2)$, $B = (B_1, B_2)$, $C = (C_1, C_2)$) [5–8]. Находясь в этом равновесии, ни одному из игроков невыгодно в одиночку изменять свою оптимальную стратегию. При этом, согласно теореме Нэша, такое равновесие в игре Γ_3 существует [5] ($NE(\Gamma_3) \neq \emptyset$). В [5] также доказано, что эта задача эквивалентна поиску глобального решения следующей невыпуклой оптимизационной задачи ($\sigma \triangleq (x, y, z, \alpha, \beta, \gamma)$):

$$\left. \begin{aligned} \Phi(\sigma) &\triangleq \langle x, A_1y + A_2z \rangle + \langle y, B_1x + B_2z \rangle + \langle z, C_1x + C_2y \rangle - \alpha - \beta - \gamma \uparrow \max_{\sigma}, \\ \sigma \in S &\triangleq \{(x, y, z, \alpha, \beta, \gamma) \in \mathbb{R}^{m+n+l+3} \mid x \in S_m, y \in S_n, z \in S_l, \\ &A_1y + A_2z \leq \alpha e_m, B_1x + B_2z \leq \beta e_n, C_1x + C_2y \leq \gamma e_l\}, \end{aligned} \right\} \quad (\mathcal{P})$$

где $e_p = (1, 1, \dots, 1) \in \mathbb{R}^p$, $p = m, n, l$; α, β, γ — вспомогательные скалярные переменные. Тройка (x^*, y^*, z^*) , входящая в решение задачи (\mathcal{P}) , является ситуацией равновесия по Нэшу, кроме того, оптимальные значения переменных α_* , β_* и γ_* — это выигрыши игроков, а оптимальное значение задачи $\mathcal{V}(\mathcal{P}) \triangleq \Phi(x^*, y^*, z^*, \alpha_*, \beta_*, \gamma_*) = 0$. При этом справедливо:

$$\alpha_* = \max_{1 \leq i \leq m} (A_1y^* + A_2z^*)_i, \beta_* = \max_{1 \leq j \leq n} (B_1x^* + B_2z^*)_j, \gamma_* = \max_{1 \leq t \leq l} (C_1x^* + C_2y^*)_t. \quad (2)$$

Можно также показать, что в случае отыскания приближенного глобального решения задачи (\mathcal{P}) мы имеем приближенное равновесие по Нэшу ($NE(\Gamma_3, \varepsilon)$) [5]. Отметим, что количество переменных в получившейся задаче равно $(m + n + l + 3)$, количество ограничений — $2(m + n + l) + 3$, однако для краткости, говоря ниже о размерности сгенерированных задач, будем иметь в виду количество стратегий у одного из игроков.

Для применения ТГП к задаче (\mathcal{P}) [9, 10] прежде всего построим явное d.c. разложение ее целевой функции на разность двух выпуклых:

$$\Phi(x, y, z, \alpha, \beta, \gamma) = h(x, y, z) - g(x, y, z, \alpha, \beta, \gamma), \quad (3)$$

где

$$\left. \begin{aligned} h(x, y, z) &= \frac{1}{4} \left(\|x + A_1y\|^2 + \|x + A_2z\|^2 + \|B_1x + y\|^2 + \|y + B_2z\|^2 + \right. \\ &+ \|C_1x + z\|^2 + \|C_2y + z\|^2 \Big), \quad g(\sigma) = \frac{1}{4} \left(\|x - A_1y\|^2 + \|x - A_2z\|^2 + \right. \\ &+ \|y - B_1x\|^2 + \|y - B_2z\|^2 + \|C_1x - z\|^2 + \|C_2y - z\|^2 \Big) + \alpha + \beta + \gamma. \end{aligned} \right\} \quad (4)$$

Таким образом, мы получаем задачу d.c. максимизации вида (DC), где базовая невыпуклость заключена в функции $h(\cdot)$. Далее с учетом свойств задачи (P) и d.c. разложения (3)–(4) представим теоретические основы для построения алгоритмов ее решения на основе ТГП.

2 «Базовый» и гибридный методы глобального поиска

Фундаментом АГП являются так называемые условия глобальной оптимальности (УГО), позволяющие охарактеризовать именно глобальное решение исследуемой задачи [9, 10]. Приведем здесь УГО для задачи (P) (в контрпозитивной форме) [6–8].

Обозначим: $\alpha(y, z) \triangleq \max_{1 \leq i \leq m} (A_1 y + A_2 z)_i$, $\beta(x, z) \triangleq \max_{1 \leq j \leq n} (B_1 x + B_2 z)_j$, $\gamma(x, y) \triangleq \max_{1 \leq t \leq l} (C_1 x + C_2 y)_t$. Тогда согласно свойству (2), $\alpha_* = \alpha(y^*, z^*)$, $\beta_* = \beta(x^*, z^*)$, $\gamma_* = \gamma(x^*, y^*)$, если $(x^*, y^*, z^*) \in NE(\Gamma_3)$.

Теорема 2.1 [6–8]. Если допустимый набор $\sigma^* = (x^*, y^*, z^*, \alpha_*, \beta_*, \gamma_*)$ не является глобальным решением задачи (P), тогда найдутся тройка $(u, v, w) \in \mathbb{R}^{m+n+l}$, вектор $(\bar{x}, \bar{y}, \bar{z}) \in S_m \times S_n \times S_l$ и число ξ такие, что

$$\begin{aligned} h(u, v, w) - \xi &= \zeta \triangleq \Phi(\sigma^*) < 0, \\ g(u, v, w, \alpha(v, w), \beta(u, w), \gamma(u, v)) &\leq \xi \leq \sup(g, S), \end{aligned} \quad (5)$$

и при этом справедливо следующее неравенство:

$$g(\bar{x}, \bar{y}, \bar{z}, \bar{\alpha}, \bar{\beta}, \bar{\gamma}) - \xi < \langle \nabla_{xyz} h(u, v, w), (\bar{x}, \bar{y}, \bar{z}) - (u, v, w) \rangle, \quad (6)$$

где $\bar{\alpha} \triangleq \alpha(\bar{y}, \bar{z})$, $\bar{\beta} \triangleq \beta(\bar{x}, \bar{z})$, $\bar{\gamma} \triangleq \gamma(\bar{x}, \bar{y})$.

Конструктивное (алгоритмическое) свойство сформулированных УГО, которое позволяет в случае отыскания набора $(u, v, w, \bar{x}, \bar{y}, \bar{z}, \xi)$, удовлетворяющего неравенству (6), сразу получить точку лучше, чем исследуемая (даже стационарная или критическая), является основой АГП, построенных на базе теории глобального поиска.

Напомним основные этапы таких алгоритмов (подробнее см. [3, 4, 6–9, 13–15, 18]). Прежде всего осуществляется локальный поиск, принимающий во внимание специфику исследуемой задачи, который доставляет приближенно критические точки. В данном случае используется метод типа «mountain climbing» [22], заключающийся в последовательном решении задачи по группам переменных (специально сконструированных «частичных» задач ЛП), который доказал свою практическую эффективность для задач с билинейной структурой, в том числе для гексаматричных игр [4, 6–8, 13–15, 22]. Далее реализуется один из ключевых для успешного глобального поиска этапов — построение аппроксимации поверхности уровня выпуклой функции $h(\cdot)$. Важным элементом АГП является также решение линеаризованных по базовой

невыхуклости задач $(\mathcal{P}\mathcal{L}_r) := (\mathcal{P}\mathcal{L}(u^r, v^r, w^r))$ (для получения хороших стартовых точек при локальном поиске):

$$g(\sigma) - \langle \nabla h(u^r, v^r, w^r), (x, y, z) \rangle \downarrow \min_{\sigma} \text{, } \sigma \in S, \quad (\mathcal{P}\mathcal{L}_r)$$

где линейаризация проводится в точках аппроксимации (u^r, v^r, w^r) , $r = 1, \dots, N$. Кроме того, осуществляется одномерный поиск по параметру УГО ξ ; проводится отбор перспективных точек с поверхности уровня (с помощью неравенства из (5)), и лучшие по целевой функции критические точки оцениваются с точки зрения получения окончательного результата (как уже говорилось, ее значение в глобальном решении равно нулю). Подробное описание «базового» алгоритма глобального поиска в гексаматричных играх см. в [6–8, 18].

Построение эволюционных, генетических и гибридных (на их основе) алгоритмов обычно начинается с выбора так называемой функции приспособленности, с помощью которой можно оценить точки популяции (набор допустимых точек задачи) [19, 20]. Однако в нашем случае в качестве популяции особей на каждой итерации k глобального поиска выступают не допустимые точки, а множество точек, лежащих на текущей поверхности уровня: $Pop_k = \{(u^r, v^r, w^r) \mid (u^r, v^r, w^r) \in \mathcal{A}_k, r = 1, \dots, N\}$, где $\mathcal{A}_k = \{(u^1, v^1, w^1), \dots, (u^N, v^N, w^N) \mid h(u^r, v^r, w^r) = \xi + \zeta_k, r = 1, \dots, N\}$, $\zeta_k := \Phi(\sigma^k)$, $\sigma^k \triangleq (x^k, y^k, z^k, \alpha_k, \beta_k, \gamma_k) \in S$ — критическая точка (текущий рекорд), $\xi \in [\inf(g, S), \sup(g, S)]$.

При этом функция приспособленности не является, как в классических вариантах ГА, просто целевой функцией задачи, поскольку точки аппроксимации могут быть в задаче недопустимыми, что влечет бессмысленность подобного их оценивания. Выбранная с этой целью функция $PLoc(\cdot)$ представляет собой значение целевой функции в критической точке $\hat{\sigma}^r$, построенной с помощью метода локального поиска исходя из решения $\bar{\sigma}^r$ линейаризованной задачи $(\mathcal{P}\mathcal{L}_r)$, которая, в свою очередь, линейаризуется в соответствующей точке аппроксимации (u^r, v^r, w^r) , так что $PLoc(u^r, v^r, w^r) \triangleq \Phi(\hat{\sigma}^r)$. Такой выбор функции приспособленности обусловлен свойствами задачи и соответствующего метода локального поиска (подробнее см. в [18]). Будем обозначать критическую точку также через $ArgPLoc(\cdot)$, так что $\hat{\sigma}^r = ArgPLoc(u^r, v^r, w^r)$.

Точки популяции в ГА проходят специальным образом организованный процесс «развития» с помощью операторов, моделирующих процесс селекции, скрещивания (кроссовера) и мутации в течение определенного количества поколений (итераций) [19, 20]. Эти операторы были внедрены в «базовый» АГП, в результате чего в [18] был построен следующий

Гибридный алгоритм глобального поиска (ГАГП).

Пусть даны стартовая точка $\sigma^0 \in S$, последовательности $\{\tau_k\}, \{\delta_k\}$ ($\tau_k > 0, \delta_k > 0, k = 1, 2, \dots, \tau_k \downarrow 0, \delta_k \downarrow 0 (k \rightarrow \infty)$), множество $Dir = \{(\bar{u}^1, \bar{v}^1, \bar{w}^1), \dots, (\bar{u}^N, \bar{v}^N, \bar{w}^N) \in \mathbb{R}^{m+n+l} | (\bar{u}^r, \bar{v}^r, \bar{w}^r) \neq 0, r = 1, \dots, N\}$, числа $\xi_- \triangleq \inf(g, S)$ и $\xi_+ \triangleq \sup(g, S)$, вероятность мутации P_m , максимальное число поколений G_{max} и ε — точность решения задачи.

Шаг 0. Положить $k := 1, \bar{\sigma}^k := \sigma^0, \xi := \xi_-, \Delta\xi = (\xi_+ - \xi_-)/N$.

Шаг 1. Начиная с точки $\bar{\sigma}^k$ методом локального поиска найти τ_k -критическую точку $\sigma^k = (x^k, y^k, z^k, \alpha_k, \beta_k, \gamma_k) \in S$ в задаче (\mathcal{P}) . Положить $\zeta_k := \Phi(\sigma^k)$.

Шаг 2. Если $\zeta_k \geq -\varepsilon$, то стоп; найдено приближенное ε -равновесие по Нэшу в игре Γ_3 : $(x^k, y^k, z^k) \in NE(\Gamma_3, \varepsilon)$.

Шаг 3. По точкам $(\bar{u}^r, \bar{v}^r, \bar{w}^r) \in Dir$ построить точки (u^r, v^r, w^r) аппроксимации поверхности уровня функции $h(\cdot)$, $r = 1, \dots, N$, так что $h(u^r, v^r, w^r) = \xi + \Delta\xi \cdot (r - 1) + \zeta_k$, т. е. построить начальную популяцию точек Pop_k с поверхности уровня. Для каждой точки популяции вычислить значение функции приспособленности: $\zeta_r := PLoc(u^r, v^r, w^r)$, $r = 1, \dots, N$. Положить $p : \zeta_p \leq \zeta_r \forall r = 1, \dots, N$.

Шаг 4. Если для некоторого номера $\bar{r} \in \{1, \dots, N\}$ выполнено неравенство $\zeta_{\bar{r}} \geq -\varepsilon$, то стоп; найдено $(x^*, y^*, z^*, \alpha_*, \beta_*, \gamma_*) := ArgPLoc(u^{\bar{r}}, v^{\bar{r}}, w^{\bar{r}})$ — приближенное глобальное решение в задаче (\mathcal{P}) ; $(x^*, y^*, z^*) \in NE(\Gamma_3, \varepsilon)$.

Шаг 5. Положить $r_1 := Rand\{1, \dots, N\}, r_2 := Rand\{1, \dots, N\}$. Используя какую-либо процедуру кроссовера, по точкам $(u^{r_1}, v^{r_1}, w^{r_1})$ и $(u^{r_2}, v^{r_2}, w^{r_2})$ построить точки $(\tilde{u}, \tilde{v}, \tilde{w})$ и $(\check{u}, \check{v}, \check{w})$.

Шаг 6. Для точек $(\tilde{u}, \tilde{v}, \tilde{w})$ и $(\check{u}, \check{v}, \check{w})$ реализовать выбранную процедуру мутации с вероятностью P_m .

Шаг 7. Вычислить $\tilde{\alpha} := \alpha(\tilde{u}, \tilde{w}), \tilde{\beta} := \beta(\tilde{u}, \tilde{w}), \tilde{\gamma} := \gamma(\tilde{u}, \tilde{v}); \check{\alpha} := \alpha(\check{u}, \check{w}), \check{\beta} := \beta(\check{u}, \check{w}), \check{\gamma} := \gamma(\check{u}, \check{v})$. Положить $\tilde{\sigma} := (\tilde{u}, \tilde{v}, \tilde{w}, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}), \check{\sigma} := (\check{u}, \check{v}, \check{w}, \check{\alpha}, \check{\beta}, \check{\gamma})$. Вычислить также значения $\zeta := \Phi(\tilde{\sigma}), \check{\zeta} := \Phi(\check{\sigma}); \xi := g(\tilde{\sigma}), \check{\xi} := g(\check{\sigma})$ и построить две новые точки, лежащие на поверхностях уровня: $h(\bar{u}^{r_1}, \bar{v}^{r_1}, \bar{w}^{r_1}) = \xi + \zeta$ и $h(\bar{u}^{r_2}, \bar{v}^{r_2}, \bar{w}^{r_2}) = \check{\xi} + \check{\zeta}$.

Шаг 8. Вычислить $PLoc(\bar{u}^{r_1}, \bar{v}^{r_1}, \bar{w}^{r_1})$ и $PLoc(\bar{u}^{r_2}, \bar{v}^{r_2}, \bar{w}^{r_2})$. Положить

$$(\bar{u}^r, \bar{v}^r, \bar{w}^r) := \arg \max\{PLoc(\bar{u}^{r_1}, \bar{v}^{r_1}, \bar{w}^{r_1}), PLoc(\bar{u}^{r_2}, \bar{v}^{r_2}, \bar{w}^{r_2})\}.$$

Шаг 9. Если $PLoc(\bar{u}^r, \bar{v}^r, \bar{w}^r) \geq -\varepsilon$, то стоп; найдено приближенное ε -равновесие по Нэшу в игре Γ_3 .

$(x^*, y^*, z^*, \alpha_*, \beta_*, \gamma_*) := ArgPLoc(\bar{u}^r, \bar{v}^r, \bar{w}^r)$ — приближенное глобальное решение задачи (\mathcal{P}) .

Шаг 10. Если $PLoc(\bar{u}^r, \bar{v}^r, \bar{w}^r) > \zeta_p$, то обновить точку j популяции:
 $(u^p, v^p, w^p) := (\bar{u}^r, \bar{v}^r, \bar{w}^r)$.

Шаг 11. Если $k < G_{max}$, то $k := k + 1$ и перейти на шаг 3, иначе
стоп. $(u^*, v^*, w^*) := \arg \max\{PLoc(u^r, v^r, w^r) \mid (u^r, v^r, w^r) \in Pop_k, r =$
 $1, \dots, N\}$, $(x^*, y^*, z^*, \alpha_*, \beta_*, \gamma_*) := ArgPLoc(u^*, v^*, w^*)$ — полученное
решение задачи. #

Допустимая стартовая точка σ^0 , как и для «базового» алгоритма,
легко строится с помощью барицентров канонических симплексов и
свойства (2) [6–8, 18]. Точности τ_k и δ_k используются в ГАГП внут-
ри функции $PLoc(\cdot)$, где мы решаем линейаризованную задачу и осу-
ществляем локальный поиск. Поэтому упоминание о τ_k и δ_k на шагах
алгоритма отсутствует. Вспомогательные выпуклая линейаризованная
задача и задачи ЛП на шагах локального поиска могут быть решены
одним из классических методов линейного и квадратичного програм-
мирования (см., например, [11, 12]). Построение точек, принадлежащих
аппроксимации поверхности уровня, по заданным векторам из Dir и
на шаге 7 проводится аналитически (подробнее см. [6, 7]). При этом за-
метим, что эта процедура может быть реализована для любой точки
евклидова пространства (кроме 0), поэтому проблем с допустимостью
точек, полученных в результате применения операторов ГА, здесь не
возникает.

Отметим также необходимость специального вычисления значений
скалярных переменных задачи на шаге 7, поскольку аппроксимация по-
верхности уровня функции, задающей базовую невыпуклость в такого
типа задачах, строится только в пространстве трех векторных перемен-
ных. Кроме того, немаловажным элементом ГАГП в гексаматричных
играх (равно как и «базового» АГП) являются дополнительные крите-
рии останова (см. шаги 2, 4 и 9), которые определяют приближенное
глобальное решение задачи (\mathcal{P}) согласно свойствам редукции гексамат-
ричной игры.

Наконец, из всего широкого спектра возможных вариантов генети-
ческих операторов для реализации в ГАГП были выбраны процеду-
ры однородного кроссовера и равномерной мутации [19, 20], которые
осуществляются следующим образом. Пусть $r_1, r_2 \in \{1, \dots, N\}$ ($r_1 \neq$
 r_2) — номера особей для осуществления однородного кроссовера и $\kappa :=$
 $Rand[0, 1]$. Тогда $\forall j = 1, \dots, m + n + l$, если $\kappa < 0.5$ то $(\check{u}, \check{v}, \check{w})_j :=$
 $(u^{r_1}, v^{r_1}, w^{r_1})_j$, $(\tilde{u}, \tilde{v}, \tilde{w})_j := (u^{r_2}, v^{r_2}, w^{r_2})_j$, иначе
 $(\tilde{u}, \tilde{v}, \tilde{w})_j := (u^{r_1}, v^{r_1}, w^{r_1})_j$, $(\check{u}, \check{v}, \check{w})_j := (u^{r_2}, v^{r_2}, w^{r_2})_j$. Далее, пусть P_m —
вероятность осуществления мутации, и вследствие особенностей зада-
чи $\underline{K} = 0$, $\bar{K} = K > 0$. Для каждого $j = 1, \dots, m + n + l$ вычис-
ляются свои $\kappa_1 := Rand[0, 1]$ и $\kappa_2 := Rand[0, 1]$. Если $\kappa_1 < P_m$, то
 $(\check{u}, \check{v}, \check{w})_j := Rand[0, K]$. Если $\kappa_2 < P_m$, то $(\tilde{u}, \tilde{v}, \tilde{w})_j := Rand[0, K]$.

Предварительное численное тестирование разработанного алгоритма, проведенное в [18], показало работоспособность предложенного гибридного подхода к решению гексаматричных игр. Ниже подробно представим сравнительный вычислительный эксперимент по решению большого количества случайно сгенерированных игр различной сложности и размерности.

3 Вычислительный эксперимент

Программы, реализующие разработанный ГАГП, равно как и «базовый» АГП из [6–8], были реализованы в системе MATLAB 7.11.0.584 R2010b. При этом для возможности корректного сравнения счет проводился на том же компьютере, что и в [8], с процессором Intel Core i5-4690K CPU (3.5 GHz) и 16 Gb RAM. Вспомогательные задачи линейного и выпуклого квадратичного программирования решались с помощью подпрограмм «cplexlp» и «cplexqp» пакета IBM ILOG CPLEX 12.6.2 соответственно, с установками по умолчанию.

Наборы тестовых гексаматричных игр были случайно сгенерированы с помощью подпрограммы «sprand» системы MATLAB. Элементы матриц A_1 , A_2 , B_1 , B_2 , C_1 , и C_2 представляют собой целые псевдослучайные числа, равномерно распределенные на отрезке $[-(m+n+l)/10, (m+n+l)/10]$. Заполненность матриц была равна 0.1. При этом параметры подпрограммы генерации были установлены так, что при каждом запуске программы генерируется одна и та же псевдослучайная последовательность. Это также дает возможность корректно сравнивать результаты работы двух алгоритмов, поскольку для определенной размерности строится одна и та же серия гексаматричных игр.

Как и во время второго этапа вычислительного эксперимента из [8], с помощью ГАГП решались игры размерности от $(10 + 10 + 10)$ до $(100 + 100 + 100)$ ($m = n = l$), количество игр в серии зависело от размерности и изменялось от 10000 до 10 (Sr).

Таким образом, все условия эксперимента из [8] были сохранены. Напомним полученные в этом эксперименте результаты решения тестовых задач. Они приведены ниже в табл. 1 со следующими обозначениями: $SLoc$ — количество задач из серии, решенное только с помощью первичного локального поиска; Loc_{av} — среднее (на одну задачу серии) количество запусков локального поиска, потребовавшееся при глобальном поиске; LP_{av} и QP_{av} — среднее количество решенных задач ЛП и выпуклых квадратичных задач соответственно. Можно считать эти показатели некой «мерой» эффективности АГП наряду с T_{av} — средним временем решения одной задачи (в секундах). Наконец, uS — количество задач в серии, в которых не была достигнута заданная точность $\varepsilon = 10^{-3}$ приближенной ситуации равновесия по Нэшу.

Таблица 1. Тестирование «базового» АГП на сериях задач различной
размерности

m	Sr	$SLoc$	LP_{av}	Loc_{av}	QP_{av}	T_{av}	uS
10	10000	1325	200.06	14.75	3.11	0.21	0
15	10000	97	806.57	45.84	8.30	0.94	0
20	1000	1	6057.86	291.23	49.20	6.98	4
25	1000	0	8243.24	575.84	96.65	10.08	7
30	1000	0	14038.57	983.67	164.64	17.81	8
40	1000	0	29920.71	1741.52	290.94	42.49	11
50	100	0	40650.39	2077.65	346.98	58.55	1
60	100	0	58387.44	2584.34	431.37	91.39	1
70	100	0	119430.68	5777.18	963.46	215.88	2
80	10	0	29301.5	985.0	164.8	57.6	0
90	10	0	219911.8	3055.7	509.9	363.5	0
100	10	0	117174.9	3328.4	555.3	265.0	0

Напомним также, что локальный поиск в «базовом» АГП представлял собой последовательную реализацию шести возможных процедур [6–8, 22] типа «mountain climbing», получающихся, в частности, с помощью изменения порядка решения задач ЛП. И, как можно видеть из таблицы, в этом случае удалось решить с заданной точностью более 99.8% сгенерированных задач (всего 34 из 24330 задач остались нерешенными).

При реализации ГАГП после предварительных экспериментов прежде всего было уменьшено количество вызовов процедур в локальном поиске с 6 до 2. Стартовая допустимая точка и значения общих параметров выбирались, как и в [8].

Далее, выбор специфических «генетических» параметров (размера популяции (N), вероятности мутации (P_m) и количества поколений (G_{max})) проводился последовательно из наборов $\{m; (m+n+l); (m+n+l)*3; (m+n+l)*2\}$, $\{0.02; 0.01; 0.03; 0.04\}$ и $\{(m+n+l); (m+n+l)*3; (m+n+l)*5; (m+n+l)*\lfloor\sqrt{m}\rfloor\}$ соответственно. В результате было организовано 4 запуска ГАГП, после окончания каждого из них заново проводилось построение начальной популяции. Сама же начальная популяция строилась с помощью векторов множеств Dir , которые наиболее эффективно зарекомендовали себя в качестве векторов множеств направлений для построения аппроксимаций поверхностей уровня в «базовом» АГП, а именно: первые m точек этого множества выбирались из набора:

$$Dir1 = \{e^r \in \mathbb{R}^{m+n+l}, r = 1, \dots, m+n+l\},$$

где e^r — соответствующий вектор стандартного евклидова базиса. А

остальные выбирались случайным образом из множеств

$$Dir2 = \{(e^i, e^j, e^t), i = 1, \dots, m, j = 1, \dots, n, t = 1, \dots, l\},$$

$$Dir3 = \{(e^i + x^k, e^j + y^k, e^t + z^k), i = 1, \dots, m, j = 1, \dots, n, t = 1, \dots, l\},$$

где e^i, e^j, e^t — также векторы евклидоваго базиса соответствующей размерности, а (x^k, y^k, z^k) — компоненты текущей критической точки в задаче (\mathcal{P}) (с шага 1 ГАГП). При этом было реализовано два варианта ГАГП: в первом случае выбор из множества $Dir2$ происходил на 1-м и 3-м запусках ГАГП, а из $Dir3$ — на 2-м и 4-м (прямой шрифт в таблице); во втором случае — наоборот (из $Dir3$ на 1-м и 3-м; из $Dir2$ на 2-м и 4-м запусках — *курсивный шрифт в таблице*). В табл. 2, приведенной ниже, представлены результаты, полученные лучшим из этих двух вариантов. При этом в столбце $Par_{1,2,3,4}$ указано количество задач из серии, решенных с помощью соответствующего набора параметров N, P_m и G_{max} . Остальные обозначения и количество задач в сериях остались прежними.

Из табл. 2 прежде всего можно видеть, что количество нерешенных задач из 24330 сгенерированных сократилось с 34 до 11, т. е. более чем в три раза. Естественно, что при реализации ГАГП уменьшилось количество задач, решенных на небольших размерностях только одним локальным поиском, поскольку теперь внутри него действует только 2 процедуры, а не 6.

Тем не менее среднее время решения задач значительно уменьшилось. ГАГП проиграл «базовому» АГП только на размерностях 10, 80 и 100. В первом случае это объясняется тем, что, действительно, при повышении размерности в среднем эффективность ГАГП по времени увеличивается. А во втором и третьем случаях, равно как и для задач с размерностью 90, в сгенерированных сериях слишком мало задач, чтобы можно было судить о какой-то общей тенденции. На размерности 90 ГАГП выигрывает по времени более чем в 4.5 раза, а на размерностях 80 и 100 — проигрывает соответственно в 3.2 и 1.25 раза. Более надежные выводы можно сделать на задачах средних размерностей от 15 до 70. Здесь выигрыш ГАГП очевиден и колеблется от 1.1 до 3.4 раза.

Таблица 2. Тестирование ГАГП на тех же сериях задач

m	$SLoc$	LP_{av}	Loc_{av}	QP_{av}	$Par_{1,2,3,4}$	T_{av}	uS
10	591	195.89	21.36	10.74	(9399;8;2;0)	0.31	0
15	67	497.82	40.48	20.78	(9796;126;8;2)	0.86	1
20	0	1197.91	85.68	45.15	(943;49;8;0)	2.08	0
25	0	2548.52	163.97	87.37	(904;83;9;2)	4.51	2
30	0	4454.42	276.03	146.47	(842;130;24;2)	8.62	2
40	0	11015.89	578.68	306.00	(739;213;34;8)	23.76	6
50	0	13405.28	744.73	400.79	(65;33;1;1)	30.23	0
60	0	28232.14	1340.10	711.88	(56;38;2;4)	68.14	0
70	0	29017.16	1367.57	725.96	(42;52;6;0)	78.22	0
80	0	62126.4	2469.3	1258.1	(1;7;2;0)	183.69	0
90	0	19781.7	944.2	489.1	(6;4;0;0)	80.04	0
100	0	85091.4	3607.6	1896.2	(0;8;2;0)	329.09	0

Можно заметить также, что при повышении размерности происходит «смещение» доли задач, для решения которых требуются дополнительные запуски ГАГП. Если для задач небольших размерностей более 90% решаются на первом запуске, то для задач размерности 100 уже ни в одной из них равновесие на первом запуске найти не удастся. Наконец, первый вариант ГАГП оказался более эффективным по сравнению со вторым, поскольку показал лучшие по качеству результаты в 7 сериях из 12-ти.

Итак, на основании вышесказанного можно сделать окончательный вывод о преимуществе разработанного ГАГП над «базовым» АГП на достаточно широком спектре случайно сгенерированных задач различной размерности.

В завершение статьи (табл. 3) представим результаты решения «усложненных» задач, на которых «базовый» алгоритм показал неприемлемые результаты (до 50% нерешенных задач в серии). При этом была использована модификация разработанного алгоритма, заключающаяся в выборе не m , а до $m + n + l$ точек из набора $Dir1$ при построении начальной популяции. Эта модификация показала себя еще более эффективно во время решения задач таблиц 1 и 2.

Здесь заполненность матриц была повышена до 0.5 для задач размерностей 10 и 15, с постепенным уменьшением до 0.15 для задач высоких размерностей. Отметим, что этим вариантом алгоритма удалось решить задачи вплоть до размерности 200. Обозначения в таблице остались прежние, при этом из нее был исключен столбец $SLoc$, поскольку вследствие сложности новых задач одним локальным поиском удалось решить только 16 из 10000 задач с 10 стратегиями у каждого из игроков.

Таблица 3. Тестирование ГАГП на усложненных задачах

m	Sr	LP_{av}	Loc_{av}	QP_{av}	$Par_{1,2,3,4}$	T_{av}	uS
10	10000	1076.80	60.14	31.20	(8817;1066;75;10)	1.58	16
15	1000	4309.67	165.62	85.75	(732;238;21;2)	6.84	7
20	1000	10446.60	419.75	215.83	(571;337;60;8)	17.38	24
25	1000	22649.31	819.83	422.18	(410;387;160;18)	36.20	25
30	100	25337.93	1106.68	567.93	(34;41;21;0)	47.27	4
40	100	56583.75	2368.12	1217.07	(9;28;33;25)	114.16	9
50	100	82016.14	3687.08	1897.53	(14;41;28;6)	170.48	11
60	10	79521.8	3225.1	1665.3	(2;3;5;0)	185.59	0
70	10	134172.0	5280.4	2707.8	(1;4;3;1)	366.06	1
80	10	151104.9	6715.2	3601.3	(0;3;5;2)	517.18	0
90	10	220415.0	9677.2	5065.0	(1;3;3;1)	894.40	2
100	1	501558	23425	12242	(0;0;0;1)	2355.30	0
120	1	122514	4095	2485	(0;1;0;0)	618.80	0
150	1	50003	1782	944	(1;0;0;0)	431.32	0
175	1	164588	8041	4314	(0;1;0;0)	2085.30	0
200	1	654850	23544	12315	(0;0;1;0)	8576.27	0

По среднему времени решения задач, числу запусков локального поиска и количеству решенных вспомогательных задач из табл. 3 можно видеть, что «усложненные» задачи труднее для алгоритма примерно в 5–8 раз. Количество нерешенных с точностью 10^{-3} задач остается достаточно низким (не решены с заданной точностью 99 (менее 1 %) из 13345 сгенерированных задач). При этом в 93 из этих 99 задач удалось приблизиться к нулю с точностью 10^{-1} , так что здесь получено просто более грубое приближенное равновесие по Нэшу. Также сохраняется эффект «смещения» доли задач, для решения которых требуются дополнительные запуски ГАГП. И более эффективным остается первый вариант алгоритма (в 12 сериях из 16).

В целом можно сделать вывод, что гибридный подход к решению гексаматричных игр оказался достаточно эффективным, хотя и сохраняется необходимость его совершенствования с целью дальнейшего повышения размерности и заполненности матриц в сгенерированных задачах.

Заключение

В работе представлены результаты детализации гибридного алгоритма, разработанного в [21] и предназначенного для поиска ситуаций равновесия по Нэшу в гексаматричных играх. Подробно описан выбор его параметров, а также процесс организации и итоги обширного вычислительного эксперимента.

По результатам эксперимента можно сделать вывод, что ГАГП позволяет быстрее и качественнее, чем «базовый» алгоритм [8], решать

гексаматричные игры, особенно это касается игр с повышенной размерностью и заполненностью матриц. Дальнейшее совершенствование алгоритма будет направлено, в частности, на модификацию блока локального поиска, чтобы можно было решать еще более сложные задачи данного класса.

Литература

1. Петросян Л. А., Зенкевич Н. А., Семина Е. А. Теория игр. Москва: Высш. школа, 1998. 304 с. Текст: непосредственный.
2. Pang J.-S. Three modeling paradigms in mathematical programming // *Mathematical Programming, Ser. B.* 2010. Vol. 124, №2. P. 297–323. DOI: 10.1007/s10107-010-0395-1.
3. Орлов А. В., Стрекаловский А. С. О поиске ситуаций равновесия в биматричных играх // *Автоматика и телемеханика.* 2004. № 2. С. 55–68. Текст: непосредственный.
4. Стрекаловский А. С., Орлов А. В. Биматричные игры и билинейное программирование. Москва: ФИЗМАТЛИТ, 2007. 224 с. Текст: непосредственный.
5. Стрекаловский А. С., Энхбат Р. Полиматричные игры и задачи оптимизации // *Автоматика и телемеханика.* 2014. № 4. С. 51–66. DOI: 10.1134/S0005117914040043. Текст: непосредственный.
6. Орлов А. В., Батбилэг С. Олигополистический банковский сектор Монголии и полиматричные игры трех лиц // *Известия Иркутского государственного университета. Сер. Математика.* 2015. Т. 11. С. 80–95. Текст: непосредственный.
7. Orlov A. V., Strekalovsky A. S., Batbileg S. On computational search for Nash equilibrium in hexamatrix games // *Optimization Letters.* 2016. V. 10(2). P. 369–381. DOI: 10.1007/s11590-014-0833-8.
8. Orlov A. V. Finding the Nash equilibria in randomly generated hexamatrix games // *Proc. of the 14th Intern. Symposium on Operational Research (SOR'17, Bled, Slovenia, September 27-29, 2017)* / Ed. by L. Zadnik Stirn et. al. Ljubljana: Slovenian Society Informatika, Section for Operational Research, 2017. P. 507–512.
9. Strekalovsky A. S. On Solving Optimization Problems with Hidden Nonconvex Structures // *Optimization in Science and Engineering* / Ed. by T.M. Rassias et al. New-York: Springer, 2014. P. 465–502. DOI: 10.1007/978-1-4939-0808-0_23.
10. Strekalovsky A. S. On a Global Search in D.C. Optimization Problems // *Optimization and Applications. OPTIMA 2019. Communications in Computer and Information Science* / Ed. by M. Jacimovic et al. Cham: Springer, 2020. V. 1145. P. 222–236. DOI: 10.1007/978-3-030-38603-0_17.
11. Васильев Ф. П. Методы оптимизации. Москва: Факториал-пресс, 2002. 824 с. Текст: непосредственный.
12. Bonnans J.-F., Gilbert J. C., Lemarechal C., Sagastizabal C. A. Numerical optimization: theoretical and practical aspects. Berlin-Heidelberg: Springer, 2006. 494 p.

13. Стрекаловский А. С., Орлов А. В. Линейные и квадратично-линейные задачи двухуровневой оптимизации. Новосибирск: Изд-во СО РАН, 2019. 262 с. Текст: непосредственный.
14. Orlov A. V. The global search theory approach to the bilevel pricing problem in telecommunication networks // Computational Aspects and Applications in Large Scale Networks / Ed. by V.A. Kalyagin et al. Cham: Springer, 2018. P. 57–73. DOI: 10.1007/978-3-319-96247-4_5
15. Strekalovsky A. S., Orlov A. V. Global Search for Bilevel Optimization with Quadratic Data // Springer Optimization and Its Applications. Bilevel optimization: advances and next challenges / Ed. by S. Dempe, A. Zemkoho. Cham: Springer, 2020. V. 161. P. 313–334. DOI: 10.1007/978-3-030-52119-6_11
16. Audet C., Belhaiza S., Hansen, P. Enumeration of all the extreme equilibria in game theory: bimatrix and polymatrix games // Journal of Optimization Theory and Applications. 2006. Vol. 129, № 3. P. 349–372.
17. Golshteyn E., Malkov U., Sokolov N. The Lemke-Howson Algorithm Solving Finite Non-Cooperative Three-Person Games in a Special Setting // DEStech Transactions on Computer Science and Engineering. 2019. Suppl. vol. P. 265–272.
18. Orlov A. V. Hybrid Global Search Algorithm with Genetic Blocks for Solving Hexamatrix Games // Известия Иркутского государственного университета. Серия Математика. 2022. Т. 41. С. 40–56. DOI: 10.26516/1997-7670.2022.41.40.
19. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. New York: Springer-Verlag, 1994. 340 p.
20. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы. Москва: ФИЗМАТЛИТ, 2010. 368 с. Текст: непосредственный.
21. Орлов А. В. Гибридный генетический алгоритм глобального поиска оптимистических решений в задачах двухуровневой оптимизации // Вестник Бурятского гос. ун-та. 2013. Вып. 9. С. 25–32. Текст: непосредственный.
22. Orlov A. V., Strekalovsky A. S. On a Local Search for Hexamatrix Games // CEUR Workshop Proceedings. DOOR-SUP 2016. 2016. V. 1623. P. 477–488. <https://ceur-ws.org/Vol-1623/>

Статья поступила в редакцию 24.04.2023; одобрена после рецензирования 14.06.2023; принята к публикации 23.06.2023.

COMPUTATIONAL STUDY OF HYBRID GLOBAL SEARCH ALGORITHM IN HEXAMATRIX GAMES

Andrei V. Orlov

Cand. Sci. (Phys. and Math.), Assoc. Prof.,
Senior Researcher, Head of the Laboratory
Matrosov Institute for System Dynamics and Control Theory SB RAS
134 Lermontov St., Irkutsk 664033, Russia

Abstract. The paper addresses a special hybrid approach to the development of methods for finding Nash equilibrium points in three-person polymatrix games (hexamatrix games). On the one hand, this approach is based on the Global Search Theory created by A.S. Strekalovsky for nonconvex optimization problems with d.c. functions (representable as a difference of two convex functions). On the other hand, to implementation of one of the key stages of the global search — constructing an approximation of the level surface of a convex function — operators of genetic algorithms are used.

After the description of the hybrid approach, it is described in detail about the organization and carrying out of an extensive computational experiment comparing the hybrid algorithm with the "basic" global search algorithm developed earlier. The results of a computational experiment on a series of randomly generated problems are presented, which confirm the efficiency of the proposed hybrid approach to solving hexamatrix games.

Keywords: polymatrix games of three players, hexamatrix games, Nash equilibrium, Global Search Theory, local search, global search algorithm, level surface approximation, genetic operators, hybrid algorithm, computational experiment.

For citation

Orlov A. V. Computational Study of Hybrid Global Search Algorithm in Hexamatrix Games. Bulletin of Buryat State University. Mathematics, Informatics. 2023. N. 2. P. 14–29.

The article was submitted 24.04.2023; approved after reviewing 14.06.2023; accepted for publication 23.06.2023.