

В.И. ЗИНОВ, В.М. КАРТАК, Ю.И. ВАЛИАХМЕТОВА
**РЕШЕНИЕ МНОГОКРИТЕРИАЛЬНОЙ ЗАДАЧИ
РАЦИОНАЛЬНОГО РАЗМЕЩЕНИЯ НЕСУЩИХ СТЕН
С ПОМОЩЬЮ ГЕНЕТИЧЕСКОГО АЛГОРИТМА**

Зинов В.И., Картак В.М., Валиахметова Ю.И. Решение многокритериальной задачи рационального размещения несущих стен с помощью генетического алгоритма.

Аннотация. Рациональное размещение несущих стен остается сложной и малоизученной задачей, несмотря на большое количество алгоритмов и моделей решения схожей задачи размещения колонн. Основными факторами сложности являются большое количество вариантов решения, большое время, требуемое для оценки полученного плана размещения на деформации, а также – многокритериальный характер. Кроме нелинейного критерия оценки деформаций требуется минимизировать протяженность несущих стен и количество различных уникальных типоразмеров. В статье предложена модель задачи размещения несущих стен, разбивающая стены на функциональные участки с некоторым шагом и учитывающая перечисленные критерии. При этом соседствующие участки стен одной функциональности объединяются в сегменты. Комбинаторный вид разбиения стен в модели задачи позволяет применить генетические алгоритмы для её решения. Поэтому была предложена новая схема многокритериального генетического алгоритма, содержащая метрики расчёта разнообразия фенотипа и генотипа популяции. Представлены модификации операторов скрещивания, мутации и селекции, учитывающие сегментальный вид генотипа стен. Анализ сравнения разработанного алгоритма с другими многокритериальными генетическими алгоритмами показал, что, несмотря на в 2 раза более длительное время выполнения, разработанный алгоритм находит в среднем в 3 раза больше недоминируемых решений на общем множестве, особенно с меньшим значением оценки деформаций. Предложенная модель разительно отличается от предлагаемых ранее с точки зрения оперирования с деформациями в системах опоры-плита, лишь сравнивая между собой планы размещения, а не рассчитывая точные оценки армирования, что зачастую бывает излишне на ранних этапах. Предложенная схема генетического алгоритма увеличивает количество найденных недоминируемых решений без потери их разнообразия, жертвуя временем выполнения, и может быть использована для решения других многокритериальных задач с учётом указанных особенностей. Разработанный алгоритм был легко интегрирован в программное средство поддержки принятия решений на базе САПР и может быть использован на практике специалистами.

Ключевые слова: оптимизация в проектировании зданий, задача размещения стен, многокритериальный генетический алгоритм, комбинаторная оптимизация, сравнительная оценка деформаций плит.

1. Введение. Расположение несущих стен – одна из важнейших характеристик проекта здания. Несущие стены воспринимают нагрузки от плит перекрытий, распределяя их между собой. Плиты перекрытий прогибаются, что определяет их требуемое арматурное усиление. Расчет деформаций производится на программно-расчётных комплексах (ПРК). В силу необходимости учёта множества условий и факторов такие расчеты сложны и выполняются долго.

Сложность подбора расположения стен складывается из ряда причин. Во-первых, из сложности расчета явной функциональной связи между планом стен и деформациями, что в том числе приводит к интуитивному выбору расположения стен на основе опыта. Во-вторых, из-за большого числа вариантов планов стен и отсутствия автоматизированной системы их анализа, что не позволяет в полной мере оценить планы по различным критериям и выбрать наилучший вариант среди всего множества. В-третьих, из-за больших временных затрат на этапе расчета деформаций в ПРК, что исключает возможность автоматизированного просчёта множества вариантов. Таким образом, задача рационального подбора плана несущих стен является сложной задачей с большим числом вариантов и, по крайней мере, одним нелинейным целевым критерием – оценкой деформаций плит. Был проведен анализ научных работ, посвященных проблемам оптимизации расположения структурных конструкций здания.

В работе [1] предлагается модель размещения колонн и балок на основе приоритетного подхода. Для оптимизации их расположения по критерию стоимости применены генетические алгоритмы и алгоритмы роя частиц. Оценка прочности конструкций производится по модели балок Эйлера. Предложенная модель учитывает множество ограничений и позволяет получить допустимые планы колонн, однако подходит только для малоэтажных зданий и приспособлена для размещения колонн, но не стен. Похожая модель на основе генетических алгоритмов применяется в работе [2].

Один из исследователей проблемы оптимизации размещения конструкций – Пезман Шарафи (Pezhman Sharafi). В его докторской работе [3] предложена методология оптимизации конструкций, в т.ч. балок, колонн и плит, на основе популяционных алгоритмов. В другой работе [4] представлен эвристический алгоритм поиска оптимального регулярного размещения колонн. В качестве целевой функции принята стоимость армирования плиты, которая зависит от пролетов между колоннами и определяется по расчёту изгибающих моментов методом заменяющих рам. Недостатком подхода является ограничение метода заменяющих рам, который подходит только для регулярных схем опор, что ограничивает область применения. В статье [5] разработана модель поиска оптимального размещения, размеров и армирования колонн. Целевой критерий – стоимость конструкций, а ограничения – условие рассчитываемой прочности структур на изгиб. Тем не менее, область применения также ограничивается регулярными планами колонн.

В работе [6] предложен метод оптимизации колонн с помощью множителей Лагранжа. Целевой критерий – жесткость конструкций, рассчитываемая по модулю Юнга. Позиции для колонн заданы дискретно, неизвестное – количество колонн. В работе [7] предложена модель оптимизации размещения колонн на основе расчёта градиента объёма бетона в плите перекрытия над колоннами при ограничении прогиба плиты, напряжения и изгибающего момента, что повышает вычислительную эффективность, однако сильно усложняет модель.

В работе [8] колонны и балки представлены в виде планарного графа. Для расчёта нагрузок на конструкции предложен эвристический метод, распределяющий вес плиты на колонны и балки. Реализованная программная система позволяет сравнивать автоматизированный план и план, разработанный специалистами. В статье [9] рассмотрена задача оптимизации размещения несущих стен. Для анализа геометрии плиты решается задача о разбиении многосвязного ортогонального полигона. Толщина стен константна. Для решения предлагается эвристический алгоритм поиска рационального плана размещения стен с правилами, ограничивающими поиск в пределах области допустимости. Однако модель не содержит практико-ориентированного инструмента анализа деформаций и не приспособлена для многокритериального решения.

В работе [10] описывается интегрированная в BIM-процесс проектирования процедура задания зон дополнительного армирования в перекрытии на основе расчётных значений моментов напряжений, полученных с помощью метода конечных элементов. Несмотря на простоту практического применения, ключевой недостаток подхода – отсутствие достаточной гибкости при изменении исходных данных: изменив плиту перекрытия или её опоры, весь процесс расчётов придется повторить заново. Подобный подход оптимизации каркасов с применением средств глубокого обучения применен в [11]. Кроме общепринятых методов оценки прочности, например, упомянутым методом заменяющих рам, разрабатываются также оценки на основе нейросетевого подхода [12]. ИНС также могут применяться в качестве инструмента генерации всего плана конструкций [13].

В большинстве работ решается задача оптимизации размещения колонн, но не стен. Причина этого – в особенностях проектирования: колонны зачастую располагают по сетке с регулярным шагом, что упрощает задачу, но ограничивает область применения предлагаемых методов. Кроме того, чаще всего для оценки деформаций в плитах применяются рассчитываемые значения, что усложняет структуру модели и увеличивает вычислительную сложность. Однако на этапе планирования опорных конструкций точные оценки не обязательны;

подход оценки, направленный на сравнение планов по деформациям между собой, может быть более быстрым и рациональным. Поэтому цель данной работы состоит в разработке модели и алгоритма решения задачи рационального размещения несущих стен с применением сравнительного подхода оценки деформаций в плитах по планам стен.

В данной статье описываются научные результаты реализации этой цели. В главе 2 представлена новая математическая модель задачи рационального размещения несущих стен в комбинаторной многокритериальной постановке. Критерии задачи – протяженность несущих стен, количество различных длин несущих стен и оценка деформаций, которая выражается через разработанную ранее модель [14]. В главе 3 описана схема многокритериального генетического алгоритма (МГА), решающая задачу рационального размещения несущих стен на основе предлагаемой модели. Т.к. стены в модели представлены в комбинаторном виде с однородными сегментами, то в предлагаемом МГА гены также рассматриваются сегментно. Поэтому предлагаются новые алгоритмы мутации и скрещивания, которые оперируют не с отдельными генами, а со связанными цепочками генов. Для того чтобы учесть особенности и условия задачи, был разработан новый алгоритм оператора селекции. В главе 4 проведено сравнение эффективности решения задачи рационального размещения несущих стен разработанным МГА и аналогами на основе предложенной модели. Разработанные модель и алгоритмы реализуются в виде автоматизированной системы поддержки принятия решений.

2. Модель задачи рационального размещения стен. Дано множество несущих стен $T = \{T_1, \dots, T_N\}$, где $N \in \mathbb{Z}$ – количество стен. Каждая i -ая стена характеризуется как $T_i = \langle (x_i, y_i), w_i, l_i, n_i \rangle$, где $x_i \in \mathbb{Z}$, $y_i \in \mathbb{Z}$ – координаты стены; $w_i \in \mathbb{Z}$ – толщина стены; $l_i \in \mathbb{Z}$ – длина стены; $n_i \in \mathbb{Z}$ – количество участков разбиения в стене; $i = 1, \dots, N$. Участки разбиения в стене определены как $t_{ij} = \langle l_{ij} \rangle$, где $l_{ij} \in \mathbb{Z}$ – длина j -го участка; $j = 1, \dots, n_i$, $i = 1, \dots, N$.

Введём переменные $X_{ij} \in \{0, 1\}$, отражающие функциональное свойство (ФС) j -го участка i -ой стены: если $X_{ij} = 0$, то участок является ненесущим; если $X_{ij} = 1$, то участок является несущим.

Соседствующие участки стены с одинаковым ФС объединяются в сегменты. Планом опор назовём множество планов сегментов $Q = \{Q_1, \dots, Q_N\}$. Пусть для каждой стены задано количество сегментов $m_i \in \mathbb{Z}$, $1 \leq m_i \leq n_i$. Тогда каждый i -ый план сегментов задан как множество сегментов $Q_i = \{Q_{ik}: k = 1, \dots, m_i\}$. Пусть для каждого

сегмента задано количество участков $\eta_{ik} \in \mathbb{Z}$, $1 \leq \eta_{ik} \leq n_i$. Тогда каждый k -ый сегмент i -ой стены задан множеством индексов его участков:

$$\forall k = \overline{1, \dots, m_i} : Q_{ik} = \{j : j = \sum_{k^*=1}^{k-1} \eta_{ik^*} + 1, \dots, \sum_{k^*=1}^{k-1} \eta_{ik^*} + \eta_{ik}\}.$$

Зададим начальное ФС в i -ой стене как $\chi_i \in \{0, 1\}$, а ФС отдельного сегмента – $\chi_{ik} \in \{0, 1\}$. Принимается, что сегменты чередуются по ФС. Тогда ФС любого сегмента может быть определено как $\chi_{ik} = (\chi_i + k + 1) \bmod 2$, $\forall k = 1, \dots, m_i$. Обозначим значение ФС j -ого участка i -ой стены по плану опор Q как:

$$\forall j = \overline{1, \dots, n_i} : X_{ij}(Q) = \chi_{ik} \Leftrightarrow j \in Q_{ik}. \quad (1)$$

Длина k -ого сегмента в i -ой стене определяется как:

$$length(Q_{ik}) = \sum_{j \in Q_{ik}} l_{ij}.$$

Длины всех сегментов, выстроенных по плану опор Q , должны быть не меньше минимумов длин по соответствующему ФС:

$$\forall i = 1, \dots, N, \forall k = 1, \dots, m_i : length(Q_{ik}) \geq \begin{cases} l^0, & \text{если } \chi_{ik} = 0 \\ l^1, & \text{если } \chi_{ik} = 1 \end{cases}, \quad (2)$$

где l^0 и l^1 – минимальные длины ненесущих и несущих сегментов.

Требуется, чтобы значения трёх целевых функций по построенному плану опор Q стремились к минимуму:

$$f_L(Q) = \frac{\sum_{i=1}^N \sum_{k=1}^{m_i} (length(Q_{ik}) \cdot \chi_{ik})}{\sum_{i=1}^N l_i} \rightarrow \min, \quad (3)$$

где $f_L(Q)$ – функция протяженности несущих сегментов, нормированная относительно общей длины стен;

$$f_U(Q) = \frac{|\{length(Q_{ik}) : \chi_{ik} = 1; k = \overline{1, \dots, m_i}, i = \overline{1, \dots, N}\}|}{|\{Q_{ik} : \chi_{ik} = 1; k = \overline{1, \dots, m_i}, i = \overline{1, \dots, N}\}|} \rightarrow \min, \quad (4)$$

где $f_U(Q)$ – функция количества типов длин несущих сегментов, нормированная относительно общего количества несущих сегментов;

$$f_{Sp}(Q) = V(\overline{Sp}) = \frac{\sum_{Sp \in \overline{Sp}} V(Sp)}{|\overline{Sp}|} \rightarrow \min, \quad (5)$$

где $f_{Sp}(Q)$ – функция оценки деформаций плит по плану опор; $V(\overline{Sp}) \in [0, 1]$ – оценка деформации плиты, описанная в работе [14]; $\overline{Sp} = \{Sp_1, \dots, Sp_u\}$ – множество пролетов, $u \in \mathbb{Z}$ – количество пролетов; $Sp = \langle (x_{Sp}, y_{Sp}), (w_{Sp}, l_{Sp}) \rangle$ – пролет в плите, (x_{Sp}, y_{Sp}) – координаты пролета, (w_{Sp}, l_{Sp}) – размеры пролета; $V(Sp) \in [0, 1]$ – оценка деформаций плиты в пролете $Sp \in \overline{Sp}$. Оценка деформации плиты в пролете задана как:

$$V(Sp) = \hat{V}(Sp) + Penalty(Sp),$$

где $\hat{V}(Sp) \in [0, 0.5]$ – основная оценка пролета, $Penalty(Sp) \in [0, 0.5]$ – штрафное слагаемое пролета. Штрафное слагаемое определяется как:

$$Penalty(Sp) = \begin{cases} 0.5, & \text{если } size(Sp) > \max_{size}, \\ 0, & \text{иначе} \end{cases}$$

где $size(Sp)$ – размер пролета, \max_{size} – допустимый размер пролета. Основная оценка пролета $\hat{V}(Sp)$ определена следующим образом:

$$\hat{V}(Sp) = \frac{(1 + |\phi - 13/18|)^{-(Pr+2/3)^2} \cdot S(Sp)^{1+Pr^2} \cdot (\log_4 S(FSI))^2}{2 \cdot (\log_4 S(Sp))^{1+Pr^2} \cdot S(FSI)^2},$$

где $\phi \in (0, 1]$ – соотношение сторон пролета, $S(Sp) \in \mathbb{Z}$ – площадь пролета Sp , $S(FSI) \in \mathbb{Z}$ – площадь плиты, FSI – прямоугольная плита, Pr – эвристический параметр расположения опор (максимальное расстояние между двумя опорами вдоль периметра MPD [14]).

Определим переход между моделью рационального размещения стен (план опор Q) и моделью определения пролётов перекрытия [14] (препятствия Ob). Пусть позиция сегмента k в стене i задана как:

$$pos(Q_{ik}) = (x_i, y_i) + dir(i) \cdot \sum_{k*=1}^{k-1} (length(Q_{ik*})),$$

где $dir(i) \in \{(0, 1), (1, 0)\}$ – направление стены i . Определим множество препятствий по стенам как $Ob = \{Ob_i\}$, где множество препятствий по i -ой стене определяется как:

$$Ob_i = \{\langle pos(Q_{ik}), (length(Q_{ik}), w_i) \rangle \mid \chi_{ik} = 1, k = \overline{1, ..., m_i}\}.$$

Тогда модели могут быть использованы совместно: результаты модели детерминации пролётов используются в целевом критерии оценки деформации в модели рационального размещения стен (рисунок 1):

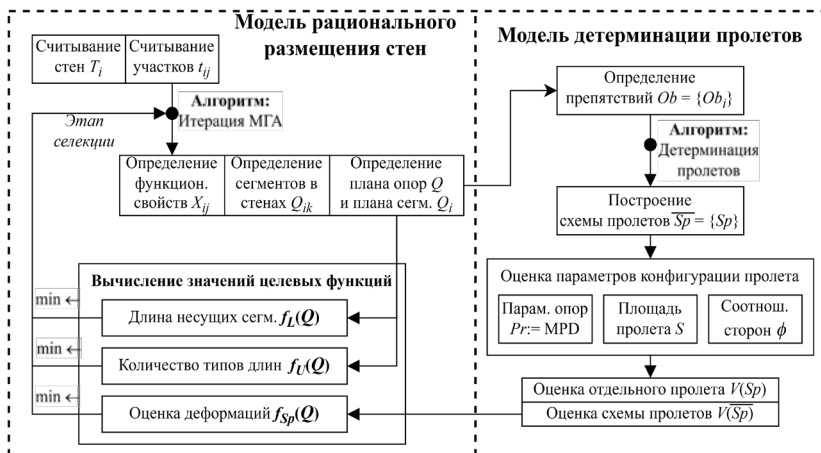


Рис. 1. Схема связи моделей детерминации пролётов и размещения стен

Описанная математическая модель позволяет формализовать параметры задачи и, в дальнейшем, разработать алгоритмы решения. Таким образом, представленная математическая модель объединяет подзадачи проблемы рационального размещения несущих стен и системно структурирует задачу. Далее представлены алгоритмы многокритериального анализа, основанные на предлагаемой модели.

3. Многокритериальный генетический алгоритм.

Многокритериальный генетический алгоритм (МГА) – это способ многокритериальной оптимизации, применяющий для поиска решения

методы случайного подбора, изменения и комбинирования вариантов решений. МГА применяются в задачах поддержки принятия решений.

Пусть на итерации МГА была получена популяция $\Lambda = \{Q^1, \dots, Q^\Omega\}$, где Ω – мощность популяции, Q^ζ – план опор, полученный ζ -ой особью, $\zeta = 1, \dots, \Omega$. Для решения задачи рационального размещения несущих стен предлагается новая схема МГА (рисунок 2). Схема содержит три метрики для оценки особей популяции: метрику различности M_{div} – оценивает расстояние между особями в пространстве критериев; метрику схожести M_{sim} – оценивает схожесть между генотипами особей по расстоянию Хэмминга; и метрику элитарности M_{eli} – оценивает степень доминирования по Парето между особями. Также используются ранг по разнообразию R_{div} для мутации и скрещивания и дискриминированный ранг по элитарности R_{dis} для селекции. Далее перечисленные метрики и ранги рассматриваются подробнее.

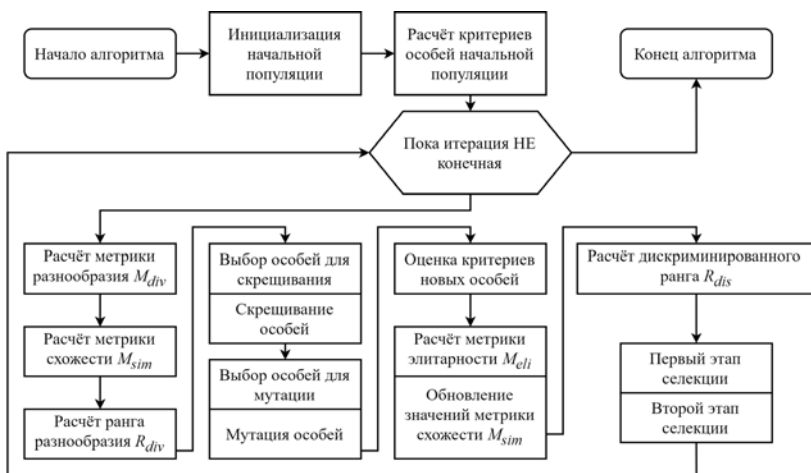


Рис. 2. Схема разработанного многокритериального генетического алгоритма

3.1. Метрики оценки и ранжирования популяции. Расчет метрики различности $M_{div} \in \mathbb{R}$ оценивает расстояние между особями в пространстве критериев. Наиболее подходящий алгоритм расчёта метрики различности описан в работе [15]. Для особи, находящейся на наибольшем суммарном отдалении от других особей, оценка будет ζ наименьшей. Метрика различности M_{div} для особи $Q^\zeta \in \Lambda$ задана как:

$$\forall Q^\theta \in \Lambda : d(Q^\zeta, Q^\theta) = \sqrt{\sum_{f \in \{f_L, f_U, f_{Sp}\}} (f(Q^\zeta) - f(Q^\theta))^2};$$

$$\beta(Q^\zeta, Q^\theta) = 1 - \frac{d(Q^\zeta, Q^\theta)}{\min(\{d(Q^\zeta, Q^{\theta*}) : \forall Q^{\theta*} \in \Lambda \setminus \{Q^\zeta\}\})};$$

$$M_{div}(Q^\zeta) = \sum_{\forall Q^\theta \in \Lambda \setminus \{Q^\zeta\}} \beta(Q^\zeta, Q^\theta). \quad (6)$$

Метрика схожести $M_{sim} \in (0, 1]$ применена в работе [16] и оценивает количество схожих по генотипу особей. Для каждой особи рассчитывается расстояние Хэмминга до генотипов других особей. Тогда если расстояние до одной из особей меньше порога, то дискриминация ранга по метрике схожести увеличивается. Обозначим расстояние Хэмминга между генотипами особей $Q^\zeta \in \Lambda$ и $Q^\theta \in \Lambda$ в соответствии с формулой индекса сегмента для участка (1) как:

$$d_H(Q^\zeta, Q^\theta) = \sum_{i=1}^N \sum_{j=1}^{n_i} |X_{ij}(Q^\zeta) - X_{ij}(Q^\theta)|.$$

Пусть $sr \in (0, 1]$ – параметр дискриминации особи за схожесть, $cr \in \mathbb{Z}$ – параметр максимальной различности двух схожих планов. Тогда определим метрику схожести M_{sim} для некоторой особи Q^ζ как:

$$M_{sim}(Q^\zeta) = \prod_{Q^\theta \in \Lambda \setminus \{Q^\zeta\}} \begin{cases} 1, & \text{если } d_H(Q^\zeta, Q^\theta) > cr \\ sr + (1 - sr) \cdot d_H(Q^\zeta, Q^\theta) / cr, & \text{иначе} \end{cases}. \quad (7)$$

Метрика элитарности $M_{eli} \in \mathbb{Z}$ оценивает особей по степени Парето-доминируемости: если особь не доминируема в популяции, то ей присваивается ранг 1; если особь доминируема только особями с рангом 1, то ей присваивается ранг 2, и т.д. Для этого вычисляются показатели доминирования между парами особей (Q^ζ и $Q^\theta \in \Lambda$):

$$\alpha(Q^\zeta, Q^\theta) = \begin{cases} 1, & \text{если } f(Q^\zeta) \leq f(Q^\theta) \\ -1, & \text{если } f(Q^\zeta) \geq f(Q^\theta), \forall f \in \{f_L, f_U, f_{Sp}\}. \\ 0, & \text{иначе} \end{cases}$$

Алгоритм расчёта метрики элитарности M_{eli} для особей популяции Λ представлен на листинге 1.

```

1.  $v := 1$  – ранг;  $\omega := 1$  – индекс подмножества;  $\Lambda^* := \Lambda$  – временный список;
2. while  $\Lambda^* \neq \emptyset$  do
3.    $\Lambda^\omega := \emptyset$ ;
4.   for  $\zeta \in (1 : |\Lambda^*|)$  do
5.     if  $\nexists Q^\theta \in \Lambda^* : \alpha(Q^\zeta, Q^\theta) = -1$  then
6.        $M_{eli}(Q^\zeta) := v$ ;
7.        $\Lambda^\omega.add(Q^\zeta)$ ;
8.     end if;
9.   end for;
10.   $\Lambda^* := \Lambda^* \setminus \Lambda^\omega$ ;
11.   $v := v + 1$ ;  $\omega := \omega + 1$ ;
12. end while;
```

Листинг 1. Псевдокод алгоритма метрики элитарности.

Таким образом, указанный алгоритм позволяет получить ранг по метрике элитарности для всех особей популяции.

Цель *ранжирования по разнообразию* $R_{div} \in \mathbb{Z}$ состоит в выборе для мутации и скрещивания особей с наименьшей дискриминацией по схожести и наибольшим значением метрики различности. В таком случае популяция будет наполняться наиболее различными особями. Алгоритм оценки ранга разнообразия R_{div} для особей популяции Λ представлен на листинге 2.

```

1.  $\rho := 1$  – ранг;  $\tau := 1$  – индекс по схожести (7);  $\Lambda^* := \Lambda$  – временный список;
2. while  $\Lambda^* \neq \emptyset$  do
3.    $\Lambda^\tau := \emptyset$ ;
4.   for  $\zeta \in (1 : |\Lambda^*|)$  do
5.     if  $\nexists Q^\theta \in \Lambda^* : M_{sim}(Q^\zeta) > M_{sim}(Q^\theta)$  then  $\Lambda^\tau.add(Q^\zeta)$ ;
6.   end for;
7.    $\Lambda^{\tau*} := \Lambda^\tau$ ;
8.   while  $\Lambda^{\tau*} \neq \emptyset$  do
9.      $\mu := 1$  – индекс по различности (6);  $\Lambda^{\tau\mu} := \emptyset$ ;
10.    for  $\zeta \in (1 : |\Lambda^{\tau*}|)$  do
11.      if  $\nexists Q^\theta \in \Lambda^{\tau*} : M_{div}(Q^\zeta) < M_{div}(Q^\theta)$  then
12.         $R_{div}(Q^\zeta) := \rho$ ;
13.         $\Lambda^{\tau\mu}.add(Q^\zeta)$ ;
14.      end if;
15.    end for;
16.     $\Lambda^{\tau*} := \Lambda^{\tau*} \setminus \Lambda^{\tau\mu}$ ;
17.     $\rho := \rho + 1$ ;  $\mu := \mu + 1$ ;
```

18. **end while;**
19. $\Lambda^* := \Lambda^* \setminus \Lambda^\tau;$
20. $\tau := \tau + 1;$
21. **end while;**

Листинг 2. Псевдокод алгоритма ранжирования по разнообразию

Таким образом, при выборе особей для скрещивания и мутации по рангам R_{div} популяция будет наполняться более разнообразными потомками, в генотипическом и фенотипическом смыслах.

Ранжирование по элитарности с дискриминацией по схожести $R_{dis} \in \mathbb{R}$ использован в операторе селекции для улучшения разнообразия в формируемой популяции. Чем меньше ранг по метрике элитарности M_{eli} и чем меньше в популяции схожих особей по метрике M_{sim} (7), тем выше шанс особи быть выбранной в новую популяцию. Дискриминированный ранг элитарности для особи $Q^\zeta \in \Lambda$ задан как:

$$R_{dis}(Q^\zeta) = \frac{M_{eli}(Q^\zeta)}{M_{sim}(Q^\zeta)^2}. \quad (8)$$

Таким образом, дискриминированный ранг будет равен рангу по элитарности, если в популяции отсутствуют схожие по генотипу особи, и будет больше в противном случае.

3.2. Скрещивание – алгоритм сегментального обобщения.

Оператор скрещивания обеспечивает итеративное улучшение решения, создавая новые особи комбинированием особей-предков. Скрещивание применяется для особей с наилучшими показателями разнообразия R_{div} . Для обработки сегментов стен требуется алгоритм скрещивания не на уровне отдельных генов, а на уровне генных цепочек. Такой алгоритм назовем алгоритмом сегментального обобщения.

Алгоритм сегментального обобщения состоит в построении планов ФС новой особи при выполнении нескольких правил.

Во-первых, разница плана стены дочерней особи от планов стен родительских особей по расстоянию Хэмминга должна составлять не более отношения среднего минимальных протяжённости несущих и ненесущих сегментов по ограничению (2) к шагу разбиения стен.

Во-вторых, разница между протяженностью несущих сегментов дочерней особи и средним протяженностей несущих сегментов родительских особей должна быть наименьшей из допустимых и проходящих первое правило планов.

В-третьих, оценка плана стены должна быть наименьшей из допустимых и проходящих первое и второе правила. Не должны

нарушаться ограничения математической модели на минимально допустимые протяженности несущих и ненесущих сегментов (2).

Алгоритм сегментального обобщения достигает выполнения указанных правил благодаря операциям балансировки и обработки нарушений, а покрытия всевозможных допустимых вариантов – благодаря механизму критических точек (КТ). Алгоритм проходится вдоль стены, расставляя ФС участкам в соответствии с выполнением ограничений или балансом. Если ограничения не выполняются, то следующий участок будет установлен с ФС аналогичным нынешнему. Иначе, ФС устанавливается по той родительской стене, баланс для которой нарушен (рисунок 3). Далее алгоритм описывается подробнее.

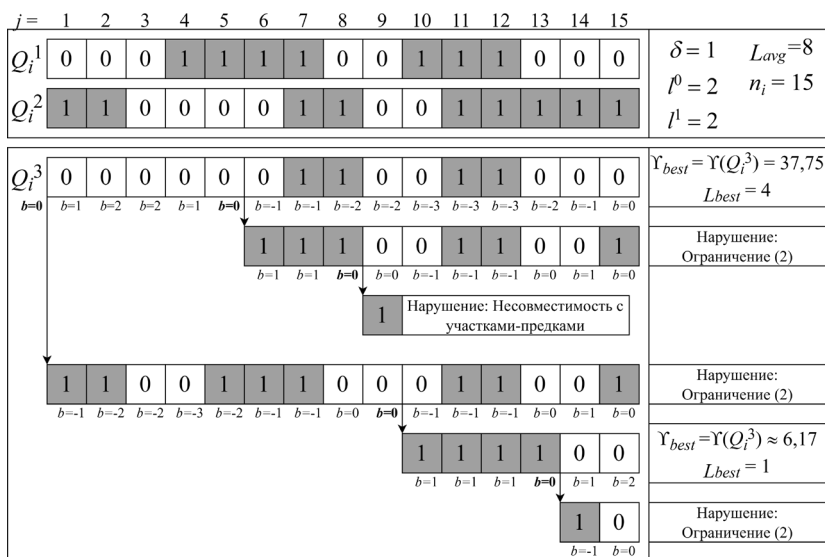


Рис. 3. Пример работы алгоритма сегментального обобщения

Даны два плана опор Q^1 и $Q^2 \in \Lambda$. Рассмотрим для некоторой стены $i \in \{1, \dots, N\}$ их планы сегментов $Q_i^1 = \{Q_{ik}^1: \forall k = 1, \dots, m_i^1\}$ и $Q_i^2 = \{Q_{ik}^2: \forall k = 1, \dots, m_i^2\}$; $Q_i^1 \in Q^1$, $Q_i^2 \in Q^2$. Определим по ним планы ФС в i -ой стене по формуле (1) как $X_i^1 = \{X_{ij}^1: X_{ij}^1 = X_{ij}(Q^1), \forall j = 1, \dots, n_i\}$ и $X_i^2 = \{X_{ij}^2: X_{ij}^2 = X_{ij}(Q^2), \forall j = 1, \dots, n_i\}$. Сформируем по ним новый план $X_i^3 = \{X_{ij}^3: X_{ij}^3 \in \{0, 1\}, \forall j = 1, \dots, n_i\}$. Введем параметр баланса плана-потомка между планами-предками $b \in \mathbb{Z}$, текущий тип ФС $v \in \{0, 1\}$ и стек КТ $J \subseteq \{j/ \forall j = 1, \dots, n_i\}$, хранящий

индексы участков с равновесным состоянием $b = 0$. Описание алгоритма сегментального обобщения представлено на листинге 3.

```

1.  $j := 1$  – индекс участка;  $b := 0$  – баланс;  $v := 0$  – тип ФС;  $J := \{1\}$  – стек КТ;
2. while  $J \neq \emptyset$  do
3.   if  $j = n_i$  then
4.      $j := J.pop()$ ;  $v := 1$ ;
5.   end if;
6.   while  $j \leq n_i$  do
7.      $j^* := \min(\{j^{**}: X_{ij^{**}}^3 = v, j^{**} = 1, \dots, j\})$ ;
8.     if  $j \neq 1$  и  $\sum_{j=j^*, \dots, j}(l_{ij}) \geq l^v$  (ограничение (2)) then
9.       if  $b = 0$  then
10.         $v := 0$ ;  $J.push(\{j\})$  – новая КТ;
11.      end if;
12.    else
13.      if  $b < 0$  then  $v := X_{ij}^1$ ;
14.      else if  $b > 0$  then  $v := X_{ij}^2$ ;
15.    end else;
16.     $X_{ij}^3 := v$ ;
17.    if  $X_{ij}^3 \neq X_{ij}^1$  и  $X_{ij}^3 \neq X_{ij}^2$  (нарушение) then
18.       $j := J.pop()$ ;  $v := 1$ ;
19.    else
20.      if  $X_{ij}^3 \neq X_{ij}^1$  then  $b := b - 1$ ;
21.      else if  $X_{ij}^3 \neq X_{ij}^2$  then  $b := b + 1$ ;
22.       $j := j + 1$ ;
23.    end else;
24.  end while;
25.   $j^* := \min(\{j^{**}: X_{ij^{**}}^3 = v, j^{**} = 1, \dots, j\})$ ;
26.  if  $\sum_{j=j^*, \dots, j}(l_{ij}) > l^v$  (ограничение (2)) then
27.    Оценка построенного плана;
28.  end if;
29. end while;

```

Листинг 3. Псевдокод алгоритма сегментального обобщения

Суть оценки плана (шаг 27) состоит в расчёте суммы отношений квадрата протяженности прилегающих несущих сегментов к каждому несущему сегменту; при этом ситуации, когда несущие сегменты расположены по краям стены, оцениваются хуже. Наилучший план сегментов должен иметь наименьшую оценку.

Для сравнения различных планов ФС построим полученный план сегментов в i -ой стене. Для этого определим количество сегментов в новом плане как $m_i^3 = 1 + |\{j: X_{ij}^3 \neq X_{ij+1}^3, \forall j = 1, \dots, n_i - 1\}|$. Определим количество участков в первом сегменте нового плана как

$\eta_{i1}^3 = \min(\{j: X_{ij}^3 \neq X_{ij+1}^3, j = 1, \dots, n_i - 1\} \cup \{n_i\})$. Тогда количество участков сегментов $\forall k = 2, \dots, m_i^3$ определяется следующим образом:

$$\eta_{ik}^3 = \min(\{j - \eta(k): X_{ij}^3 \neq X_{ij+1}^3, j = \overline{\eta(k)}, \dots, n_i\} \cup \{n_i - \eta(k)\}),$$

где $\eta(k) = \sum_{k*=1, \dots, k-1} (\eta_{ik*}^3)$ – количество участков до k -го сегмента. План сегментов по новому плану ФС задан как $Q_i^3 = \{Q_{ik}^3: k = 1, \dots, m_i^3\}$, где k -ый сегмент задается как $Q_{ik}^3 = \{j: j = \eta(k) + 1, \dots, \eta(k) + \eta_{ik}^3\}$.

Определим число несущих участков по плану Q как $L(Q_i)$:

$$L(Q_i) = \sum_{j=1}^{n_i} X_{ij}(Q), Q_i \in Q.$$

Определим лучший план как Q_i^{best} , среднее количество несущих участков родительских планов как $L_{avg} \in \mathbb{R}$, разницу длин несущих участков между лучшим планом и L_{avg} как $L_{best} \in \mathbb{R}$, оценку наилучшего плана как $\Upsilon_{best} \in \mathbb{R}$. Изначально $Q_i^{best} = \emptyset$, $L_{best} = \infty$, $\Upsilon_{best} = \infty$. Описание алгоритма оценки плана представлено на листинге 4.

1. **Begin**
2. $Q_i^{best} = \emptyset$; $L_{best} = \infty$; $\Upsilon_{best} = \infty$; $\delta \in \mathbb{Z}$ – шаг разбиения стен на участки;
3. $L_{avg} = (L(Q_i^1) + L(Q_i^2)) / 2$;
4. **if** $|b| > |l| + l^0 / (2 \cdot \delta)$ **then End** (по первому правилу);
5. **if** $|L(Q_i^3) - L_{avg}| > L_{best}$ **then End** (по второму правилу);
6. **if** $m_i^3 = 1$ **then** (стена из одного типа ФС)
7. **if** $X_{ij}^3 = 1$ **then** (полностью несущая стена)
8. **if** $L_{avg} \leq 3 \cdot n_i / 5$ **then** $\Upsilon(Q_i^3) = \infty$;
9. **else** $\Upsilon(Q_i^3) = n_i / (L_{avg} - 3 \cdot n_i / 5)^3$;
10. **else** (полностью ненесущая стена)
11. **if** $L_{avg} \geq 2 \cdot n_i / 5$ **then** $\Upsilon(Q_i^3) = \infty$;
12. **else** $\Upsilon(Q_i^3) = n_i / (2 \cdot n_i / 5 - L_{avg})^3$;
13. **end else**;
14. **else** (стена из разных типов ФС)
15. $\Upsilon_k \in \mathbb{R}$ – слагаемое оценки; $\gamma_k \in \{1, 2, 3\}$ – коэффициент по стыку;
16. $k_s \in \{1, \dots, m_i^3\}$ – индекс несущего; $k_{ns} \in \{1, \dots, m_i^3\}$ – индекс ненесущего;
17. **for** $k = \{1, \dots, m_i^3 - 1\}$ **do**
18. **if** $\chi_{ik}^3 = 0$ **then** $k_{ns} = k$; $k_s = k + 1$;
19. **else** $k_{ns} = k + 1$; $k_s = k$;
20. **if** $k_{ns} = 1$ или $k_{ns} = m_i^3$ **then** $\gamma_k = 3$;
21. **else if** $k_s \neq 1$ и $k_s \neq m_i^3$ **then** $\gamma_k = 2$;
22. **else** $\gamma_k = 1$;

23. Определим слагаемое по стыку сегментов:

$$\Upsilon_k = \gamma_k \cdot \frac{(\text{length}(Q_{ik_{ns}}^3))^2}{2 \cdot \text{length}(Q_{ik_s}^3)};$$

24. **end for**;

25. Оценка полученного плана опор Q_i^3 :

$$\Upsilon(Q_i^3) = (\sum_{k=1}^{m_i^3-1} \Upsilon_k) \sum_{j=1}^{n_i} x_{ij} / n_i;$$

26. **end else**;

27. **if** $\Upsilon(Q_i^3) < \Upsilon_{best}$ **then**

28. $\Upsilon_{best} = \Upsilon(Q_i^3)$;

29. $L_{best} = |L(Q_i^3) - L_{avg}|$;

30. $Q_i^{best} = Q_i^3$;

31. **end if**;

32. **End**;

Листинг 4. Псевдокод алгоритма оценки построенного плана

Таким образом, результатом работы алгоритма сегментального обобщения является наилучший план распределения Q_i^{best} .

3.3. Модифицированный алгоритм оператора мутации.

Операция мутации применяется для избегания чрезмерной сходимости популяции к однотипным решениям и позволяет увеличивать разнообразие популяции точечными изменениями в генотипе особи. Однако при разработке алгоритма мутации необходимо следить за нарушениями допустимости решения, т.е. ограничений задачи.

В связи с сегментальным представлением генотипа в задаче рационального размещения несущих стен было принято решение разработать алгоритм изменения генов в цельных сегментах. Суть алгоритма состоит в попытке уменьшить протяженность наибольших сегментов в различных стенах некоторого плана. Для этого обозначим множество наибольших сегментов по каждой стене как Sg :

$$Sg = \{(i, k) : \text{length}(Q_{ik}) = \max(\text{length}(Q_{ik*})) \\ \forall k, k* = \{1, \dots, m_i\}, \forall i = \{1, \dots, N\}\}.$$

Обозначим отдельный сегмент за $sg = (i, k)$, компоненты кортежа обозначим за $sg.i$ – индекс стены, $sg.k$ – индекс сегмента. Упорядочим сегменты из Sg по их длине. Пусть мутации подвергнуты первые

несколько из них $Sg' \subseteq Sg$. Рассмотрим один из этих сегментов $sg' \in Sg'$, $X' = \chi_{sg'.i \ sg'.k}$ – ФС сегмента. Длина уменьшения сегмента равна минимальной протяженности по противоположному ФС $L_{\downarrow} = l^{1-X'}$. Если протяженность сегмента после уменьшения меньше допустимой $length(sg') - L_{\downarrow} < l^X$, то алгоритм обрабатывает следующий сегмент.

Если сегмент – единственный в стене $m_{sg'.i} = 1$, то создается новый сегмент по случайному направлению минимальной длины $sg'_{new} = (i, 1 \text{ или } 2)$, $length(sg'_{new}) = L_{\downarrow}$, ФС нового сегмента $X'_{new} = 1 - X'$. Если сегмент не единственный, то из направлений $-sg' = (sg'.i, \max(sg'.k - 1, 1))$ (предыдущего) и $+sg' = (sg'.i, sg'.k + 1)$ (последующего) выбирается направление с меньшим сегментом и увеличивается на минимальную протяженность $\min(length(-sg'), length(+sg')) + L_{\downarrow}$. Протяженность исходного сегмента в обоих случаях уменьшается на соответствующую длину $length(sg') - L_{\downarrow}$. Таким образом, удастся изменить генотип особи, не нарушая установленных ограничений.

3.4. Схема оператора селекции по дискриминированному рангу. Алгоритм оператора селекции основан на дискриминированном ранге по элитарности R_{dis} (8), описанном ранее. Селекция направлена на отбор особей с наилучшими рангами элитарности и наибольшими показателями разнообразия. Пусть $\Lambda = \{Q^1, \dots, Q^{\Omega}\}$, – исходная популяция, тогда Ω – исходная мощность популяции, $\Lambda_{new} = \emptyset$ – новая популяция. Алгоритм селекции разделен на три этапа (рисунок 4).



Рис. 4. Схема разработанного алгоритма оператора селекции

На предварительном этапе особи, полученные при мутации Λ_{mut} и скрещивании Λ_{scr} , объединяются с исходной популяцией:

$$\Lambda^+ = \Lambda \cup \Lambda_{\text{мут}} \cup \Lambda_{\text{скр}}, \quad \Omega^+ = \left| \Lambda^+ \right|,$$

где Ω^+ – мощность итоговой популяции. Полученная популяция Λ^+ упорядочивается по R_{dis} (8) и усекается рангом Ω -ой особи:

$$\Lambda_{dis}^+ = \{Q^\zeta \in \Lambda^+ \mid R_{dis}(Q^\zeta) \leq R_{dis}(Q^{\zeta+1}), \forall \zeta = \{1, \dots, \Omega^+\}\};$$

$$R^\Omega = R_{dis}(Q^\zeta), Q^\zeta \in \Lambda_{dis}^+, k = \Omega;$$

$$\Lambda_{dis}^* = \{Q^\zeta \in \Lambda_{dis}^+ : R_{dis}(Q^\zeta) \leq R^\Omega, \forall \zeta = \{1, \dots, \Omega^+\}\}, \quad \Omega^* = \left| \Lambda_{dis}^* \right|, \quad (9)$$

где Ω^* – мощность усеченной популяции Λ_{dis}^* . Таким образом, к селекции остаются лишь особи с лучшими рангами R_{dis} . Если в усеченной популяции (9) остались особи с разными рангами, то проводится первый этап селекции: в новую популяцию добавляются особи с рангом меньшим предпоследнего ранга $R^{\downarrow\Omega}$ и выделяется часть популяции, ранг которых равен предпоследнему рангу:

$$R^{\Omega\downarrow} = \max(\{Q^\zeta \in \Lambda_{dis}^* \mid R_{dis}(Q^\zeta) < R^\Omega, \forall \zeta = \{1, \dots, \Omega^*\}\});$$

$$\Lambda_{new} = \Lambda_{new} \cup \{Q^\zeta \in \Lambda_{dis}^* \mid R_{dis}(Q^\zeta) < R^{\Omega\downarrow}, \forall \zeta = \{1, \dots, \Omega^*\}\};$$

$$\Lambda^{\Omega\downarrow} = \{Q^\zeta \in \Lambda_{dis}^* \mid R_{dis}(Q^\zeta) = R^{\Omega\downarrow}, \forall \zeta = \{1, \dots, \Omega^*\}\}, \quad \Omega^\downarrow = \left| \Lambda^{\Omega\downarrow} \right|, \quad (10)$$

где Ω^\downarrow – количество особей с рангом, равным $R^{\downarrow\Omega}$. Тогда особи с предпоследним рангом (10) упорядочиваются по метрике различности M_{div} (6), лучшая половина из них добавляется в новую популяцию и удаляется из усеченной популяции:

$$\Lambda_{div}^{\Omega\downarrow} = \{Q^\zeta \in \Lambda^{\Omega\downarrow} \mid M_{div}(Q^\zeta) \leq M_{div}(Q^{\zeta+1}), \forall \zeta = \{1, \dots, \Omega^\downarrow\}\}.$$

$$\Lambda_{new} = \Lambda_{new} \cup \{Q^\zeta \in \Lambda_{div}^{\Omega\downarrow} \mid \forall \zeta = \{1, \dots, \Omega^\downarrow/2\}\};$$

$$\Lambda_{dis}^* = \Lambda_{dis}^* \setminus \{Q^\zeta \in \Lambda_{div}^{\Omega\downarrow} \mid \forall \zeta = \{1, \dots, \Omega^\downarrow/2\}\}.$$

После первого этапа в усеченной популяции Λ_{dis}^* остаются только особи с последним рангом и половина особей с предпоследним рангом элитарности с большими значениями метрики различности. На втором этапе оставшиеся в усеченной популяции особи ранжируются по метрике различности M_{div} (6) и дозаполняют новую популяцию. На обоих этапах оценка метрики различности ищется в том числе и на множестве уже добавленных в новую популяцию особей.

$$\Lambda_{div}^* = \{Q^\zeta \in \Lambda_{dis}^* \mid M_{div}(Q^\zeta) \leq M_{div}(Q^{\zeta+1}), \forall \zeta = \{1, \dots, \Omega^*\}\},$$

$$\Lambda_{new} = \Lambda_{new} \cup \{Q^\zeta \in \Lambda_{div}^* \mid \forall \zeta = \{1, \dots, \Omega - |\Lambda_{new}|\}\}.$$

Представленные метрики и алгоритмы генетических операторов позволяют полноценно применить многокритериальные генетические алгоритмы для задачи рационального размещения стен. Описанные метрики различности, схожести и элитарности и основанные на них ранг разнообразия и дискриминированный ранг элитарности позволяют оценивать особи популяции как с точки зрения доминируемости, так и с точки зрения разнообразия генотипа и фенотипа.

Разработанные алгоритмы скрещивания и мутации позволяют создать новые планы без нарушения ограничений задачи. Предложенный алгоритм селекции использует описанные метрики, поддерживая как Парето-сходимость, так и разнообразие популяции.

4. Численный эксперимент: сравнение эффективности работы многокритериального генетического алгоритма. Для проведения численных экспериментов было разработано программное средство, реализующее описанные математическую модель и алгоритмы. Программа выполнена в виде надстройки над САПР Revit Autodesk версии 2021 года. Численные эксперименты были проведены на машине со следующими характеристиками: 4-х ядерный Intel Core i7-7700 3.60GHz; RAM 32Gb. Проект для эксперимента предоставлен компанией ООО «Конструктив», г. Уфа.

Проект содержит 31 стену, из которых изменяемых стен $N = 21$. Общая протяженность стен – 98.670 м, протяженность изменяемых стен – 61.480 м. Шаг разбиения стен на участки $\delta = 125$ мм, суммарное количество участков стен – 485. Ограничения протяженности l^0 и $l^1 = 500$ мм. Эксперимент направлен на сравнение разработанного алгоритма с другими МГА при решении задачи рационального

размещения стен. Для сравнения были выбраны известные модели МГА [17], их описание содержится в таблице 1. Настройки МГА: мощность популяции – 350, количество итераций – 375, доля скрещивания – 20% от мощности популяции; доля обобщаемых сегментов – 20% стен, доля мутации – 15% от популяции, шанс мутации для случайной стены – 25%.

Таблица 1. Многокритериальные генетические алгоритмы для сравнения

	Наименование	Описание
[15]	Optimum Order Multiobjective Genetic Algorithm	выстраивается новая многокритериальная задача по рангу недоминируемости и метрике схожести; строится общая метрика, аналогичная рангам Парето-доминирования
[16]		модификация; вводится дискриминация за однообразие особей; применяется в численном эксперименте
[18]	Non-dominated Sorting Genetic Algorithm	упорядочивание по рангам недоминируемости; для равномерности покрытия фронта Парето применяется метрика разреженности на основе метрики Манхэттена
[19]	Non-dominated Sorting Genetic Algorithm-II	для улучшения качества Парето-аппроксимации используется инструмент сохранения лучших индивидов в отдельном архиве
[20]	Strength Pareto Evolutionary Algorithm 2	применяется селекция по Парето-доминированию, нишевание, архивирование; ранжирование по слабости – суммарной Парето-силе доминирующих особей, где Парето-сила – количество доминируемых особей
[21]	Dynamical Multiobjective Evolutionary Algorithm	основан на принципе минимальной свободной энергии термодинамики; применяется агрегирующая функция, сочетающая ранжирование с энтропией и плотностью
[22]	Knee Point Driven Evolutionary Algorithm	для увеличения разнообразия в популяции использует точки перегиба – решения, для которых улучшение одной цели приведет к сильному ухудшению другой

Эффективность работы алгоритмов оценивается по времени выполнения алгоритма, по количеству найденных недоминируемых решений в популяции алгоритма и на общем множестве среди всех алгоритмов, а также по оценке разнообразия популяций, полученных алгоритмами, в индивидуальной популяции и на общем множестве. Для оценки разнообразия использовался алгоритм, оценивающий разнообразие популяций по количеству получаемых по особям кластеров относительно общего количества особей. Для построения кластеров использован алгоритм DBSCAN [23].

Для сравнения решений алгоритмов было проведено несколько испытаний с разным ограничением минимальной общей протяженности несущих стен, т.е. ограничение снизу для f_L . Испытания разделены на две группы: 1-я – с ограничением в 30%;

и 2-я – с ограничениями выше. В первой группе было произведено три испытания. Во второй группе проведено по одному испытанию для каждого ограничения (35%, 45%, ..., 85%). Результаты оценки времени выполнения представлены в таблице 2, оценки количества получаемых недоминируемых решений – в таблице 3 и таблице 4, оценки разнообразия полученных популяций особей – в таблице 5 и таблице 6 ((РА) – разработанный алгоритм).

Таблица 2. Время выполнения алгоритмов (минуты : секунды)

1 гр.	(РА)	OOMOGA	NSGA	NSGA-II	SPEA2	DMOEА	KNEA
< 30%	27:52	18:29	13:39	06:55	21:42	33:01	05:58
	25:57	24:12	15:32	14:29	32:35	50:17	05:41
	26:18	19:48	14:02	07:54	23:31	34:36	06:49
2 гр.	(РА)	OOMOGA	NSGA	NSGA-II	SPEA2	DMOEА	KNEA
35%	30:11	18:28	21:33	11:00	22:58	36:36	07:52
45%	28:47	19:56	30:40	15:39	26:07	49:00	09:36
55%	30:35	23:08	28:19	19:52	32:07	67:25	09:37
65%	32:42	22:20	14:46	10:51	22:23	40:47	06:54
75%	25:27	21:39	14:05	11:19	29:26	44:28	07:52
85%	24:23	19:48	10:00	09:25	16:12	18:03	09:11

По времени решения разработанный алгоритм работает быстрее DMOEA, сопоставим со SPEA2, и медленнее других. Следует отметить, что время работы разработанного алгоритма практически не уменьшается при повышении ограничения на протяженность, в отличие от других МГА. Это в том числе является следствием большого количества упорядочиваний в генетических операторах. Поэтому одним из способов увеличения скорости работы алгоритма может быть применение более быстрых алгоритмов упорядочивания.

Таблица 3. Количество недоминируемых решений в своей популяции (шт.)

1 гр.	(РА)	OOMOGA	NSGA	NSGA-II	SPEA2	DMOEА	KNEA
< 30%	174	95	49	38	54	97	23
	143	90	49	174	55	195	13
	173	100	42	70	48	92	25
2 гр.	(РА)	OOMOGA	NSGA	NSGA-II	SPEA2	DMOEА	KNEA
35%	147	90	43	143	44	148	15
45%	139	73	54	94	38	128	14
55%	66	52	43	33	54	90	8
65%	33	46	15	13	29	27	10
75%	48	21	18	35	18	18	11
85%	19	15	13	28	16	18	3

Таблица 4. Количество недоминируемых решений в общей популяции (шт.)

1 гр.	(PA)	OOMOGA	NSGA	NSGA-II	SPEA2	DMOEA	KNEA
< 30%	96	1	20	22	28	44	0
	68	5	9	12	47	26	1
	75	1	4	39	34	66	1
2 гр.	(PA)	OOMOGA	NSGA	NSGA-II	SPEA2	DMOEA	KNEA
35%	129	2	7	32	40	6	1
45%	126	1	30	2	0	31	0
55%	65	1	0	0	7	11	0
65%	22	3	5	0	0	7	0
75%	48	2	0	0	0	0	0
85%	19	0	0	0	0	7	0

Таблица 5. Разнообразие решений в своей популяции (% участия в кластерах)

1 гр.	(PA)	OOMOGA	NSGA	NSGA-II	SPEA2	DMOEA	KNEA
< 30%	6,6	6,79	3,21	2,64	4,53	3,96	2,64
	3,76	3,89	2,09	5,7	3,89	5,29	1,67
	6,18	8,18	2,91	3,82	2,91	4,18	3,64
2 гр.	(PA)	OOMOGA	NSGA	NSGA-II	SPEA2	DMOEA	KNEA
35%	3,65	4,76	3,65	6,03	2,38	4,6	2,06
45%	3,15	3,33	1,11	5,37	1,85	4,26	1,48
55%	2,31	2,02	5,2	4,05	3,18	2,89	1,73
65%	4,05	5,78	4,05	2,31	3,47	3,47	2,89
75%	5,92	5,33	4,14	5,33	2,96	2,96	4,14
85%	4,46	6,25	4,46	7,14	4,46	7,14	2,68

Таблица 6. Разнообразие решений в общей популяции (% участия в кластерах)

1 гр.	(PA)	OOMOGA	NSGA	NSGA-II	SPEA2	DMOEA	KNEA
< 30%	6,16	0,47	4,27	2,84	7,11	4,74	0
	5,36	1,79	3,57	5,95	13,1	5,95	0,6
	6,36	0,45	1,36	3,18	5,45	3,64	0,45
2 гр.	(PA)	OOMOGA	NSGA	NSGA-II	SPEA2	DMOEA	KNEA
35%	7,37	0,92	1,38	5,07	5,99	0,92	0,46
45%	8,95	0,53	2,63	0,53	0	6,84	0
55%	7,14	1,19	0	0	3,57	4,76	0
65%	16,22	8,11	5,41	0	0	5,41	0
75%	20	4	0	0	0	0	0
85%	23,08	0	0	0	0	15,38	0

По количеству недоминируемых решений на индивидуальной популяции разработанный алгоритм в среднем находит в 1.28 раз больше по сравнению с DMOEA, в 1.58 раз – с OOMOGA, в 1.89 раз – с NSGA-II, и больше – с остальными. На общем множестве особей разработанный алгоритм находит в среднем в 3 раза больше

недоминируемых решений в сравнении с DMOEA, в 4 раза больше в сравнении с NSGA-II, и больше в сравнении с другими МГА.

По разнообразию найденных решений на индивидуальной популяции разработанный алгоритм справляется в среднем в 1.11 раз хуже OOMOGA, наравне с NSGA-II и DMOEA, и в 1.5 раза лучше других алгоритмов. Однако каждый из указанных выше алгоритмов получает разнообразие популяции с меньшей мощностью Парето-оптимального множества решений, что оставляет потенциал для дальнейшего улучшения описанного в статье генетического алгоритма в сторону увеличения разнообразия получаемой популяции.

Дополнительно, представлены графики распределения особей разных алгоритмов в пространстве критериев на рисунке 5. Каждая ось отображает оценку по критерию (3), (4) или (5).

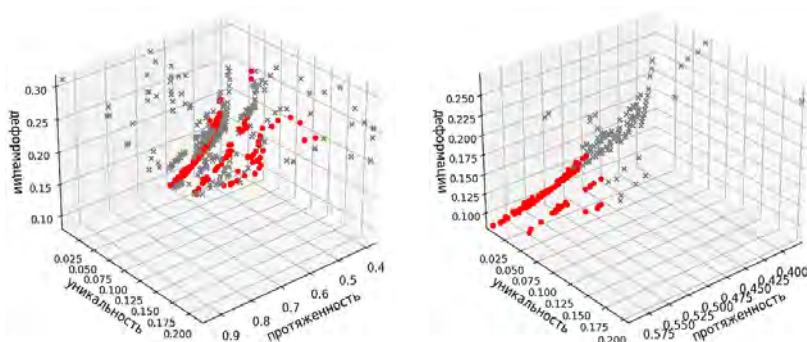


Рис. 5. Распределение особей; слева – по популяции; справа – по общему множеству; круги – особи разработанного МГА, кресты – особи других МГА

По графикам видно, что разработанный алгоритм лучше находит планы с большим количеством несущих стен и меньшими деформациями, жертвуя разнообразием популяции решений в пользу их недоминируемости.

Таким образом, по проведенным экспериментам была подтверждена эффективность работы разработанного алгоритма. Преимуществом описанного алгоритма является нахождение большего числа недоминируемых решений по сравнению с другими МГА, а также сопоставимое разнообразие получаемых решений. Недостатком алгоритма является большое время выполнения. Кроме того, продемонстрированы возможности разработанной модели задачи, в т.ч. при применении многокритериальных генетических алгоритмов.

5. Заключение. Представлена новая модель задачи рационального размещения несущих стен в многокритериальной комбинаторной постановке. По результатам численного эксперимента подтверждается, что предлагаемая модель обладает достаточной гибкостью и репрезентативностью проблематики поставленной задачи для нахождения разнообразных решений. Модель учитывает три критерия задачи: протяженности стен, количества различных длин стен и оценки деформаций по разработанной модели [14].

Был разработан новый многокритериальный генетический алгоритм, учитывающий особенности задачи рационального размещения стен. Был разработан алгоритм сегментального обобщения для операции скрещивания, находящий наилучший план стены, наиболее схожий с родительскими планами. Был предложен алгоритм мутации, обрабатывающий гены сегментно. Кроме того была разработана схема селекции, направленная на поддержку разнообразия в популяции как в фенотипическом, так и в генотипическом смысле.

Были проведены эксперименты анализа эффективности разработанного генетического алгоритма в сравнении с другими МГА. По количеству найденных недоминируемых решений разработанный алгоритм в сравнении с ближайшим конкурентным МГА DMOEA получил в 1.3 раз и в 3 раза больше Парето-оптимальных решений по популяции и общему множеству соответственно. Алгоритм показал равную оценку разнообразия популяции в сравнении с не специализированными МГА в своей популяции, уступив лишь модификации алгоритма OOMOGA. Одним из недостатков представленного алгоритма является высокое время работы, в среднем – в 2 раза дольше других алгоритмов, что может быть вызвано неоптимизированным подходом к упорядочиванию популяции на разных шагах алгоритма. Тем не менее, разработанные модель и генетический алгоритм были реализованы в виде программного продукта и могут быть использованы на практике.

Литература

1. Petprakob W. Beam-slab floor optimization using genetic and particle swarm optimization algorithms // A Thesis for degree of master of science in engineering and technology. Thailand: Thammasat University, Sirindhorn International Institute of Technology. 2014. 90 p. DOI: 10.14457/TU.the.2014.490.
2. Nimitawat A., Nanakorn P. Automated layout design of beam-slab floors using a genetic algorithm // Computers & Structures. 2009. vol. 87(21-22). pp. 1308–1330. DOI: 10.1016/j.compstruc.2009.06.007.
3. Sharafi P. Cost optimization of the preliminary design layout of reinforced concrete framed buildings // A Thesis for degree of Doctor of Philosophy in civil engineering. Australia: University of Wollongong, School of Civil, Mining and Environmental Engineering. 2013. 286 p.

4. Hadi M.H., Sharafi P., Teh L.H. A new formulation for the geometric layout optimisation of flat slab floor systems // *Proceeding of the Australasian Structural Engineering Conference 2012: The Past, Present and Future of Structural Engineering*. Perth, Western Australia: Engineers Australia, 2012. pp. 122–129. DOI: 10.3316/informit.026275045488938.
5. Sahab M.G., Ashour A.F., Toropov V.V. Cost optimisation of reinforced concrete flat slab buildings // *Engineering Structures*. 2005. vol. 27(3). pp. 313–322. DOI: 10.1016/j.engstruct.2004.10.002.
6. Meng X., Lee T.U., Xiong Y., Huang X., Xie Y.M. Optimizing support locations in the roof–column structural system // *Applied Science*. 2021. vol. 11(6). DOI: 10.3390/app11062775.
7. Zelickman Y., Amir O. Optimization of column layouts in buildings considering structural and architectural constraints // *Engineering Archive (engrXiv)*. 2024. 45 p. DOI: 10.31224/2723.
8. Steiner B., Mousavian E., Mehdizadeh Saradj F., Wimmer M., Musialski P. Integrated structural-architectural design for interactive planning // *Computer Graphics Forum*. 2016. vol. 36(8). pp. 80–94. DOI: 10.1111/cgf.12996.
9. Валиахметова Ю.И., Васильева Л.И., Зинов В.И. Алгоритм решения задачи определения рационального плана размещения несущих конструкций при строительстве многоэтажных зданий // *Современные проблемы и перспективы развития естествознания: Сб. научн. тр. Национальной научно-практической конференции (г. Уфа, 8–9 июня 2020 г.)*. Т. 2. Уфа: БГПУ, 2020. С. 46–54.
10. Eleftheriadis S., Duffour P., Stephenson B., Mumovic D. Automated specification of steel reinforcement to support the optimisation of RC floors // *Automation in Construction*. 2018. vol. 96. pp. 366–377. DOI: 10.1016/j.autcon.2018.10.005.
11. Wang J., Chen K., Yang H., Zhang L. Ensemble deep learning enabled multi-condition generative design of aerial building machine considering uncertainties // *Automation in Construction*. 2024. vol. 157. DOI: 10.1016/j.autcon.2023.105134.
12. Лешкевич О.Н. Использование искусственных нейронных сетей для оценки армирования железобетонных плит перекрытия // *Проблемы современного бетона и железобетона: Сб. научн. тр. № 11*. Минск: Ин-т БелНИИС, 2019. С. 51–62. DOI: 10.35579/2076-6033-2019-11-04.
13. Liao W.J., Lu X.Z., Fei Y.F., Gu Y., Huang Y.L. Generative AI design for building structures // *Automation in Construction*. 2024. vol. 157. DOI: 10.1016/j.autcon.2023.105187.
14. Зинов В.И., Картак В.М., Валиахметова Ю.И. Алгоритм оценки деформации плит перекрытий по пролетно-опорным схемам здания // *Системы анализа и обработки данных*. 2023. № 4(92). С. 35–54. DOI: 10.17212/2782-2001-2023-4-35-54.
15. Long Q., Wu C.Z., Wang X.Y., Jiang L., Li J. A multiobjective genetic algorithm based on a discrete selection procedure // *Mathematical Problems in Engineering*. 2015. vol. 2015. 17 p. DOI: 10.1155/2015/349781.
16. Валиахметова Ю.И., Прокудина Е.И., Зинов В.И. Многокритериальный генетический алгоритм решения задачи рационального размещения опорных конструкций на плане зданий // *Современные физика, математика, цифровые и нанотехнологии в науке и образовании: Сб. научн. тр. II Всероссийской молодежной школы-конференции (г. Уфа, 18–20 апреля 2023 г.)*. Уфа: БГПУ, 2023. С. 95–100.
17. Rahimi I., Gandomi A.H., Nikoo M.R., Chen F. A comparative study on evolutionary multi-objective algorithms for next release problem // *Applied Soft Computing*. 2023. vol. 144. DOI: 10.1016/j.asoc.2023.110472.

18. Srinivas N., Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms // *Evolutionary Computation*. 1994. vol. 2(3). pp. 221–248. DOI: 10.1162/evco.1994.2.3.221.
19. Deb K., Pratap A., Agarwal S., Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGAII // *IEEE Transactions on Evolutionary Computation*. 2002. vol. 6(2). pp. 182–197. DOI: 10.1109/4235.996017.
20. Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm // *TIK-Report*. 2001. vol. 103. DOI: 10.3929/ethz-a-004284029.
21. Liu M., Zou X., Chen Y., Wu Z. Performance assessment of DMOEA-DD with CEC 2009 MOEA Competition Test Instances // *Proceeding of the IEEE Congress on Evolutionary Computation (CEC 2009)*. Norway: Trondheim, 2009. pp. 2913–2918. DOI: 10.1109/CEC.2009.4983309.
22. He M., Wang X., Chen H., Li X. A Knee Point-Driven Many-Objective Evolutionary Algorithm with Adaptive Switching Mechanism // *Journal of Applied Mathematics*. 2024. vol. 2024(1). 43 p. DOI: 10.1155/2024/4737604.
23. Ester M., Kriegel H.-P., Sander J., Xu X. A density-base algorithm for discovering clusters in large spatial databases with noise // *Proceeding of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*. Germany: Munich, 1996. pp. 226–231.

Зинов Владислав Игоревич — аспирант, кафедра технической кибернетики, Уфимский университет науки и технологий. Область научных интересов: оптимизация в сфере проектирования зданий, эвристические и метаэвристические методы, системы поддержки принятия решений. Число научных публикаций — 6. zinovvladislavufa@gmail.com; улица Заки Валиди, 32, 450076, Уфа, Россия; р.т.: +7(917)435-5560.

Картак Вадим Михайлович — д-р физ.-мат. наук, профессор, заведующий кафедрой, кафедра вычислительной техники и защиты информации, Уфимский университет науки и технологий. Область научных интересов: большие данные, методы оптимизации, информационная безопасность. Число научных публикаций — 80. kvmail@mail.ru; улица Заки Валиди, 32, 450076, Уфа, Россия; р.т.: +7(917)433-3919.

Валнахметова Юлия Ильясовна — канд. техн. наук, доцент кафедры, кафедра вычислительной математики и кибернетики, Уфимский университет науки и технологий. Область научных интересов: эвристические алгоритмы решения задач дискретной оптимизации, математическое моделирование, численные методы, комплексы программ. Число научных публикаций — 60. julikas@inbox.ru; улица Заки Валиди, 32, 450076, Уфа, Россия; р.т.: +7(917)348-1947.

V. ZINOV, V. KARTAK, Y. VALIAKHMETOVA
**SOLVING MULTI-OBJECTIVE RATIONAL PLACEMENT
OF LOAD-BEARING WALLS PROBLEM VIA GENETIC
ALGORITHM**

Zinov V., Kartak V., Valiakhmetova Y. Solving Multi-Objective Rational Placement of Load-Bearing Walls Problem via Genetic Algorithm.

Abstract. The rational placement of load-bearing walls remains a complex and poorly studied problem, despite the existence of numerous algorithms and models for solving the similar problem of column placement. The main complexity factors are the large number of alternative solutions, the significant time required to calculate deformations for a given wall placement, and the multi-objective nature of the problem. In addition to the nonlinear criterion for estimating deformations, it is necessary to minimize the length of load-bearing walls and the number of their unique lengths. A model for the rational placement of load-bearing walls is proposed, which divides the walls into functional parts with a specific step and considers all the required target criteria. Adjacent wall parts with the same functionality are combined into segments. The combinatorial formulation applied in the model of the problem allows the use of genetic algorithms as a solution tool. Therefore, a new approach to multi-objective genetic algorithm is proposed, containing metrics for calculating population diversity at the phenotype and genotype levels. Modifications of crossover, mutation, and selection operators, considering the segmental structure of the wall's genotype, are presented. A comparative analysis of the developed algorithm with other known multi-objective genetic algorithms showed that the developed algorithm finds, on average, three times more non-dominated solutions, particularly more plans with a lower deformation estimates, despite the twice-longer execution time. The proposed model differs significantly from previous models in terms of handling deformations in slab-support systems, comparing placement plans with each other rather than calculating precise reinforcement estimates, which is often unnecessary at the early stages. The proposed genetic algorithm scheme increases the number of found nondominated solutions without losing their diversity, and can be used to solve other multi-objective problems, taking into account the specified features. The developed algorithm was easily integrated into the CAD-based decision support software and can be used in practice by building designers.

Keywords: optimization in building design, wall placement problem, multiobjective genetic algorithm, combinatorial optimization, comparative estimation of slab deformations.

References

1. Petprakob W. Beam-slab floor optimization using genetic and particle swarm optimization algorithms. A Thesis for degree of master of science in engineering and technology. Thailand: Thammasat University, Sirindhorn International Institute of Technology. 2014. 90 p. DOI: 10.14457/TU.the.2014.490.
2. Nimitawat A., Nanakorn P. Automated layout design of beam-slab floors using a genetic algorithm. *Computers & Structures*. 2009. vol. 87(21-22). pp. 1308–1330. DOI: 10.1016/j.compstruc.2009.06.007.
3. Sharafi P. Cost optimization of the preliminary design layout of reinforced concrete framed buildings. A Thesis for degree of Doctor of Philosophy in civil engineering. Australia: University of Wollongong, School of Civil, Mining and Environmental Engineering. 2013. 286 p.
4. Hadi M.H., Sharafi P., Teh L.H. A new formulation for the geometric layout optimisation of flat slab floor systems. *Proceeding of the Australasian Structural*

- Engineering Conference 2012: The Past, Present and Future of Structural Engineering. Perth, Western Australia: Engineers Australia, 2012. pp. 122–129. DOI: 10.3316/informit.026275045488938.
5. Sahab M.G., Ashour A.F., Toropov V.V. Cost optimisation of reinforced concrete flat slab buildings. *Engineering Structures*. 2005. vol. 27(3). pp. 313–322. DOI: 10.1016/j.engstruct.2004.10.002.
6. Meng X., Lee T.U., Xiong Y., Huang X., Xie Y.M. Optimizing support locations in the roof–column structural system. *Applied Science*. 2021. vol. 11(6). DOI: 10.3390/app11062775.
7. Zelickman Y., Amir O. Optimization of column layouts in buildings considering structural and architectural constraints. *Engineering Archive (engrXiv)*. 2024. 45 p. DOI: 10.31224/2723.
8. Steiner B., Mousavian E., Mehdizadeh Saradj F., Wimmer M., Musialski P. Integrated structural-architectural design for interactive planning. *Computer Graphics Forum*. 2016. vol. 36(8). pp. 80–94. DOI: 10.1111/cgf.12996.
9. Valiakhmetova Yu.I., Vasil'eva L.I., Zinov V.I. Algoritm reshenija zadachi opredelenija racional'nogo plana razmeshhenija nesushhih konstrukcij pri stroitel'stve mnogojetazhnyh zdaniy [An algorithm to solving a rational placement of load-bearing structures problem in the design of a multi-storey building]. *Sovremennye problemy i perspektivy razvitiya estestvoznaniya: Sb. nauchn. tr. Nacional'noj nauchno-prakticheskoy konferencii [Contemporary issues and perspectives on natural science: Collected research papers of National research-to-practise conference]*. Ufa: BSPU, 2020. pp. 46–54. (In Russ.).
10. Eleftheriadis S., Duffour P., Stephenson B., Mumovic D. Automated specification of steel reinforcement to support the optimisation of RC floors. *Automation in Construction*. 2018. vol. 96. pp. 366–377. DOI: 10.1016/j.autcon.2018.10.005.
11. Wang J., Chen K., Yang H., Zhang L. Ensemble deep learning enabled multi-condition generative design of aerial building machine considering uncertainties. *Automation in Construction*. 2024. vol. 157. DOI: 10.1016/j.autcon.2023.105134.
12. Liashkevich A. Ispol'zovanie iskusstvennyh nejronnyh setej dlja ocenki armirovaniya zhelezobetonnyh plit perekrytija [Application of artificial neural networks for assessing the reinforcement of reinforced-concrete floor slabs Использование искусственных нейронных сетей для оценки армирования железобетонных плит перекрытия] *Problemy sovremennogo betona i zhelezobetona: Sb. nauchn. tr. [Contemporary Issues of Concrete and Reinforced Concrete: Collected Research Papers]*. no. 11. Minsk: Institute BelNIIS, 2019. pp. 51–62. DOI: 10.35579/2076-6033-2019-11-04. (In Russ.).
13. Liao W.J., Lu X.Z., Fei Y.F., Gu Y., Huang Y.L. Generative AI design for building structures. *Automation in Construction*. 2024. vol. 157. DOI: 10.1016/j.autcon.2023.105187.
14. Zinov V.I., Kartak V.M., Valiakhmetova Yu.I. [An algorithm for estimating the deformation of floor slabs based on the span-support topology of the building]. *Sistemy analiza i obrabotki dannyh – Analysis and data processing systems*. 2023. no. 4(92). pp. 35–54. DOI: 10.17212/2782-2001-2023-4-35-54. (In Russ.).
15. Long Q., Wu C.Z., Wang X.Y., Jiang L., Li J. A multiobjective genetic algorithm based on a discrete selection procedure. *Mathematical Problems in Engineering*. 2015. vol. 2015. 17 p. DOI: 10.1155/2015/349781.
16. Valiakhmetova Yu.I., Prokudina E.I., Zinov V.I. Mnogokriterial'nyj geneticheskij algoritm reshenija zadachi racional'nogo razmeshhenija opornyh konstrukcij na plane zdaniy [Multiobjective genetic algorithm for solution of support structure rational placement on the building plan problem] *Sovremennye fizika, matematika, cifrovye i nanotekhnologii v nauke i obrazovanii: Sb. nauchn. tr. II Vserossijskoj molodezhnoj*

- shkoly-konferencii [Modern physics, mathematics, digital and nanotechnologies in science and education: Collected research papers of II All-Russia youth school-conference]. Ufa: BSPU, 2023. pp. 95–100. (In Russ.).
17. Rahimi I., Gandomi A.H., Nikoo M.R., Chen F. A comparative study on evolutionary multi-objective algorithms for next release problem. *Applied Soft Computing*. 2023. vol. 144. DOI: 10.1016/j.asoc.2023.110472.
 18. Srinivas N., Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*. 1994. vol. 2(3). pp. 221–248. DOI: 10.1162/evco.1994.2.3.221.
 19. Deb K., Pratap A., Agarwal S., Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGAII. *IEEE Transactions on Evolutionary Computation*. 2002. vol. 6(2). pp. 182–197. DOI: 10.1109/4235.996017.
 20. Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-Report*. 2001. vol. 103. DOI: 10.3929/ethz-a-004284029.
 21. Liu M., Zou X., Chen Y., Wu Z. Performance assessment of DMOEA-DD with CEC 2009 MOEA Competition Test Instances. *Proceeding of the IEEE Congress on Evolutionary Computation (CEC 2009)*. Norway: Trondheim, 2009. pp. 2913–2918. DOI: 10.1109/CEC.2009.4983309.
 22. He M., Wang X., Chen H., Li X. A Knee Point-Driven Many-Objective Evolutionary Algorithm with Adaptive Switching Mechanism. *Journal of Applied Mathematics*. 2024. vol. 2024(1). 43 p. DOI: 10.1155/2024/4737604.
 23. Ester M., Kriegel H.-P., Sander J., Xu X. A density-base algorithm for discovering clusters in large spatial databases with noise. *Proceeding of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*. Germany: Munich, 1996. pp. 226–231.

Zinov Vladislav — Postgraduate student, Department of technical cybernetics, Ufa University of Science and Technology. Research interests: optimization in the building design industry, heuristic and metaheuristic methods, decision support systems. The number of publications — 6. zinovvladislavufa@gmail.com; 32, Z. Validi St., 450076, Ufa, Russia; office phone: +7(917)435-5560.

Kartak Vadim — Ph.D., Dr.Sci., Professor, head of the department, Department of computing and information security, Ufa University of Science and Technology. Research interests: big data, optimization methods, information security. The number of publications — 80. kvmail@mail.ru; 32, Z. Validi St., 450076, Ufa, Russia; office phone: +7(917)433-3919.

Valiakhmetova Yulia — Ph.D., Associate professor of the department, Department of computational mathematics and engineering cybernetics, Ufa University of Science and Technology. Research interests: heuristic algorithms in discrete optimization problems, mathematical modeling, numerical analysis, software systems. The number of publications — 60. julikas@inbox.ru; 32, Z. Validi St., 450076, Ufa, Russia; office phone: +7(917)348-1947.