

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ
Зав.кафедрой «Прикладная
математика и информатика»
_____ А.В.Очеповский
« ____ » _____ 2016 г.

ЗАДАНИЕ
на выполнение бакалаврской работы

Студент Сулимов Юрий Николаевич

1. Тема: «Моделирование ленточных фундаментов с использованием генетических алгоритмов»
2. Срок сдачи студентом законченной выпускной квалификационной работы:
25 мая 2016 г.
3. Исходные данные к выпускной квалификационной работе:
 - язык объектно-ориентированного программирования Java,
 - нормативные акты «Строительные нормы и правила»,
 - математический аппарат генетических алгоритмов.
4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов) :
 - Анализ состояния вопроса с описанием требований, предъявляемых современным ленточным фундаментам.
 - Разработка алгоритма моделирования ленточных фундаментов. В данном разделе вопрос моделирования ленточных фундаментов рассматривается как задача многопараметрической оптимизации.
 - Программная реализация предложенных решений. Графический интерфейс пользователя, генерирование рабочих чертежей.
 - Апробация предложенных решений на реальных данных. В этом разделе приводятся примеры работы программы.
5. Ориентировочный перечень графического и иллюстративного материала:

- блок-схемы поясняющие, работу алгоритма;
- набор формул, объясняющих математический аппарат алгоритма;
- демонстрация работы алгоритма на реальном наборе данных;
- графики и диаграммы, поясняющие результат работы алгоритма.

6. Дата выдачи задания « 11 » января 2016 г.

Заместитель директора по
строительству, Волжское
ЗАО «Гидроспецстрой»

Ю.А. Тарасов

Руководитель выпускной
квалификационной работы

В.С. Климов

Задание принял к
исполнению

Ю.Н. Сулимов

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ
Зав.кафедрой «Прикладная
математика и информатика»
_____ А.В.Очеповский

« ____ » _____ 2016 г.

**КАЛЕНДАРНЫЙ ПЛАН
выполнения бакалаврской работы**

Студента Сулимов Юрий Николаевич
по теме Моделирование ленточных фундаментов с использованием
генетических алгоритмов

Наименование раздела работы	Плановый срок выполнения раздела	Фактический срок выполнения раздела	Отметка о выполнении	Подпись руководителя
1. Анализ состояния вопроса	1.04.2016	1.04.2016	выполнено	
2. Разработка алгоритма моделирования ленточных фундаментов	11.04.2016	11.04.2016	выполнено	
3. Программная реализация предложенных решений	25.04.2016	25.04.2016	выполнено	
4. Апробация предложенных решений на реальных данных	9.05.2016	9.05.2016	выполнено	
Оформление пояснительной записки	12.05.2016	12.05.2016	выполнено	
Подготовка доклада и графического материала к защите	17.05.2016	17.05.2016	выполнено	

Предварительная защита бакалаврской работы	20.05.2016-20.06.2016	31.05.2016-	выполнено	
Проверка ВКР в системе «Антиплагиат.ВУЗ»	27.05.2016	27.05.2016	выполнено	
Сдача пояснительной записки ВКР	20.06.2016	29.06.2016	выполнено	

Руководитель выпускной
квалификационной работы

В.С. Климов

Задание принял к исполнению

Ю.Н. Сулимов

Аннотация

Тема данной выпускной квалификационной работы: Моделирование ленточных фундаментов с использованием генетических алгоритмов.

Работа посвящена исследованию проблемы автоматического генерирования технических решений при проектировании сборно-ленточных фундаментов.

Объектом исследования является автоматизированное проектирование сборно-ленточных фундаментов. Предметом исследования является генерирование оптимальной раскладки фундаментных блоков с помощью генетических алгоритмов.

Целью работы является - повышение автоматизации проектирования схем раскладки ФБС блоков за счет разработки алгоритма основанного на эволюционных вычислениях.

Во введении дается краткая характеристика исследования.

В первой главе проводится анализ состояния вопроса. Рассматриваются существующие программные решения для автоматизации процесса проектирования фундаментов.

Во второй главе проводится разработка алгоритма моделирования фундамента, основанного на эволюционных вычислениях. Приведены примеры по адаптации генетического алгоритма под решаемую задачу.

В третьей главе приводится программная реализация предложенных решений. Приводится алгоритм работы с изготовленным, в рамках бакалаврской работы, программным продуктом.

В заключении приводятся выводы по работе.

Затем приведен список используемой литературы, в котором присутствуют отечественные и зарубежные книги.

Выпускная квалификационная работа содержит пояснительную записку объемом 48 страниц, включая 35 рисунков, две таблицы и 18 формул. одно приложение и список литературы из 20 источников.

Оглавление

Введение.....	3
1 Анализ состояния вопроса	5
1.1 Обзор предметной области	5
1.2 Обзор программных аналогов	11
2 Разработка алгоритма моделирования фундамента, основанного на эволюционных вычислениях.....	19
2.1 Генетические алгоритмы как метод решения задач оптимизации ..	19
2.2 Моделирование ленточного фундамента как задача оптимизации .	25
2.3 Адаптация генетического алгоритма под решаемую задачу.....	33
3 Практическая реализация	38
3.1 Программная реализация алгоритма.....	38
3.2 Пример работы программы.....	43
Заключение	46
Список используемой литературы	47
Приложение А UML диаграммы классов	49

Введение

Классические методы оптимизации условно можно разделить на две группы: основанные на производных (например, метод градиентного спуска) и стохастические методы (например, метод Монте-Карло). Основной проблемой классических методов оптимизации являются сложность применяемого математического аппарата и как следствие – долгое время поиска приемлемого решения задачи.

Однако с развитием алгоритмов искусственного интеллекта появился новый стохастический метод решения задач оптимизации – генетические алгоритмы [1-5]. Этот метод основан на принципах естественного отбора Ч. Дарвина. С точки зрения машинного обучения данный метод относится к группе «обучение с подкреплением». В бакалаврской работе предложено применение генетического алгоритма для моделирования сборно-ленточных фундаментов, применяемых при строительстве, как многоэтажных домов, так и индивидуальных коттеджей (рисунок 1).



Рисунок 1 – Сборно-ленточный фундамент.

Сборно-ленточные фундаменты состоят из нескольких слоев железобетонных элементов, форма и размер которых фиксированы и соответствуют ГОСТ 13579 и ГОСТ 13580. Поэтому актуальной проблемой моделирования фундаментов является подбор размера и расположения этих железобетонных элементов с учетом:

- границ будущего дома, т.е. требуется избегать обрезков и остатков железобетонных элементов для экономии финансовых средств;
- соблюдения требований СНиП по правилам перевязки (расположению «в шахматном порядке») блоков (критерий прочности при решении задачи оптимизации). Т.е. требуется обеспечить прочность фундамента на изгиб.

В ходе выполнения бакалаврской работы на языке Java будет разработано программное обеспечение по поиску конфигураций фундаментов, отвечающих описанным выше критериям.

Таким образом, в бакалаврской работе будет решена актуальная проблема автоматизации моделирования сборно-ленточных фундаментов с использованием современных технологий искусственного интеллекта.

1 Анализ состояния вопроса

1.1 Обзор предметной области

Фундаментом называется железобетонная несущая конструкция, которая воспринимает нагрузки от различных вышележащих конструкций, распределяет их и передает нагрузки грунту. Другое назначение фундамента – препятствование передаче вредных изгибающих моментов возникающих в грунте (например, при морозном пучении) вышележащим конструкциям для исключения возможности появления трещин на стенах здания.

Фундаменты бывают различных типов – ленточные, столбчатые, свайные и плитные (рисунок 2).

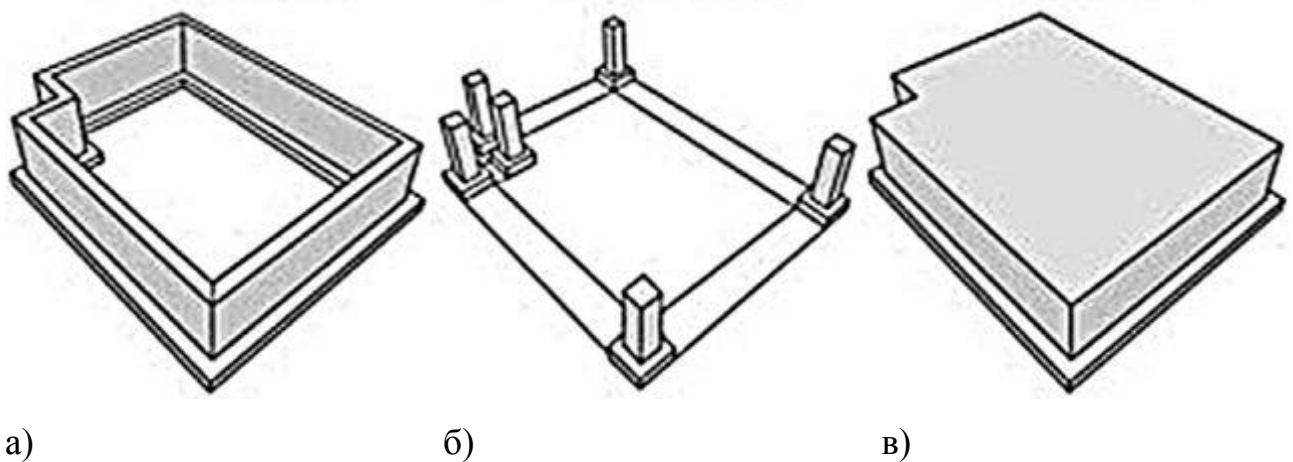


Рисунок 2 – Типы фундаментов: а – ленточный, б – свайный, в – плитный

Плитные фундаменты при правильном проектировании лучше всего препятствуют возникновению изгибающих моментов в стенах здания. Причиной этому является наибольшее (по сравнению с другими типами фундаментов) количество закладываемой в него арматуры и большая площадь опоры на грунт. Однако повсеместному использованию такого типа фундаментов в Российской Федерации мешают его недостатки:

- 1) Высокая теплопередача фундамента из-за большой площади соприкосновения с холодным грунтом. Это недостаток делает

нецелесообразным применения таких типов фундаментов в регионах с холодной зимой.

- 2) Высокий расход бетона и арматуры делают его самым дорогим в изготовлении типом фундамента.

Столбчатые фундаменты являются самыми экономичными с точки зрения расхода материалов. Однако из-за малой площади опирания на грунт у них самая низкая несущая способность. По этой причине данный тип фундамента подходит только для легких сооружений (одноэтажных деревянных домов, бань).

Для возведения свайных фундаментов требуется применение специальной техники - сваебойных машин и буровых установок. Поэтому такой тип фундамента применяется:

- 1) При слабых грунтах, когда применение других типов фундамента не возможно.
- 2) При высокой рентабельности строительства, позволяющей компенсировать затраты по использованию специализированной техники.

Самым распространенным типом фундамента, применяемым в Российской Федерации, является ленточный фундамент. Это связано с отсутствием у него перечисленных выше недостатков. Он подходит для возведения тяжелых домов, экономичен с точки зрения использования материалов.

При этом ленточные фундаменты делятся на монолитные и сборные. Одним из этапов изготовления монолитного ленточного является установка опалубки. Опалубка – это пластиковая или деревянная оболочка, повторяющая форму будущего фундамента и препятствующая растеканию бетона вплоть до его затвердевания. Процесс установки опалубки является трудозатратным мероприятием (рисунок 3). Кроме того, стоимость материалов опалубки может достигать до половины конечной стоимости материалов фундамента. При установке опалубки необходимо выполнение прочностных расчетов, чтобы

проверить, не распадется ли опалубка при заливании в нее бетона. Другим недостатком монолитных фундаментов является невозможность их заливки в дождливую погоду, так как излишки воды в бетоне значительно снижают его марку (конечную прочность).



Рисунок 3 – Установка опалубки для ленточного фундамента

Альтернативой монолитному фундаменту является сборный фундамент. Он состоит из отдельных стандартизированных железобетонных блоков, которые изготавливаются на специализированных заводах (рисунок 4). Изготовление этих блоков проходит в условиях (с поддержанием нужной температуры и влажности), позволяющих достичь им максимальную прочность. Фундамент из этих блоков возводится быстро – обычно за сутки и уже на следующие сутки можно начинать кладку этажей. Объемная стоимость данных блоков равна стоимости жидкого бетона. Если в здании предусмотрено

наличие подвалов, то вне зависимости от типа фундамента в качестве стенового материала так же применяются железобетонные блоки.

По этим причинам применение железобетонных блоков при возведении фундаментов и стен подвалов получило широкое распространение в Российской Федерации.

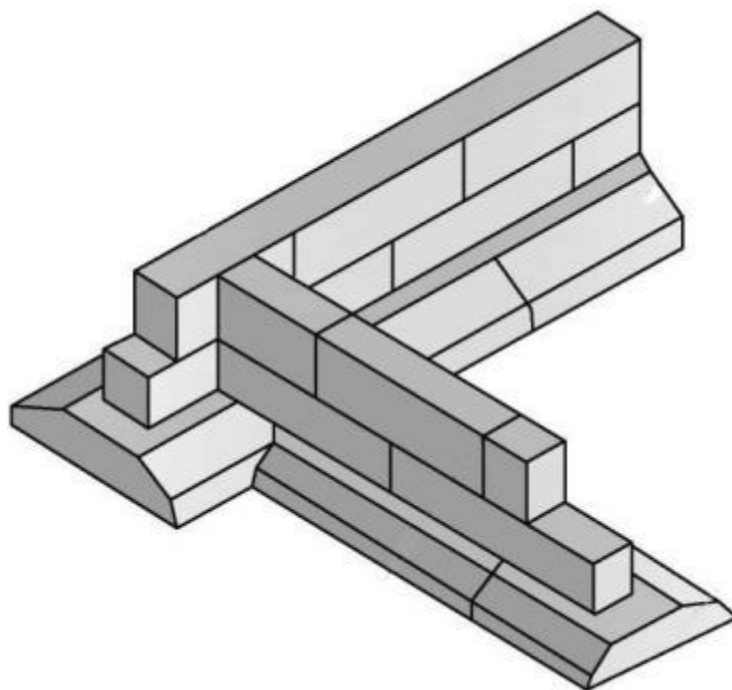


Рисунок 4 – Устройство сборно-ленточного фундамента

Железобетонные блоки, предназначенные для устройства фундаментов и стен подвалов, имеют маркировку ФБС (рисунок 5). Их форма и размер фиксированы в соответствии с ГОСТ 13579 и ГОСТ 13580 (таблица 1).

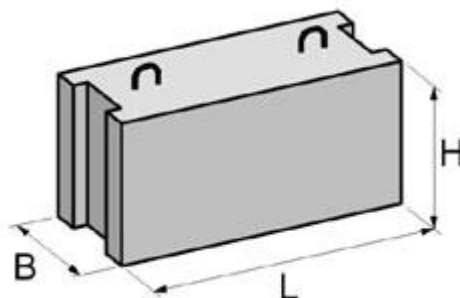


Рисунок 5 – Форма ФБС блоков

Таблица 1 – Размеры ФБС блоков в соответствии с ГОСТ 13579 и ГОСТ 13580

Обозначение	Длина L , мм	Ширина B , мм	Высота H , мм	Масса, т	Объем, м ³
ФБС 24-3-6	2380	300	580	0,97	0,41
ФБС 12-3-6	1180	300	580	0,49	0,21
ФБС 9-3-6	880	300	580	0,35	0,15
ФБС 24-4-6	2380	400	580	1,30	0,54
ФБС 12-4-6	1180	400	580	0,64	0,27
ФБС 9-4-6	880	400	580	0,47	0,20
ФБС 24-5-6	2380	500	580	1,63	0,68
ФБС 12-5-6	1180	500	580	0,79	0,34
ФБС 9-5-6	880	500	580	0,59	0,23
ФБС 24-6-6	2380	600	580	1,96	0,82
ФБС 12-6-6	1180	600	580	0,96	0,41
ФБС 9-6-6	880	600	580	0,70	0,32

Исходными данными для начала проектирования сборно-ленточного фундамента являются:

- габариты (границы) будущей постройки;
- требуемая высота фундамента/стен подвала;
- расположение внутренних несущих стен (рисунок 6).

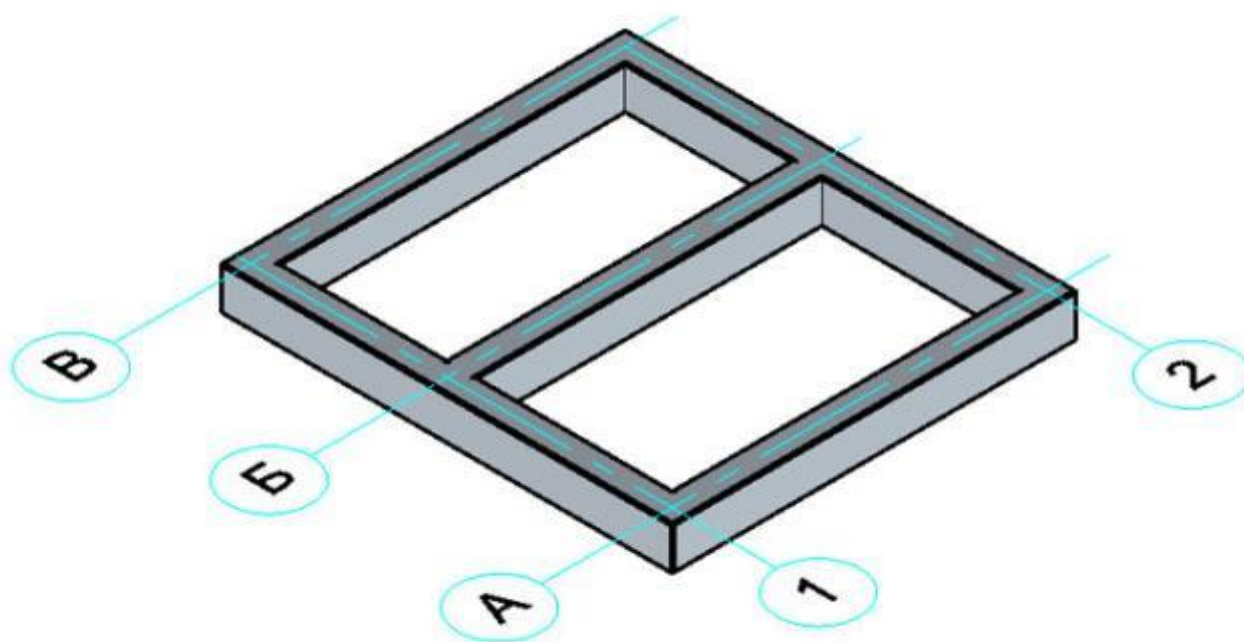


Рисунок 6 – Исходные данные для проектирования ленточного фундамента: А, Б, В, 1, 2 – расположение несущих стен здания.

По известным исходным данным производится моделирование сборно-ленточного фундамента. Целью моделирования является – определить количество и номенклатуру необходимых ФБС блоков и их взаимное расположение. Результатом такого моделирования являются рабочие чертежи по раскладки блоков (эти чертежи являются частью проектной документации по строительству домов).

Ограничения, накладываемые на расположение ФБС блоков:

- размеры ФБС блоков фиксированы в соответствии с ГОСТ 13579 и ГОСТ 13580;
- ФБС блоки на любом из уровней не должны выходить за границы (габариты) здания;
- ФБС блоки должны располагаться с перевязкой вертикальных швов, глубина которых должна быть не менее $0,4$ высоты блока при малосжимаемых грунтах и не менее высоты блока при сильносжимаемых грунтах (рисунок 7);
- допускается оставшиеся незакрытые места между блоками заливать бетоном или закладывать цокольным кирпичом. Чем больше пропусков в фундаменте (или стене подвала), которые не удалось закрыть стандартными блоками ФБС, тем выше трудоемкость и стоимость возведения такого фундамента.

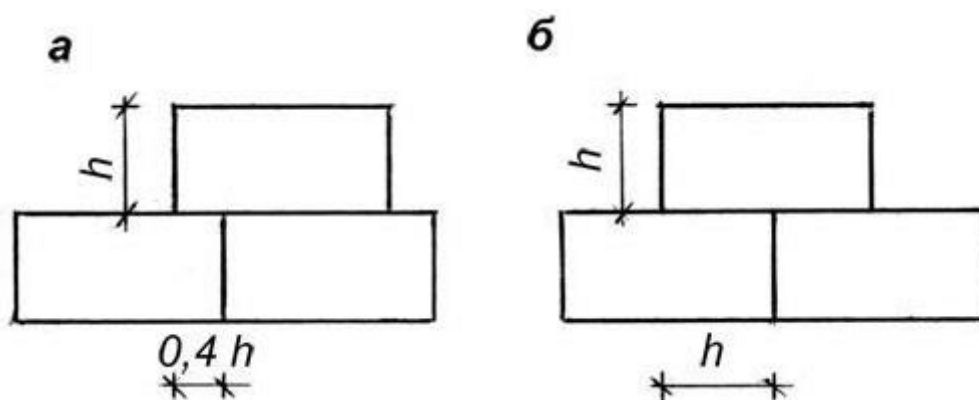


Рисунок 7 – Требования СНиП по перевязке блоков ФБС: а – при малосжимаемых грунтах, б – при сильносжимаемых грунтах.

1.2 Обзор программных аналогов

Рассмотрим, какое программное обеспечение существует для моделирования сборно-ленточных фундаментов. Сюда, в первую очередь, относятся системы автоматизированного проектирования и черчения AutoCAD (разработчик – Autodesk, США) и Компас (разработчик – Аскон, Россия). Эти системы предназначены для подготовки всей необходимой проектной документации по строительству домов. Однако моделирование раскладки ФБС блоков проектировщиком выполняется «вручную» (т.е. конечный результат моделирования напрямую зависит от опыта и знаний проектировщика) (рисунок 8). Другим недостатком данного программного обеспечения является его высокая стоимость: лицензия на 1 год AutoCad – 55 000 рублей, коробочная версия Компас-3D V16 – 110 000 руб.

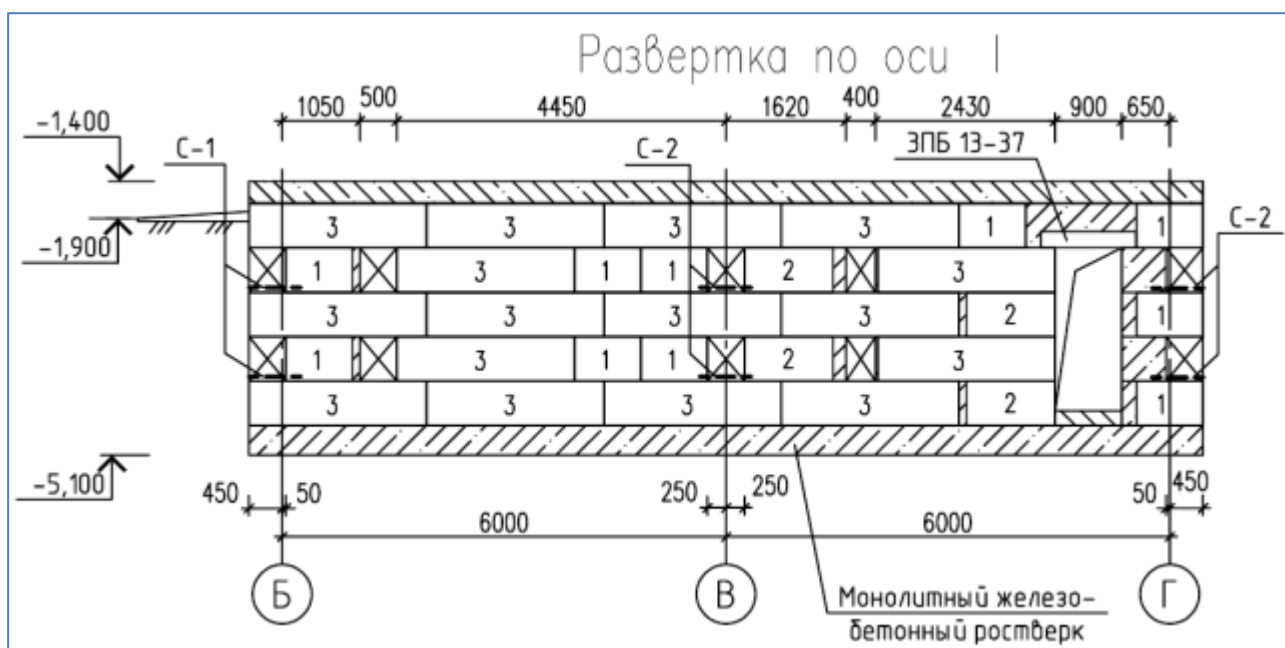


Рисунок 8 – Раскладки фундаментных блоков выполненная «вручную» с использованием САПР AutoCAD.

Другим программным инструментом моделирования является онлайн калькулятор фундаментных блоков от ООО «МосПлита» (рисунок 9). В

отличие от предыдущих аналогов он является бесплатным в использовании. Основное его назначение – расчет количества необходимых ФБС блоков выбранной номенклатуры.

Рисунок 9 – Онлайн калькулятор фундаментных блоков от ООО «МосПлита» (<http://mosplita.ru/calc-fundament>)

Математическая модель, используемая в данной реализации, для расчета количества K блоков можно выразить формулой (1):

$$K = \left\lceil \frac{gf + hf}{H} \right\rceil \cdot \sum_{i=1}^m \left\lceil \frac{lf_k}{L} \right\rceil \cdot Z, \quad (1)$$

где gf – глубина залегания фундамента, hf – высота фундамента, lf – длина несущей стены, m – количество несущих стен, Z – коэффициент запаса, в данном случае он равен 1,1 (запас – 10%), H и L – высота и длина используемых ФБС блоков. Данными прямоугольными скобками обозначается округление до целого в большую сторону.

Главным недостатком калькулятора фундаментных блоков является отсутствие функции составления раскладки блоков.

Другим программным средством моделирования сборно-ленточных фундаментов является модуль Project Studio CS ФУНДАМЕНТЫ представляющий собой дополнение к AutoCAD (рисунок 10). В отличие от предыдущих аналогов, данное программное обеспечение позволяет в

автоматическом режиме моделировать раскладку ФБС блоков и генерировать соответствующие рабочие чертежи (рисунок 11).

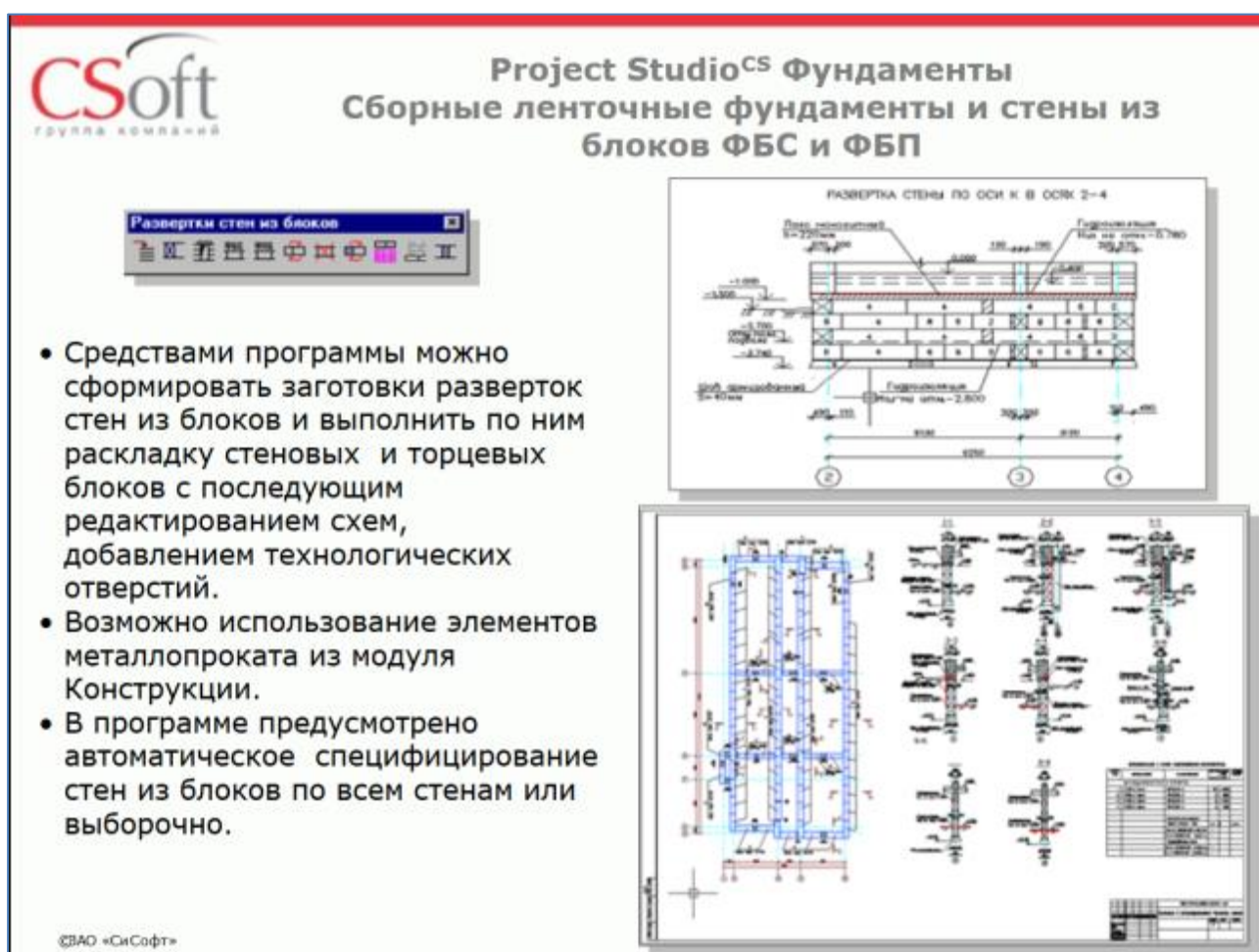


Рисунок 10 – Презентация возможностей Project Studio CS на сайте разработчиков

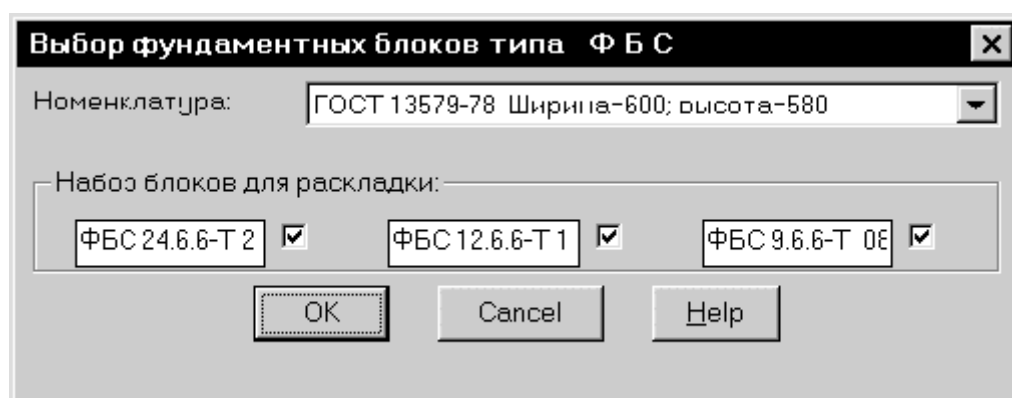


Рисунок 11 – Окно Project Studio CS для выбора применяемых в фундаменте блоков

Применяемый в данном программном обеспечении алгоритм раскладки ФБС блоков в каждой несущей стене выглядит следующим образом:

While (не конец стены) do

Begin

If (хватает места для ФБС блока длиной 2380 мм) then

установить блоки длиной 2380 мм;

else

If (хватает места для ФБС блока длиной 1180 мм) then

установить блоки длиной 1180 мм;

else

if (хватает места для ФБС блока длиной 880 мм) then

установить блоки длиной 1180 мм;

else

выполнить ручную заливку оставшегося пространства;

End;

Недостатками применения данного алгоритма являются:

- невозможность учета алгоритмом необходимости наличия в фундаменте отверстий, например, для оконных проемов, вывода коммуникаций и др.
- получаемая в результате работы алгоритма раскладка блоков не учитывает необходимость минимизации заливаемых вручную пространств между блоками. Поэтому, разработчики программного обеспечения называют получаемые, в результате работы алгоритма, схемы раскладки блоков - заготовками, требующих дальнейшего редактирования.

Другим недостатком Project Studio CS является его высокая цена, которая в сумме со стоимостью подписки AutoCAD составляет около 115000 рублей в год.

Другой программной реализацией по моделированию ленточного фундамента является Калькулятор МЗЛФ 2.2 компании ООО «МегаПлит». В отличие от предыдущих аналогов, Калькулятор МЗЛФ 2.2 производит расчет необходимых материалов с учетом основания под внутренние несущие стены (рисунок 12).

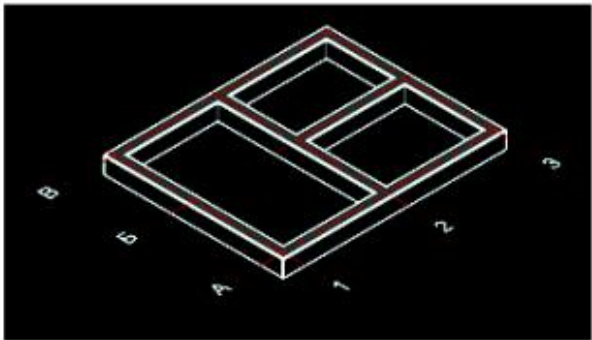
Калькулятор МЗЛФ 2.2

Шаг 2 из 6: Основные параметры будущего здания

Назад
Вперед

Введите размеры фундамента в соответствии с выбранной схемой

Схема фундамента:



Бетон М200 ▾

Примечания:
 Ширина ленты указывается Вами исходя из ширины стен здания
 Высота ленты указывается Вами исходя из высоты цокольной части, заглубления в грунт основания и обеспечения жесткости конструкций здания

Размеры в см:

300 Между осями 1-2

300 Между осями 2-3

300 Между осями А-В

300 Между осями Б-В

30 Предполагаемая ширина ленты

60 Предполагаемая высота ленты

Минимальные размеры между осями - 100 см (1м)
 Максимальные размеры между осями - 2000 см (20м)
 Ширина ленты - от 20 см до 100 см
 Высота ленты - от 20 см до 120 см
 Все размеры округляются до 10 см

Рисунок 12 – Онлайн калькулятор ленточного фундамента на сайте компании ООО «МегаПлит» (<http://www.megalit33.ru/calculators/>)

Математическую модель для расчета объема V бетона для ленточного фундамента в Калькуляторе МЗЛФ 2.2 можно выразить формулой (2):

$$V = (gf + hf) \cdot bf \cdot \sum_{i=1}^m lf_i \cdot Z, \quad (2)$$

где gf – глубина залегания фундамента, hf – высота фундамента, lf – длина несущей стены, m – количество несущих стен, Z – коэффициент запаса, в данном случае он равен 1,1 (запас – 10%).

Однако Калькулятор МЗЛФ 2.2 не позволяет подсчитывать номенклатуру и количество блоков ФБС необходимых для устройства фундамента. Также данная программная реализация не позволяет моделировать раскладку ФБС блоков.

По результатам анализа состояния вопроса была составлена сводная таблица сравнения программных реализаций по моделированию сборно-ленточных фундаментов (таблица 2).

Из проведенного анализа состояния вопроса можно сделать следующие выводы:

1) Применения ФБС блоков очень распространено в Российской Федерации при возведении сборно-ленточных фундаментов и стен подвалов.

2) Для составления проектной документация по строительству дома необходимо определить номенклатуру, количество и схему раскладки блоков (в фундаменте и/или стенах подвала здания).

3) В настоящее время существует только одна программная реализация (Project Studio CS ФУНДАМЕНТЫ) по моделирования ленточных фундаментов, которая умеет генерировать варианты раскладки ФБС блоков.

4) В существующих программных реализациях задача составления раскладки блоков никогда не рассматривалась как задача оптимизации. Поэтому генерируемые варианты раскладки всегда требуют доработки со стороны проектировщиков для минимизации пространств между блоками, требующих ручной заливки бетоном и повышающих конечную стоимость фундамента.

Таблица 2 – Сравнение программных реализаций по моделированию сборно-ленточных фундаментов

Способ моделирования фундамента	Разработчик	Сложность использования	Применяемый алгоритм	Стоимость использования	Генерирование раскладки ФБС блоков
Моделирование фундаментов «вручную» с использованием САПР по черчению (рисунок 1).	Autodesk, Inc. (программа AutoCad). Компания «Аскон» (программа Компас-3D)	Высокая	Неформализованный эвристический метод основанный на личном мировосприятии проектировщика	Лицензия на 1 год AutoCad – 55 000 руб. Коробочная версия Компас-3D V16 – 110 000 руб.	Не предусмотрено
Project Studio CS (рисунок 4).	ЗАО «СиСофт»	Высокая	По алгоритму, описанному выше	Лицензия на 1 год AutoCad (55 000 руб) + лицензия на Project Studio CS (60 000 руб)	Раскладка в виде чертежей
Калькулятор фундаментных блоков (рисунок 2)	ООО «МосПлита»	Низкая	Количественный расчет блоков по формуле (1)	Бесплатно	Не предусмотрено
Калькулятор МЗЛФ 2.2 (рисунок 3)	ООО «МегаПлит»	Низкая	Объемный расчет материалов по формуле (2).	Бесплатно	Не предусмотрено
Собственная разработка	ФГБОУ ВО ТГУ, кафедра прикладная математика и информатика	Низкая	Генетический алгоритм	Бесплатно	Раскладка в виде чертежей

Таким образом, актуальной проблемой, требующей решения можно признать - необходимость разработки алгоритма поиска оптимальной раскладки ФБС блоков с учетом исходных данных. Алгоритм должен оптимизировать номенклатуру и расположение блоков для минимизации пространств, требующих ручной заливки бетона и обеспечивать соблюдение перевязки между рядами блоков.

Таким образом, целью работы является – повышение автоматизации проектирования схем раскладки ФБС блоков за счет разработки алгоритма основанного на эволюционных вычислениях.

Поставленную цель предполагается достигнуть путем решения следующих задач:

- 1) Провести анализ состояния вопроса.
- 2) Рассмотреть проблему поиска раскладки ФБС-блоков как задачу оптимизации.
- 3) Разработать алгоритм поиска оптимальной раскладки блоков, основанный на эволюционных вычислениях.
- 4) Спроектировать, разработать и протестировать программную реализацию предложенного алгоритма.

2 Разработка алгоритма моделирования фундамента, основанного на эволюционных вычислениях

2.1 Генетические алгоритмы как метод решения задач оптимизации

Генетический алгоритм – один из алгоритмов искусственного интеллекта, позволяющий решать задачи многопараметрической оптимизации за счет направленного подбора управляющих параметров и наблюдением за тем как изменяется значение целевой функции в зависимости от параметров [6-10].

Идея работы генетического алгоритма основана на тезисах теории Дарвина, изложенной в его работе «Происхождение видов». Генетические алгоритмы являются формализованным математическим описанием следующих постулатов теории Дарвина:

1) Каждая особь популяции обладает уникальным набором генов, некоторые из которых являются полезными для выживания особи, а набор наоборот – вредными.

2) Особи с неудачным набором генов под действием внешней среды погибают, не успевая оставить потомство, благодаря чему неудачный набор генов не сохраняется. Особи с хорошим набором хромосом, являясь более приспособленными к внешней среде, выживают и скрещиваются, обеспечивая сохранение только удачных сочетаний генов.

3) Изменяющаяся внешняя среда провоцирует появление в генах случайных мутации, которые обеспечивают появление в следующем поколении особей с новыми качествами.

При переходе от терминологии теории Дарвина к задачам оптимизации принимается следующий понятийный аппарат [11-15]:

- особь O_i – набор значений управляющих параметров $O_i = (p_1, p_2, \dots, p_n)$, которые в совокупности являются одним вариантом решения задачи оптимизации;

- хромосома – значение одного из управляющих параметров p_n , обычно представляется двоичном виде посредством непозиционного двоичного кодирования (например методом Грея);
- популяция N – несколько решений рассматриваемой задачи оптимизации, найденных на данной итерации работы генетического алгоритма:

$$N = (O_1, O_2 \dots O_i)$$
- Выживаемость (приспособленность) особи – значение целевой функции y_u при данных значениях управляющих параметров $y_u = f(p_1, p_2 \dots p_n)$.

С точки зрения теории оптимизации, задачей оптимизации является та задача, в которой требуется найти наилучшее решение, удовлетворяющее ограничениям, распространяемым на область изменения управляющих параметров. Методы оптимизации направлены на поиск экстремума целевой функции $y_u = f(p_1, p_2 \dots p_n)$ (критерия оптимальности). При этом управляющие параметры $p_1, p_2 \dots p_n$ – это те параметры, изменение которых влечет за собой изменение значения целевой функции.

Ограничения оптимизационной задачи бывают следующих видов:

1) Общие ограничения, накладываемые на область значений управляющий параметров. Например:

$$p_i \geq 0, i = 1..n \text{ где } p = (p_1, p_2 \dots p_n) \quad (3)$$

2) Специальные ограничения, если они накладываются на значения функций $g(p)$, зависящих от управляющих параметров. Например:

$$g_i(p) \leq 0, \text{ где } p = (p_1, p_2 \dots p_n), i = 1..k, \quad (4)$$

где k – количество специальных ограничений.

3) Ограничения типа неравенств, накладываемые на изменение целевой функции $y_u = f(p)$. Пример:

$$f(p) \geq 0, \text{ где } p = (p_1, p_2 \dots p_n) \quad (5)$$

Таким образом, перед началом решения задачи оптимизации необходимо определить следующие ее элементы

- 1) Целевая функция.
- 2) Управляющие параметры.
- 3) Ограничения оптимизационной задачи.

После того, как данные элементы оптимизационной задачи будут определены, становится возможным для поиска ее решения применить генетические алгоритмы [15-20].

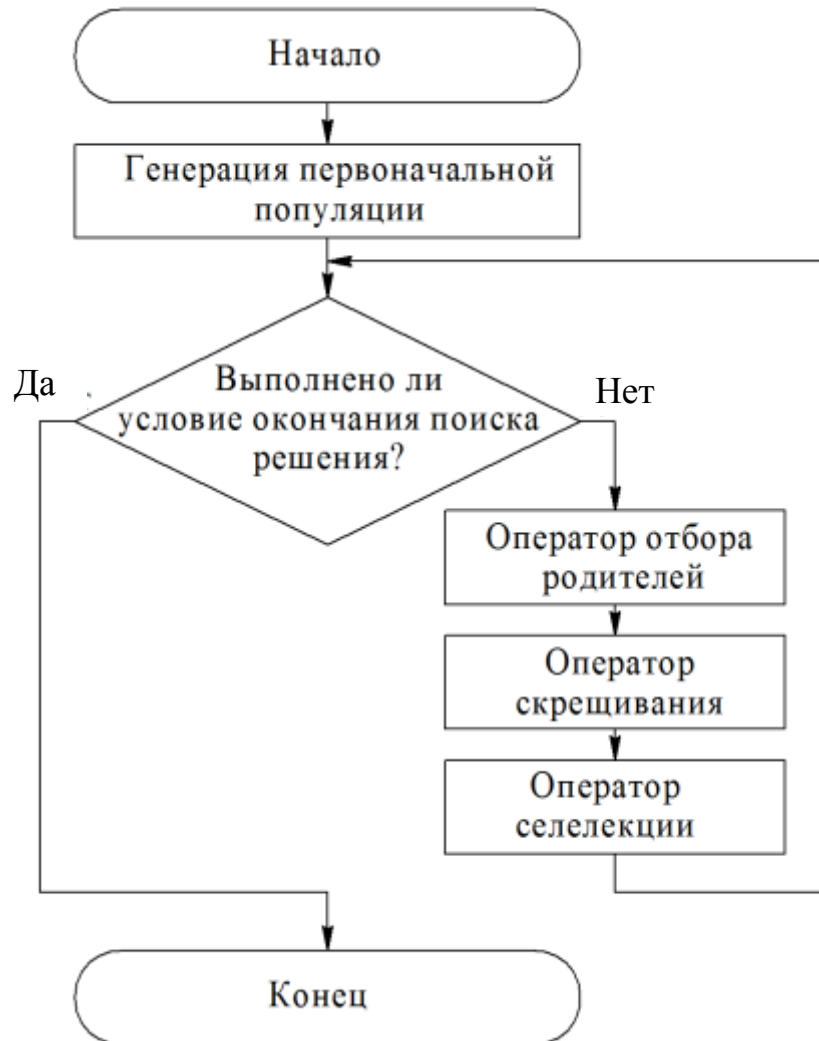


Рисунок 13 – Схема работы генетического алгоритма

Генетический алгоритм состоит из следующих этапов:

- генерация первоначальной популяции;
- оператор отбора родителей;
- оператор скрещивания;
- оператор селекции;
- проверка условия окончания поиска решения.

Блок-схема работы генетического алгоритма представлена на рисунке 13.

Генерация первоначальной популяции необходима для получения набора возможных решений решаемой оптимизационной задачи. Генерация первоначальной популяции должно происходить случайным образом для того, чтобы обеспечить популяцию как можно более разнообразным генетическим материалом. Чем разнообразней сгенерированные решения, тем большая область в пространстве решений будет подвергнута анализу на последующих итерациях работы генетического алгоритма. Тем не менее, генерируемые решения должны удовлетворять ограничениям оптимизационной задачи, в противном случае данный вариант решения не включается в первоначальную популяцию.

В генетическом алгоритме возможно применение разных операторов отбора особей для скрещивания – панмиксия, инбридинг, аутбридинг, селекция, турнирный отбор, метод рулетки и другие способы. Это связано с тем, что решаемые задачи оптимизации различны, т.е. в ней может быть один глобальный экстремум и несколько; при поиске экстремума целевой функции в задаче могут встречаться локальные ловушки, или наоборот их может не быть. Поэтому для каждой конкретной задачи эффективным может оказаться одного из операторов отбора особей для скрещивания. Так при использовании панмиксии двое родителей R_1, R_2 из популяции N , включающей в себя набор особей $O_1, O_2 \dots O_i$ выбираются случайным образом :

$$\begin{aligned} R_1 &\leftarrow \text{random}(O_1, O_2 \dots O_i) \\ R_2 &\leftarrow \text{random}(O_1, O_2 \dots O_i) \end{aligned} \quad (6)$$

где i – размер популяции.

При использовании инбридинга родители R_1, R_2 выбираются так:

$$\begin{aligned} R_1 &\leftarrow \text{random}(O_1, O_2 \dots O_i) \\ R_2 &\leftarrow O_j \text{ if } fh(R_1, O_j) = \min \end{aligned} \quad (7)$$

где $j=1...i$, $R_1 \neq R_2$, i – размер популяции, fh – функция расчета расстояния Хемминга между двумя особями, а $\min$ – функция поиска наименьшего значения.

При использовании аутбридинга родители R_1, R_2 выбираются так:

$$\begin{aligned} R_1 &\leftarrow \text{random}(O_1, O_2 \dots O_i) \\ R_2 &\leftarrow O_j \text{ if } fh(R_1, O_j) = \max \end{aligned} \quad (8)$$

где $j=1...i$, $R_1 \neq R_2$, $\max$ – функция поиска максимального значения.

При использовании селективного отбора родители выбираются так:

$$\begin{aligned} f_{\text{ср}} &= \frac{\sum_{k=1}^i f_u(O_k)}{i} \\ R_1 &\leftarrow O_j \text{ if } f_u(O_j) > f_{\text{ср}} \\ R_2 &\leftarrow O_h \text{ if } f_u(O_h) > f_{\text{ср}} \end{aligned} \quad (9)$$

где $R_1 \neq R_2$, а f_u – значение целевой функции данной особи, i – размер популяции. При этом значения j и h определяются так: $j=\text{random}(1,i)$, $h=\text{random}(1,i)$, где random – функция определения случайного целого числа на интервале от 1 до i .

При использовании отбора родителей по методу рулетки они выбираются случайным образом, однако вероятность P особи с номером k быть выбранной в качестве родителя рассчитывается так:

$$P_k(O_k) = \frac{f_u(O_k)}{\sum_{j=1}^i f_u(O_j)}, k = 1...i, \quad (10)$$

где i – размер популяции, f_u – значение целевой функции данной особи.

Операторы скрещивания предназначены для того, чтобы на основе двух старых решений (особи-родители) получать новые решения (особи-потомки) оптимизационной задачи. Для этого от особей-родителей в особи-потомки копируются фрагменты решений. Таким образом, получаются новые решения оптимизационной задачи. Операторы скрещивания бывают двух видов – дискретные и бинарные, причем каждый из этих видов делиться на подвиды.

После выполнения скрещивания особей популяция больше чем был ее первоначальный размер. Это объясняется тем, что в ней содержатся все первоначальные особи, а также особи, получившиеся в результате скрещивания. Для того, чтобы уменьшить размер популяции до первоначального размера применяют операторы селекции. Обычно в результате селекции из популяции убирают все худшие решения и оставляют только лучшие решения. Это делается для того, чтобы на каждой следующей итерации работы генетического алгоритма получались все лучшие и лучшие решения оптимизационной задачи.

Селекция особей может применяться по методу усечения. В этом случае все особи O (количество которых после скрещивания становится больше, чем i) сортируются в порядке убывания целевой функции (функции приспособленности). Затем в новую популяцию Q размер, которой равен i , особи выбираются путем повторения данных шагов i раз ($k=1 \dots i$):

$$\begin{aligned} t &\leftarrow \text{random}(1, \lceil \mu \cdot i \rceil) \\ Q_k &\leftarrow O_t \end{aligned} \quad (11)$$

где μ – управляющий параметр, выбирается из диапазона $(0, 1]$. *random* – функция нахождения случайного целого числа на данном заданном диапазоне

Также селекция может происходить с использованием элитарного отбора. В этом случае все особи O также сортируются в порядке убывания целевой функции. В новую популяцию Q размер, которой равен i , особи выбираются путем повторения данных шагов i раз ($k=1 \dots i$):

$$\begin{aligned} Q_k &\leftarrow O_k \text{ if } k \leq \lceil \mu \cdot i \rceil \\ Q_k &\leftarrow \text{generate} \text{ if } k > \lceil \mu \cdot i \rceil \end{aligned} \quad (12)$$

где i – первоначальный размер популяции, *generate* – процедура генерирования новой особи, μ – управляющий параметр, выбирается из диапазона $[0.1, 0.8]$.

Также селекция может осуществляться по методу отжига. В этом случае в новую популяцию Q размер, которой равен i , особи выбираются путем повторения данных шагов i раз ($k=1 \dots i$):

$$\begin{aligned}
m &\leftarrow \text{random}(1, s) \\
n &\leftarrow \text{random}(1, s) \\
l &\leftarrow \text{random}(0, 1) \\
t &\leftarrow \frac{1}{1 + \exp\left(\frac{f_u(O_m) - f_u(O_n)}{T}\right)}, \\
Q_k &\leftarrow O_m \quad \text{if } t > l \\
Q_k &\leftarrow O_n \quad \text{if } t \leq l
\end{aligned} \tag{13}$$

где T – управляющий параметр, s – размер популяции, содержащей исходные особи и особи-потомки, f_u – значение целевой функции для данной особи O .

Существуют и другие варианты оператора селекции особей.

После выполнения полонения оператора селекции размер популяции становится равен исходному размеру. Следующим шагом выполнения генетического алгоритма является проверка на выполнение условия окончания поиска решения. Такими условиями являются:

- 1) В популяции содержится решение, отвечающее всей совокупности предъявляемых требований, поиск лучшего решения не требуется.
- 2) За последние несколько итераций работы генетического алгоритма, у лучших решений популяции критерий оптимальности улучшить не удастся.

Чтобы разработать подходы по применению генетического алгоритма для моделирования ленточного фундамента необходимо рассмотреть эту задачу с точки зрения теории оптимизации.

2.2 Моделирование ленточного фундамента как задача оптимизации

Выделим основные элемент оптимизационной задачи моделирования раскладки ФБС блоков – целевая функцию, управляющие параметры, ограничения оптимизационной задачи.

Самой оптимальной является такая раскладка ФБС блоков, которая обеспечивает полное отсутствие не закрытых промежутков между блоками. Это связано с тем, что незакрытые промежутки между ФБС блоками необходимо вручную заливать бетоном (или закладывать кирпичом), что в конечном итоге

повышающих конечную стоимость фундамента и увеличивает время его возведения (рисунок 14).



Рисунок 14 – закрытие промежутков между блоками с помощью ручной кладки кирпича

По этой причине целевую функцию f_y целесообразно сделать зависимой от суммарного объема V промежутков требующих ручного заполнения. Тогда задача оптимизации будет выглядеть так:

$$f_y(V) \rightarrow \min \quad (14)$$

При этом, с учетом того, что толщина фундамента имеет фиксированную величину (0,3 м/0,4 м/0,5 м/0,6 м) и высота одного ряда блока также имеет фиксированную величину (0,58 м), то объем V фактически зависит только от суммы длин незакрытых блоками промежутков:

$$V(L) = 0,58 \cdot 0,4 \cdot L \quad (15)$$

По этой причине задачу оптимизации можно представить как:

$$\begin{aligned} f_y(L) &= \sum_{k=1}^m L_k, \\ f_y(L) &\rightarrow \min \end{aligned} \quad (16)$$

где m – количество незакрытых промежутков между блоками, L_k – длина k -того промежутка.

При работе с генетическим алгоритмом удобней искать максимумы целевой функции, поэтому выражения (16) преобразуем следующим образом:

$$f_y(L) = \frac{100}{1 + \sum_{k=1}^m L_k}, \quad (16)$$

$$f_y(L) \rightarrow \max$$

Таким образом, в наилучшем из возможных случаев, в варианте решения с отсутствием промежутков между блоками ($\sum_{k=1}^m L_k = 0$) значение целевой функции будет равно 100. Т.е. целевая функция может принимать значения: $f_y \in (0; 100]$.

При поиске оптимальной раскладки ФБС блоки должны располагаться с перевязкой вертикальных швов, глубина которых должна быть не менее 0,4 высоты блока (т.е. $0,4 \cdot 0,58 = 0,232\text{ м}$) при малосжимаемых грунтах и не менее высоты блока (т.е. $0,58\text{ м}$) при сильносжимаемых грунтах (рисунок 15).

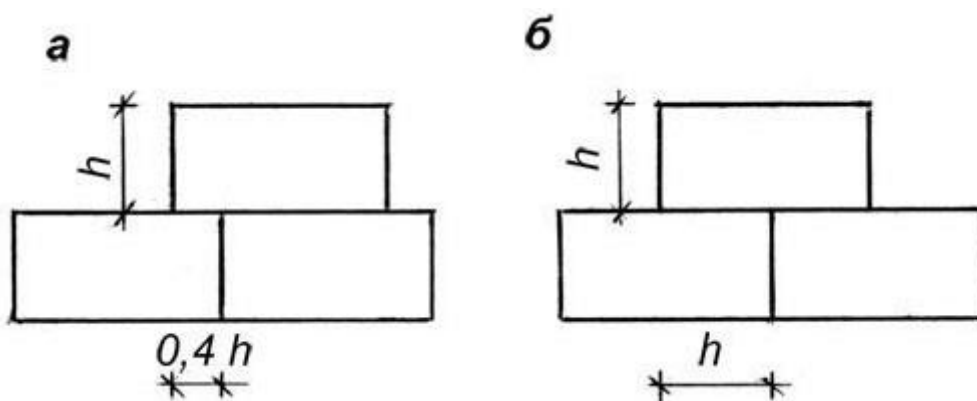


Рисунок 15 – Требования СНиП по перевязке блоков ФБС: а – при малосжимаемых грунтах, б – при сильносжимаемых грунтах.

По этой причине каждый вариант раскладки блоков необходимо проверять на соответствие требований по перевязке. Введем переменную Per ,

которая будет отвечать за результат проверки раскладки требованию по перевязке. Если перевязка соблюдена, то $Per=true$, в противном случае $Per=false$. Если, перевязка не соблюдена, то фундамент не будет обладать требуемой прочностью на изгиб и поэтому ценность такого решения раскладки блоков минимальна ($f_u = 0$).

Учтем это при определении значения целевой функции f_u :

$$f_u(L) = \begin{cases} \frac{100}{1 + \sum_{k=1}^m L_k} & \text{if } Per = true \\ 0 & \text{if } Per = false \end{cases}, \quad (17)$$

$$f_u(L) \rightarrow \max$$

где m – количество незакрытых промежутков между блоками, L_k – длина k -того промежутка. При этом целевая функция может принимать следующие значения $f_u \in [0;100]$.

Определим ограничения оптимизационной задачи:

1) Ограничения связанные размером фиксированным размером ФБС блоков. Длина ФБС блоков фиксирована ГОСТом и бывает 3 размеров – 880 мм, 1180 мм, 2380 мм (таблица 1). Не допускается укорачивание блоков так как в этом случае происходит удаление гидроизолирующего замка с одной из двух сторон и срез арматуры.

2) Ограничения связанные с расположением блоков в фундаменте. Не допускается выпирание блоков за границу фундамента. Расположение блоков на углах фундамента должно быть таким, как это представлено на рисунке 16. Расположение блоков и допустимое расположение пустот в раскладке в местах примыкания показано на рисунке 17.

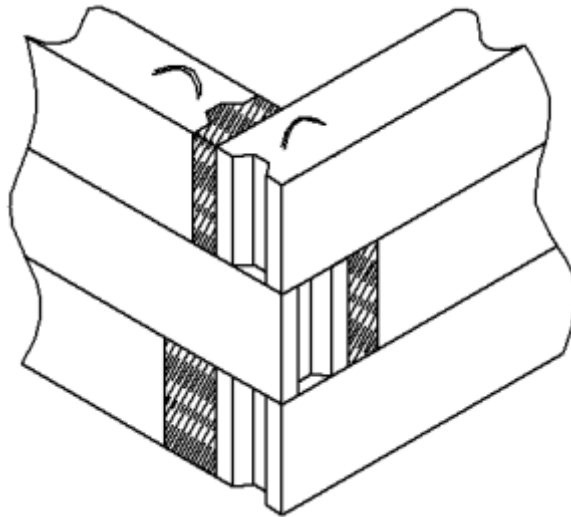


Рисунок 16 – Расположение блоков на углах, штриховкой показаны места с допустимым расположением пустот в раскладке

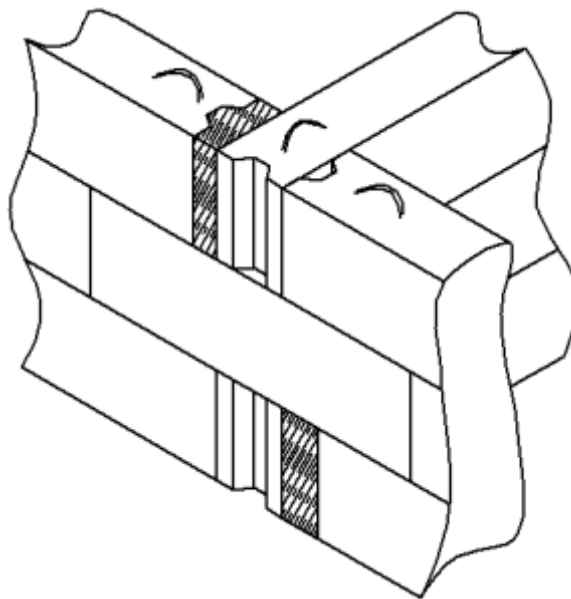


Рисунок 17 – Расположение блоков в примыканиях, штриховкой показаны места с допустимым расположением пустот в раскладке

После того как определены целевая функция, управляющие параметры и ограничения оптимизационной задачи, опишем предлагаемые подходы по применению генетического алгоритма для поиска оптимальной раскладки.

Предположим, что нам требуется найти оптимальную раскладку блоков для фундамента с внешними габаритами A и B и высотой в три ряда блоков (рисунок 18).

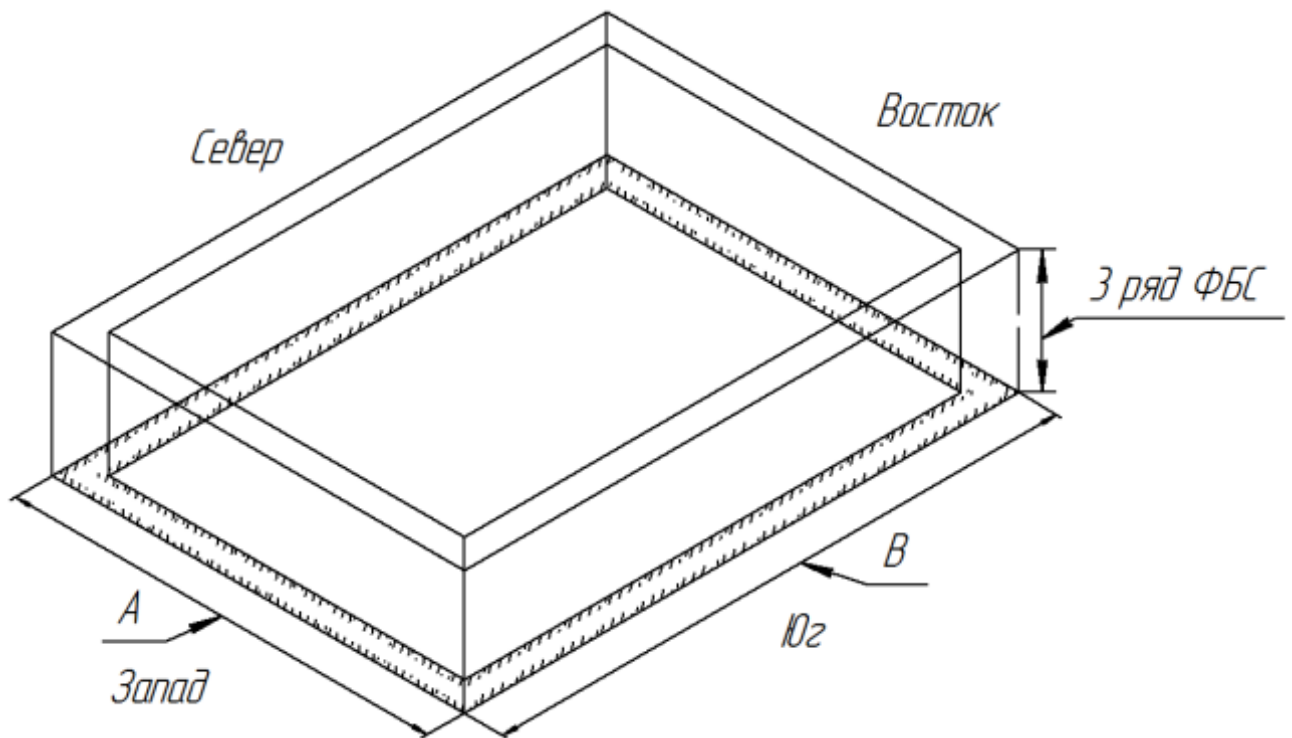


Рисунок 18 – Одна из возможных конфигурация фундамента

После того, как известна конфигурация требуемого фундамента необходимо проанализировать задачу с целью уменьшения вычислительных затрат.

Для этого сначала производится поиск стен фундамента имеющих одинаковый размер и конфигурацию. На рисунке 16 одинаковый размер и конфигурацию имеют западная и восточная стенки фундамента, а также северная и южная стенки. Очевидно, что если будет найдена для западной стены оптимальная раскладка блоков, то такую же раскладку без изменений можно применить и для восточной стены. Аналогично следует поступить с северной и южной стеной фундамента. Таким образом, объем решаемой задачи уменьшился вдвое. При этом задачу поиска оптимальной раскладки можно представить в виде двух подзадач – поиск раскладки блоков западной стены (рисунок 19) и южной стены (рисунок 20).

“Ступенчатость” на концах стен не обходима для обеспечения перевязки углов, представленной на рисунке 16.

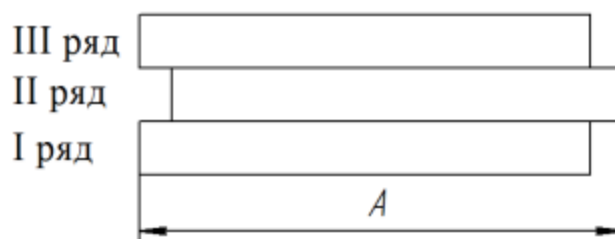


Рисунок 19 – Подзадача по поиску оптимальной раскладки для западной стены
фундамента

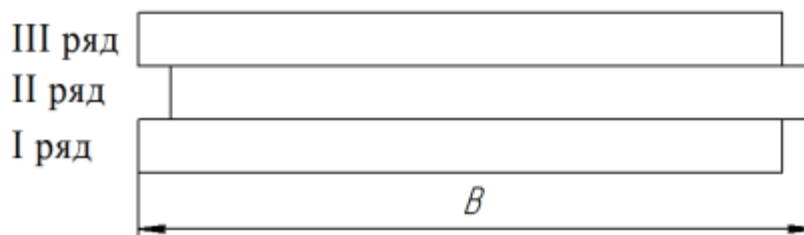


Рисунок 20 – Подзадача по поиску оптимальной раскладки для южной
стены фундамента

Для каждой уникальной стенки фундамента (в данном случае уникальных - две) производится определение количества уникальных рядов, в рамках которых требуется определение раскладки блоков. Очевидно, что если найти оптимальную раскладку блоков для I и II рядов (которые будут соответствовать требованиям перевязки), то раскладку III ряд искать нет необходимости – ее можно принять равной раскладке первого ряда (рисунок 21).

На рисунке 21 жирным контуром показаны два ряда блоков, для которых требуется выполнить поиск оптимальной раскладки. С помощью такого подхода объем вычислений при поиске оптимальной раскладки блоков фундамента с помощью генетического алгоритма снижается на треть.

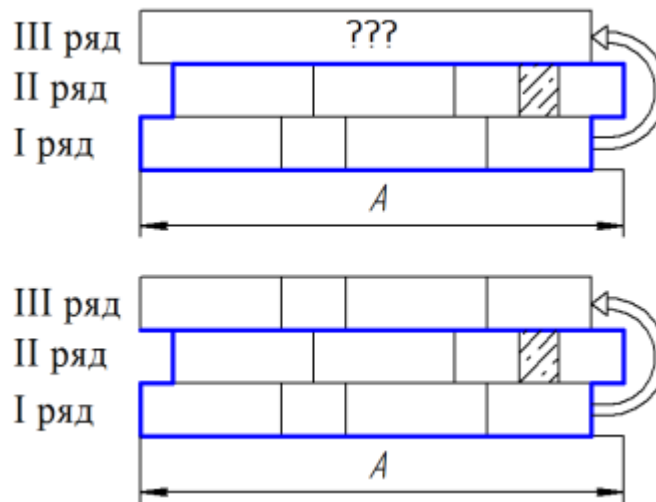


Рисунок 21 – Снижение объема вычислений для задачи поиска оптимальной раскладки блоков

Результат использование такого же подхода для более сложной конфигурации стены состоящей из 5 рядов и содержащей окно, представлен на рисунке 22. Жирным контуром показаны ряды, для которых требуется найти оптимальную раскладку блоков. III ряд можно скопировать в I и V ряды. В этом случае объем вычислений удастся снизить на $2/5$.

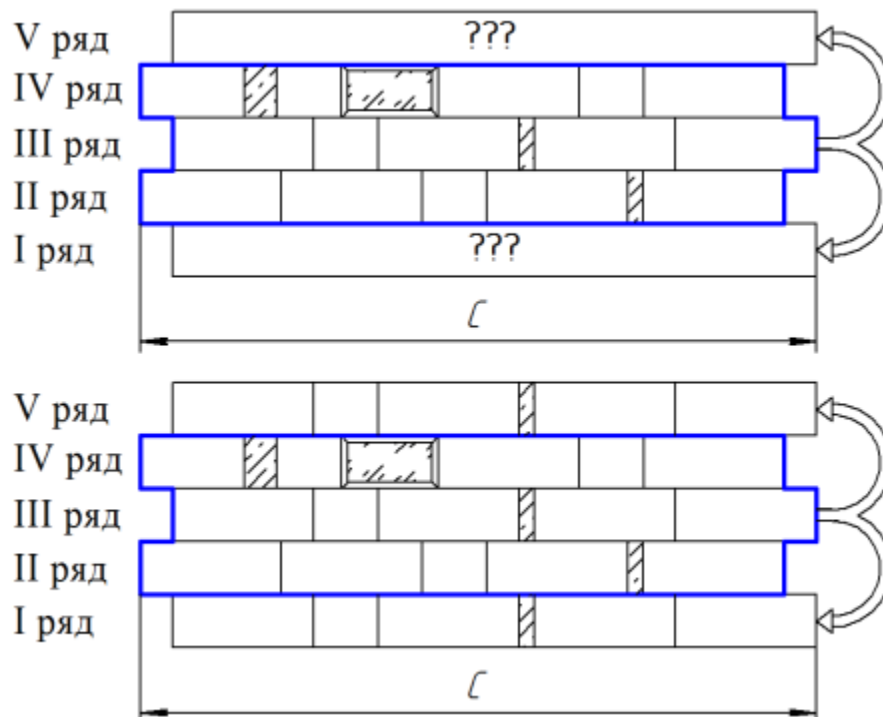


Рисунок 22 – Снижение объема вычислений для задачи поиска оптимальной раскладки блоков для стены сложной конфигурации (5 рядов, 1 окно)

Теперь, когда определено количество уникальных рядов, для которых требуется найти оптимальную раскладку, рассмотрим предложенные решения по адаптации генетического алгоритма к данной задаче.

2.3 Адаптация генетического алгоритма под решаемую задачу

Как было сказано выше, генетический алгоритм состоит из следующих этапов: Генетический алгоритм состоит из следующих этапов: генерация первоначальной популяции; выполнение оператора отбора родителей; выполнение оператора скрещивания; выполнение оператор селекции; проверка условия окончания поиска решения (рисунок 12).

Генерация особей в первоначальную популяцию на примере решения подзадачи представленной на рисунке 19. Генерация особей в первоначальную популяцию состоит из двух этапов.

Первый этап. В каждом ряду стены определяются места, где не допускается расположение промежутков, не закрытых блоками. Такими местами являются:

- выступающий край каждого ряда блоков - позиции 1,4,5 рисунка 23;
- оба края оставляемых технических проемов (оставляемых, например, под окна или коммуникации) - позиции 2,3 рисунка 21;

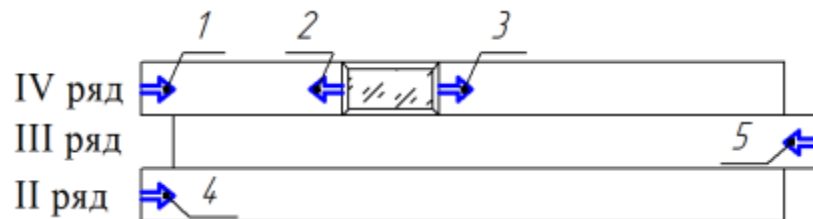


Рисунок 23 – Первый этап генерации решений – определение мест, где требуется расположить блоки в первую очередь

Последовательно проходя по этим местам, производится выставление по одному блоку случайно выбранной длины. Если данный блок не помещается, то выбирается блок с меньшим размером. Результат выполнения первого этапа представлен на рисунке 24.

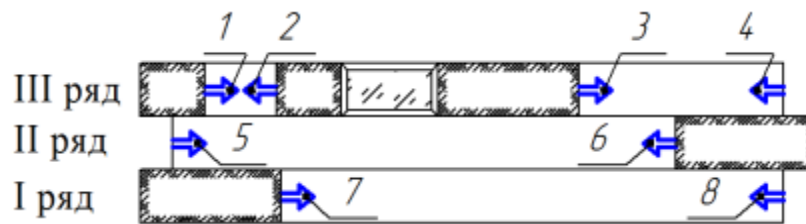


Рисунок 24 – Результат выполнения первого этапа генерации решений

Второй этап. Заключается в том, что случайным образом выбирается один из свободных концов блока (позиции 1,2,3,6,7 рисунка 24) или один из концов ряда (позиции 4,5,8 рисунка 24) и вплотную к нему выставляется блок случайно выбранной длины. Если блок выбранной длины не помещается в данное пространство, то выбирается блок меньшей длины. Второй этап выполняется до тех пор, пока не останется место под установку даже самого маленького по длине блока. Считается, что оставшееся не заполненное пространство будет заделываться рабочими вручную. Результат выполнения второго этапа представлен на рисунке 25.



Рисунок 25 – Результат выполнения второго этапа генерации решений

Генерация вариантов раскладки блоков выполняется до тех пор, пока размер популяции не достигнет заданного значения (50 штук).

В качестве оператора отбора особей на роль родителей используется панмиксия (6). Из популяции выбираются два варианта раскладки блоков случайным образом, затем производится их скрещивание. В результате одного скрещивания получают 2 особи-потомка. Для скрещивания формируется 50 пар особей.

Скрещивание особей производится по типу многоточечного кроссинговера. Скрещивание производится в два этапа.

Этап первый. В зависимости от конфигурации стенки фундамента определяются наименьшие фрагменты решений, которые можно скопировать в особи-потомки. Такими фрагментами могут быть:

- весь ряд блоков, если в нем не предусмотрено устройство технических проемов (под окна, под вывод коммуникаций и т.д.) – позиции III и IV на рисунке 26;
- фрагмент ряда блоков находящийся между техническим проемом и концом ряда блока - позиция I и II на рисунке 26;
- фрагмент ряда блока находящегося между двумя техническими проемами.

Результат определения таких фрагментов для рассматриваемого примера приведен на рисунке 26.

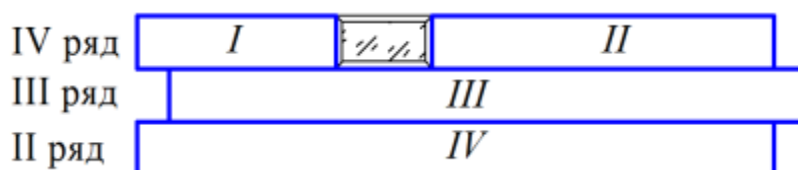


Рисунок 26 – Результат определения фрагментов решений для обмена при выполнении скрещивания

Второй этап. Для каждой пары особей, подготовленных при выполнении оператора выбора родителей случайным образом генерируются 2 потомка. Для этого из первого родителя копируется часть случайно выбранных фрагментов в первого потомка, а остальная часть во второго потомка. Все остальные недостающие фрагменты решений копируются из второго родителя.

Пример выполнения второго этапа представлен на рисунке 27.

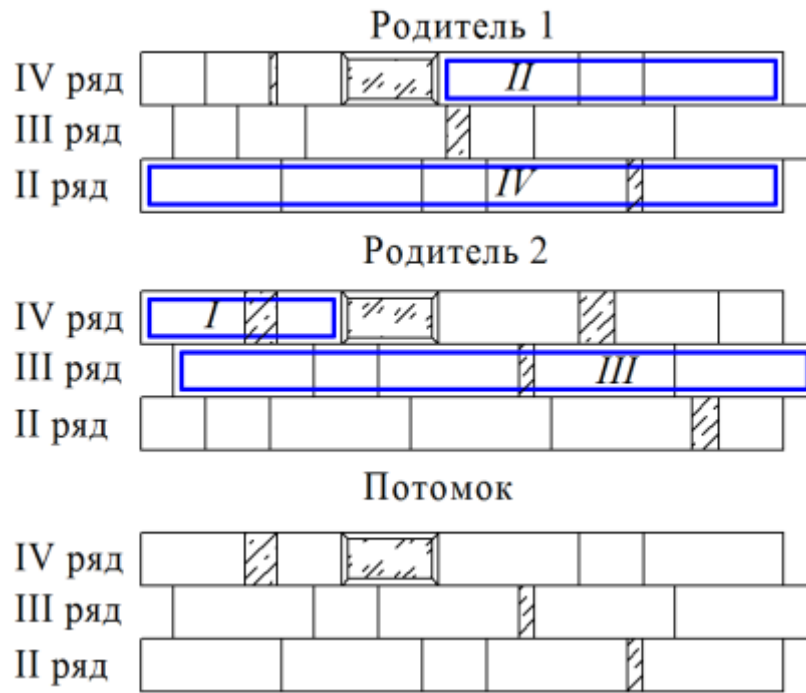


Рисунок 27 – Пример выполнения оператора скрещивания

После выполнения оператора скрещивания первоначальная популяция, состоящая из 50 особей, пополняется особями потомками в количестве равном 100 штук. Для того чтобы порадить данную популяцию применяется оператор селекции.

В нашем случае применяется элитарного отбора. Все особи O сортируются в порядке убывания целевой функции. В новую популяцию Q размер i , которой равен 50, особи выбираются путем повторения данных шагов i раз, где k – счетчик, который увеличивает на каждой итерации на единицу ($k=1 \dots i$):

$$\begin{aligned} Q_k &\leftarrow O_k \text{ if } k \leq \lceil \mu \cdot i \rceil \\ Q_k &\leftarrow \text{generate if } k > \lceil \mu \cdot i \rceil \end{aligned} \quad (18)$$

где *generate* – процедура генерирования новой особи по методу, описанному выше, μ – управляющий параметр, выбирается из диапазона $[0.1, 0.8]$, в нашем случае он равен 0,1.

Условием остановки выполнения генетического алгоритма является - отсутствие увеличения значение целевой функции у лучшей особи из популяции на протяжении 5 итераций генетического алгоритма.

3 Практическая реализация

3.1 Программная реализация алгоритма

В результате выполнения выпускной квалификационной работы в среде разработки Eclipse на языке программирования Java (рисунок 28) была спроектирована и изготовлена программа для моделирования сборно-ленточного фундамента, обладающими характеристиками:

- наличие интуитивно-понятного графического интерфейса пользователя;
- использование генетического алгоритма для генерирования вариантов раскладки ФБС блоков;
- наличие в программе основных шаблонов фундаментов, применяемых при малоэтажном строительстве с возможностью указания желаемых размеров;
- определение раскладки фундаментных блоков с учетом требований СНиП по перевязке блоков и минимизации объема доливки бетона для заполнения расстояний между блоками;
- функция учета необходимости наличия в фундаменте технологических проемов для оконных проемов, вывода коммуникаций, примыкания перпендикулярных стен;
- генерирование чертежей по раскладке блоков в формате png.

UML диаграммы классов приведены в приложении А.

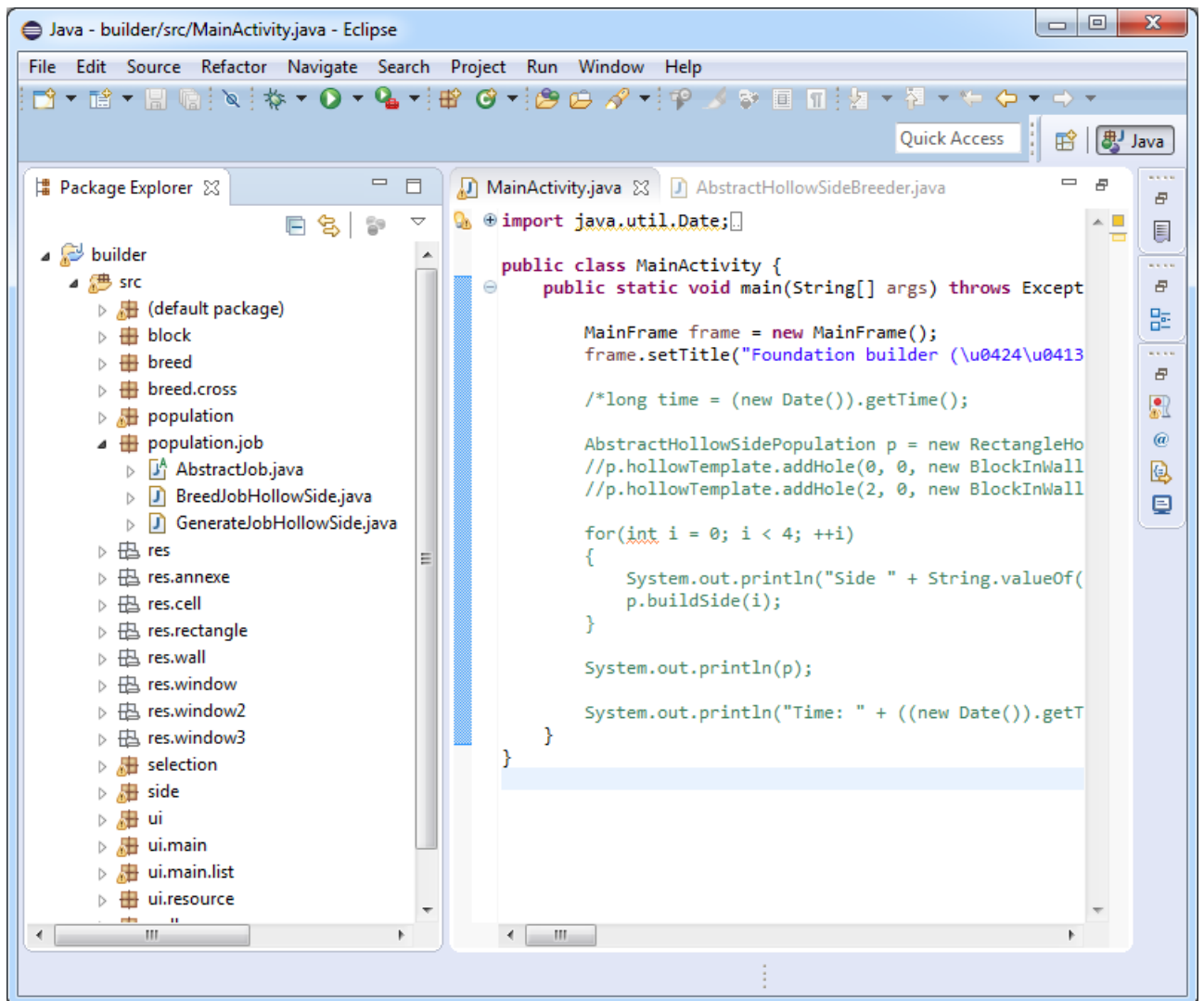


Рисунок 28 – Проект приложения в среде Eclipse

При запуске приложения перед пользователем открывается главное окно программы (рисунок 29). Главное окно состоит из четырех элементов управления. Левую часть программы занимает список наиболее распространённых форм фундамента. Правую верхнюю часть занимает панель, позволяющая в миллиметрах задавать геометрические размеры для выбора конфигурации фундамента (рисунок 30).

В правой нижней части программы расположена группа элементов управления, позволяющих определять необходимость наличия технологических проемов в фундаменте под окна (если предусмотрен подвал), под вывод и ввод

коммуникаций и т.д. Вместе с тем, технологические проемы в стенах фундамента для выполнения перевязки блоков в углах и примыканиях (рисунок 16 и 17) задавать не требуется – они учитываются программой автоматически.

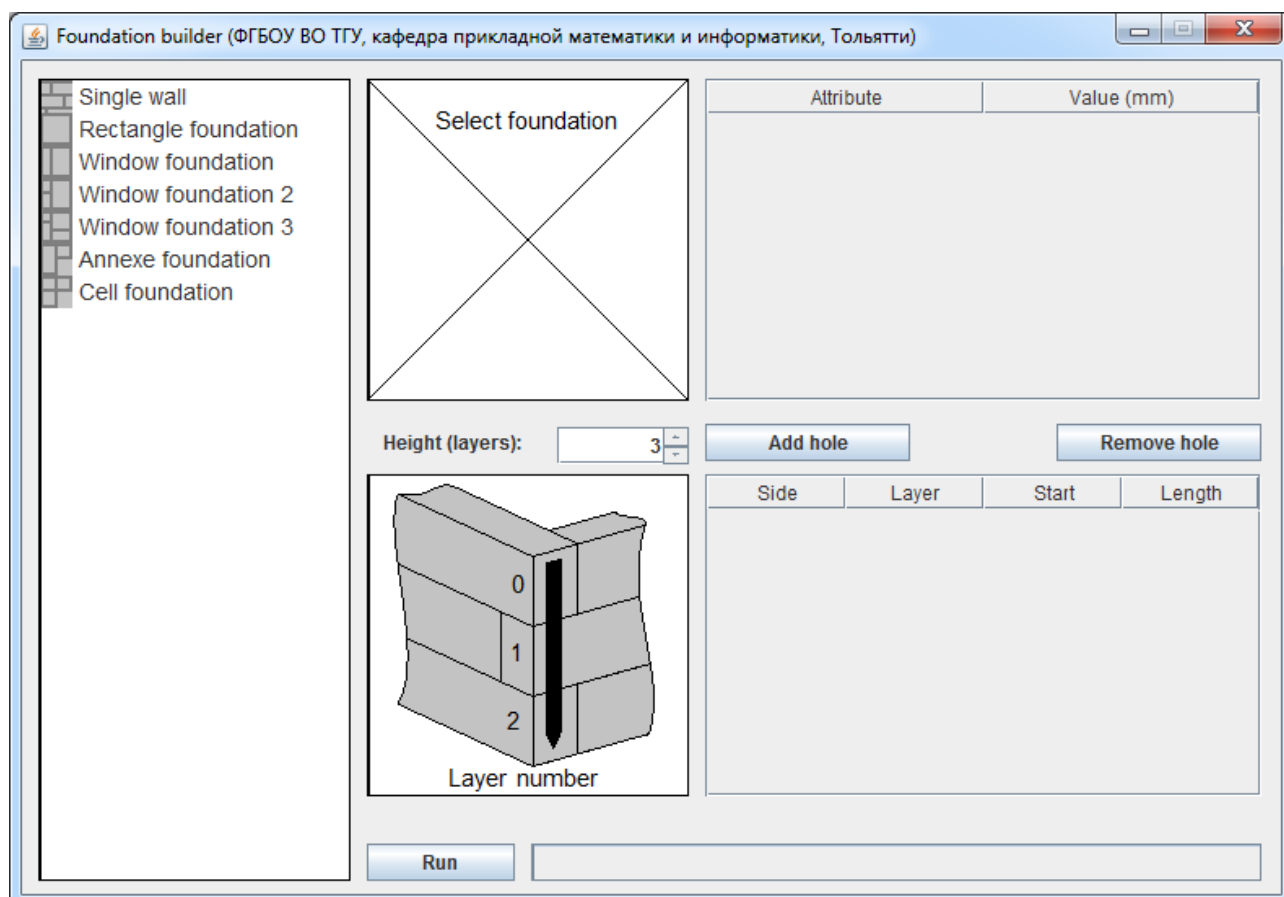


Рисунок 29 – Главное окно программы

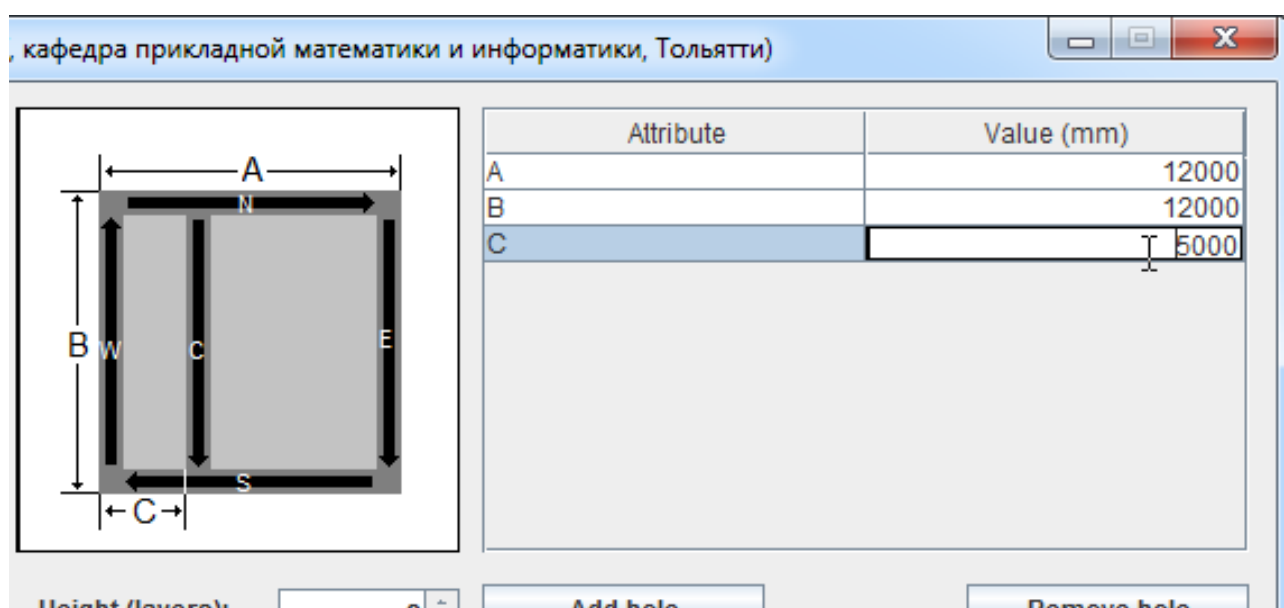


Рисунок 30 – Задание размеров фундамента

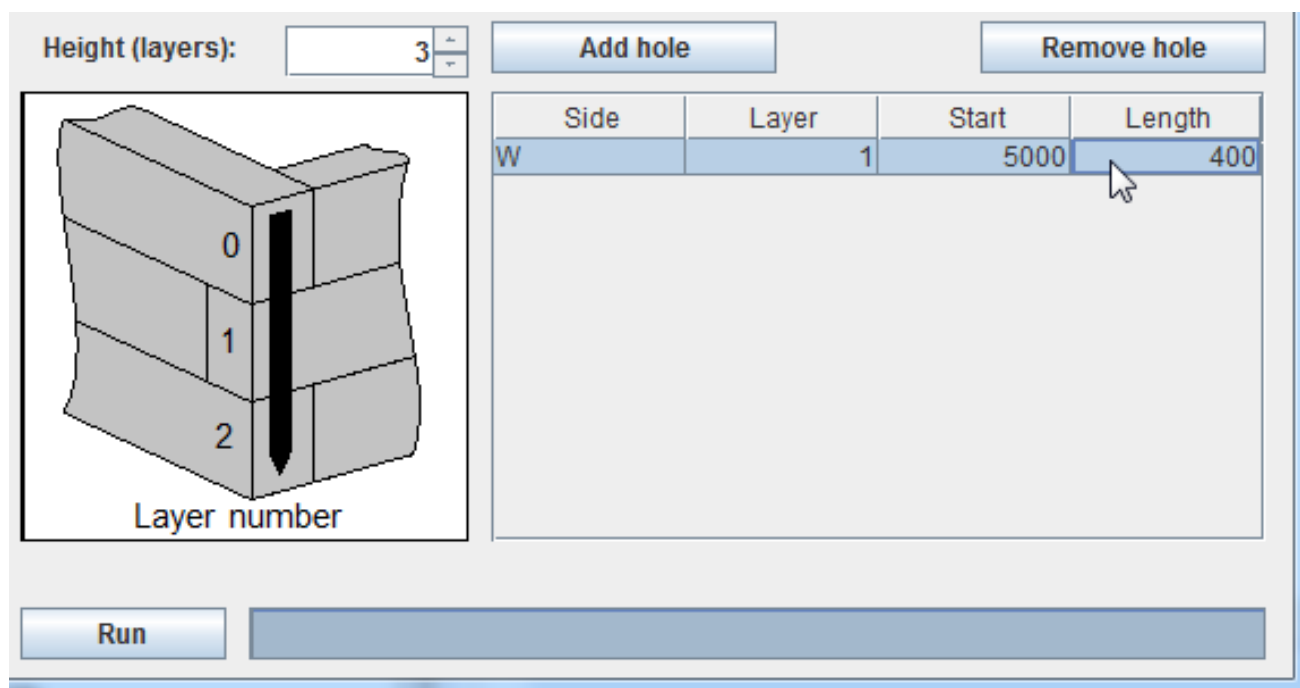


Рисунок 31 – Задание технологических проемов на любой стороне стены фундамента.

В программе предусмотрена возможность задания следующих конфигураций фундамента (рисунок 32). Однако если требуется генерирования фундамента иной формы, то в программе можно генерировать раскладку блоков для стен для каждой стены сложного фундамента по отдельности (рисунок 32, а).

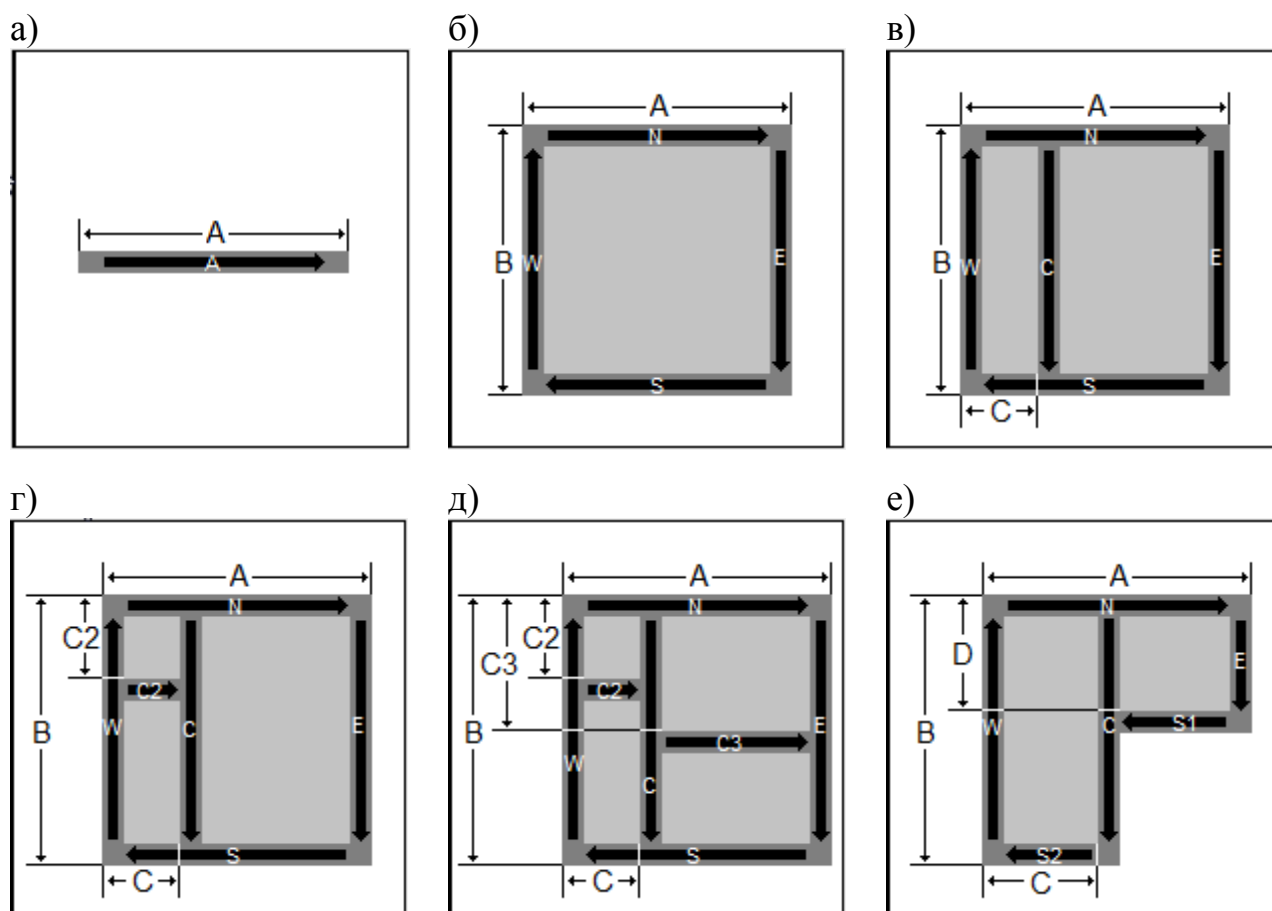


Рисунок 32 – Варианты конфигурации сборно-ленточных фундаментов предусмотренных в разработанном приложении: а – прямой участок, б – прямоугольный фундамент, в – пятистенок, г – пятистенок с одной перемычкой, д – пятистенок с двумя перемычками, е – г-образный фундамент

3.2 Пример работы программы

Предположим что, требуется сгенерировать раскладку фундаментных блоков для частного дома с габаритными размерами 12 на 12 метров. Для того, чтобы удобно было перекрывать этажи плитами перекрытия закладывается центральная стена фундамента (плиты перекрытия имеют длину от 5 до 7 метров). На западной стене фундамента на расстоянии 5 метров от края необходимо предусмотреть наличие технологического проема шириной 40 см для ввода/вывода коммуникаций. Требуется найти оптимальную раскладку ФБС блоков для выполнения фундамента с соблюдением правил перевязки.

Графически такую задачу можно представить так как это показано на рисунке 33.

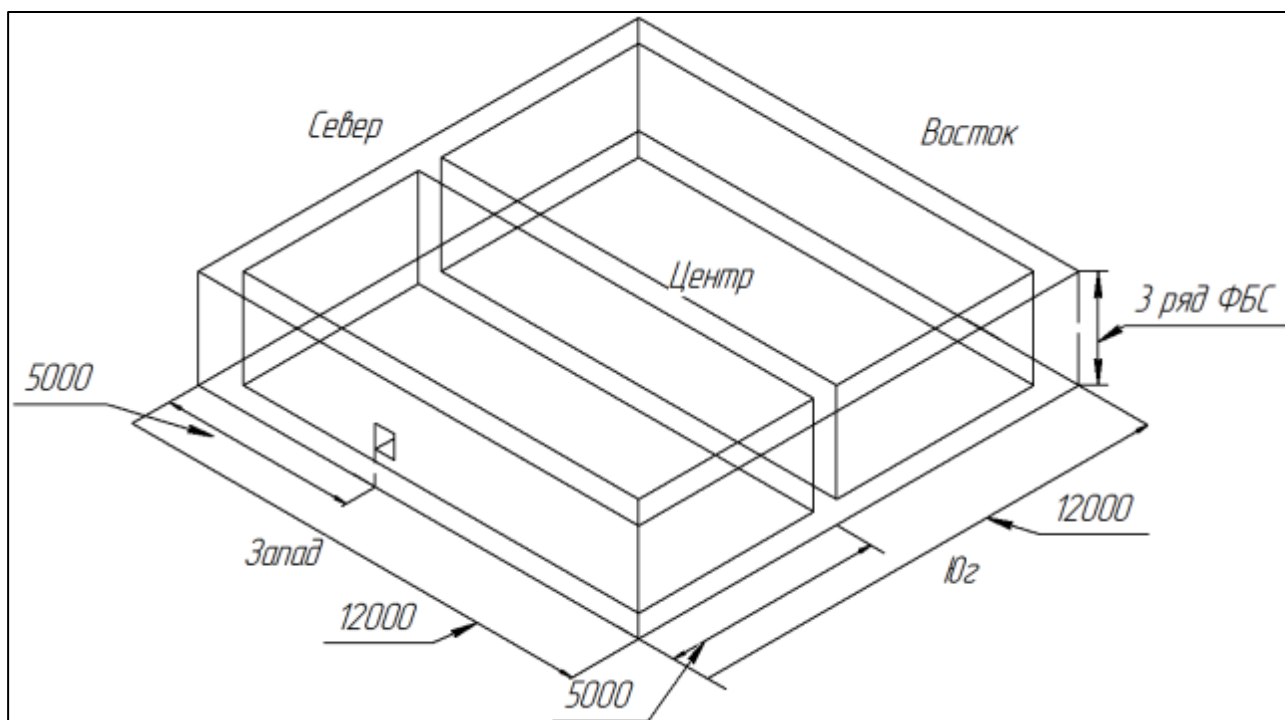


Рисунок 33 – Постановка задачи для поиска оптимальной раскладки.

Откроем разработанное приложение, выберем конфигурацию фундамента, зададим размеры фундамента и определим необходимость

наличия технологического отверстия под коммуникации в западной стене (рисунок 34). Затем запустим поиск оптимальной раскладки.

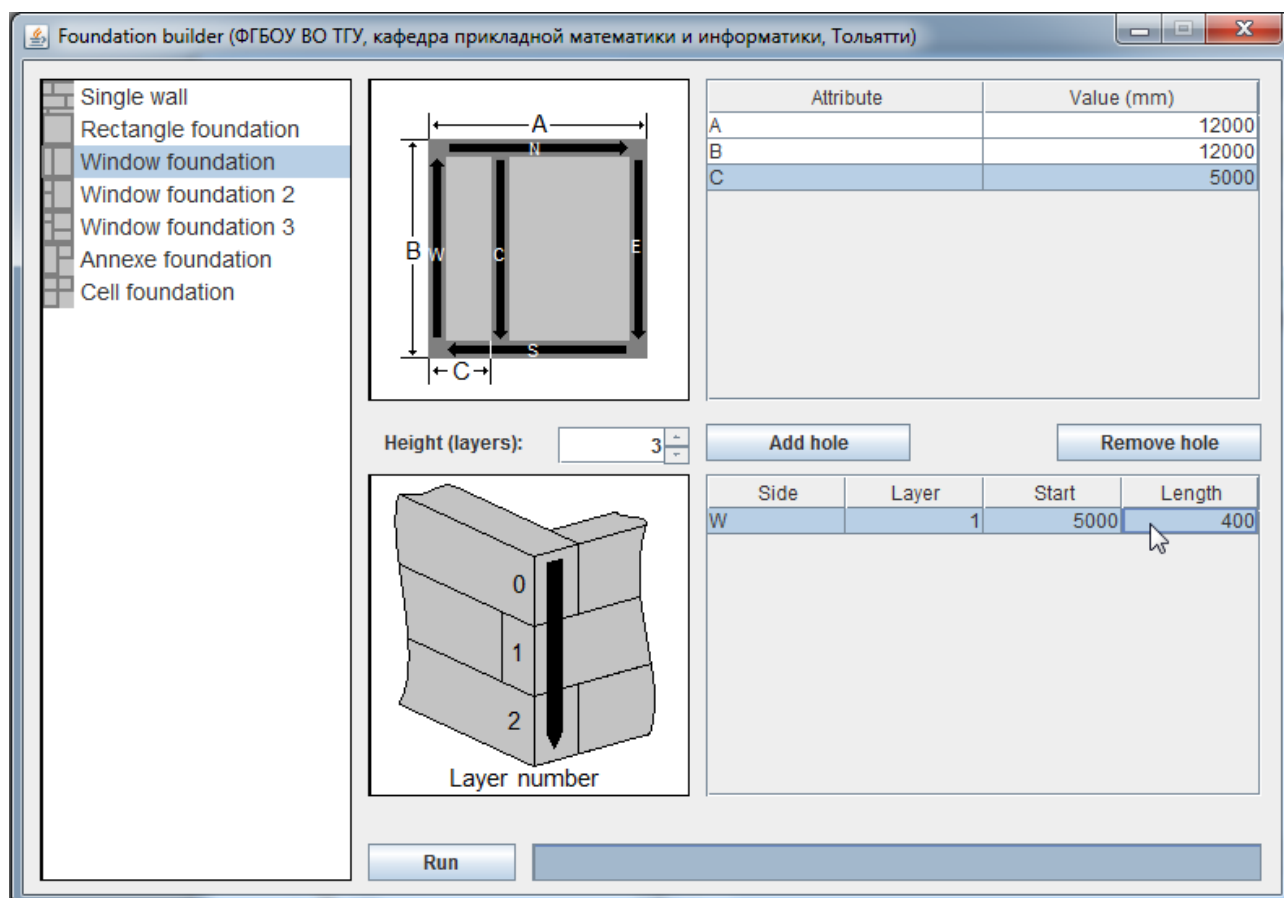
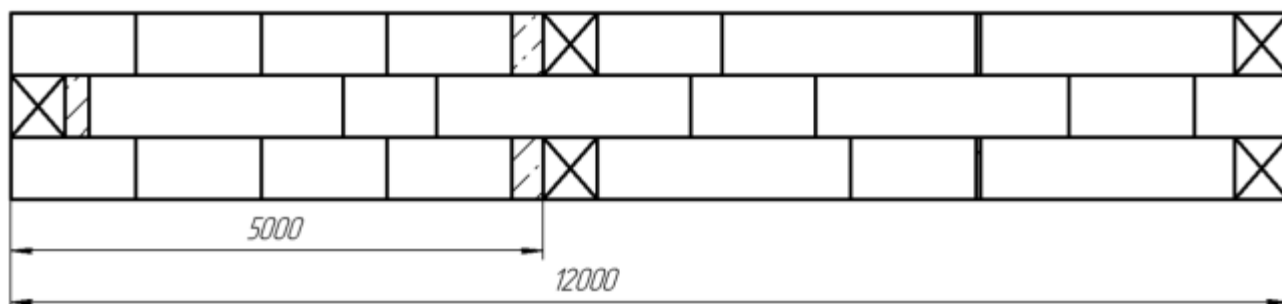


Рисунок 34 – Задание конфигурации фундамента в разработанном приложении

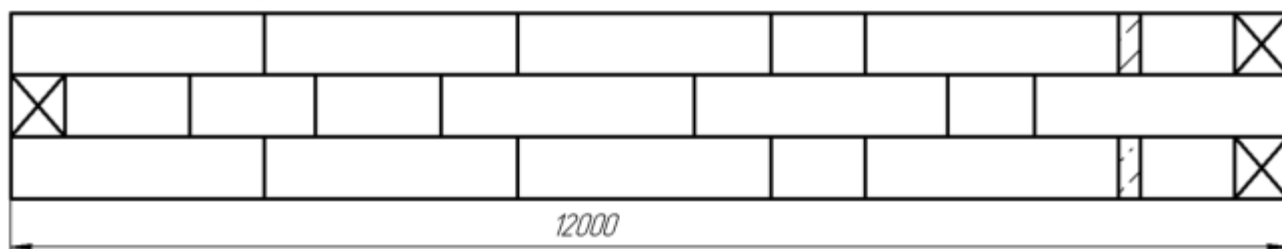
Следует отметить, что при каждом новом поиске решения программой будут предлагаться различные варианты раскладки. Это связано с тем что генетический алгоритм является стохастическим методом. Предложенный программой вариант раскладки блоков представлен на рисунке 35.

Предложенный вариант раскладки блоков получил положительную оценку у специалистов строительной предприятия Волжское ЗАО «Гидроспецстрой» по заказу которых и было разработано данное приложение.

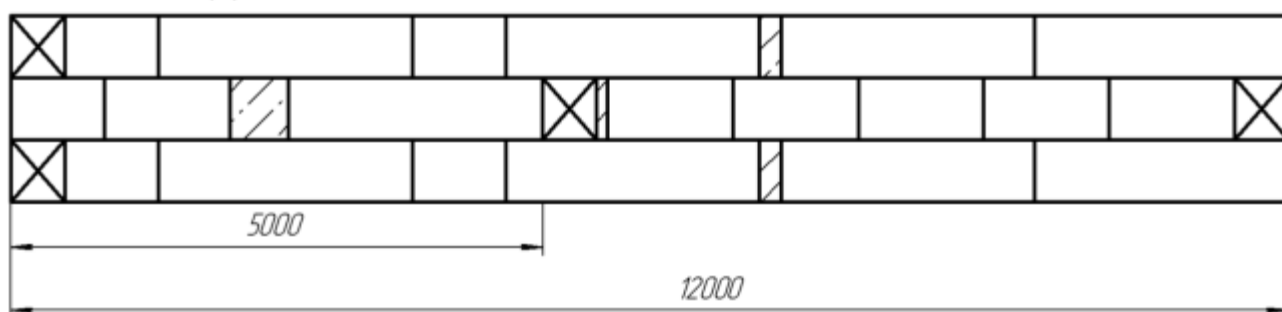
Северная стена (N):



Восточная и центральная стена (E, C):



Южная стена (S):



Западная стена (W):

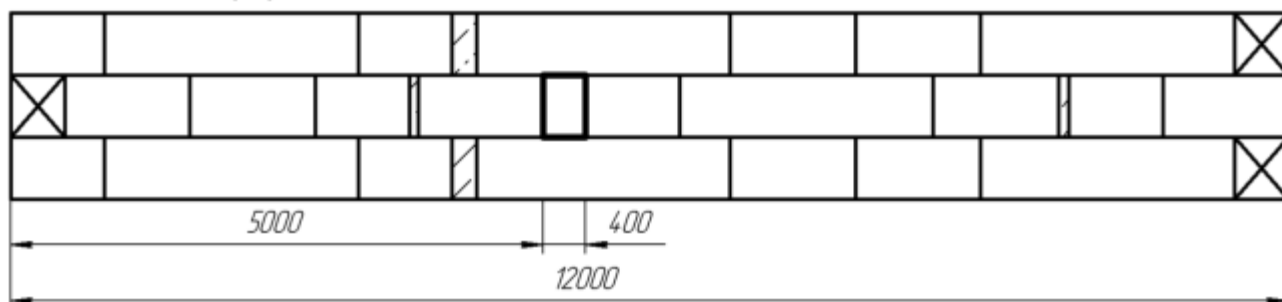


Рисунок 35 – Вариант раскладки блоков предложенных программой
(штриховкой обозначены места требующие ручной заливки бетона, крестами
обозначены торцы ФБС блоков)

Заключение

1. На основе анализа состояния вопроса установлено, что автоматизация решения технических задач в строительстве возможна за счет применения в них алгоритмов искусственного интеллекта.

2. Показано, что моделирование сборно-ленточных фундаментов можно рассматривать как задачу оптимизацию. Предложен вариант формулирования для этой задачи целевой функции, управляющих параметров и ограничений оптимизационной задачи.

3. Разработаны подходы по применению принципов генетического алгоритма для решения задачи определения оптимальной раскладки ФБС блоков при решении задач проектирования домов. При этом дана собственная интерпретация операторов генетического алгоритма с привязкой к решаемой задаче поиска оптимальной раскладки блоков.

5. Опытным путем определены параметры генетического алгоритма, позволяющие успешно решать задачу моделирования сборно-ленточных фундаментов и поиска оптимальной раскладки блоков.

4. В среде разработки Eclipse на языке программирования Java разработано программное обеспечение, на практике реализующее предложенные алгоритмические решения.

5. Экспериментально подтверждена состоятельность предложенных принципов путем их апробации на реальных данных, взятых из эскизных проектов частных домов.

6. Разработанный программный продукт планируется к применению на предприятии Волжское ЗАО «Гидроспецстрой» для решения производственных задач.

Список используемой литературы

1. Jacobson, L. Genetic Algorithms in Java Basics / L. Jacobson, B. Kanber. – Apress, 2015. – 172 p.
2. Sivanandam, S.N. Introduction to Genetic Algorithms / S.N. Sivanandam, S.N. Deepa. - Springer Berlin Heidelberg, 2008. – 442 p.
3. Chen, Y.P. Extending the Scalability of Linkage Learning Genetic Algorithms / Ying-ping Chen. - Springer Berlin Heidelberg, 2006. – 106 p.
4. Панченко, Т.В. Генетические алгоритмы : учебно-методическое пособие / Т.В. Панченко. – Астрахань : Издательский дом «Астраханский университет», 2007. – 87 с.
5. Гладков, Л.А. Генетические алгоритмы / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. – М.: Физматлит, 2006. – 320 с.
6. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М.: Издательство «Горячая Линия - Телеком», 2007. – 452 с.
7. Maulik, U. Multiobjective Genetic Algorithms for Clustering / U. Maulik, S. Bandyopadhyay, A. Mukhopadhyay. – Springer Berlin Heidelberg, 2011. – 281 p.
8. Rothlauf, F. Representations for Genetic and Evolutionary Algorithms / Franz Rothlauf. – Springer Berlin Heidelberg, 2006. – 325 p.
9. Luque, G. Parallel Genetic Algorithms: Theory and Real World Applications / G. Luque, E. Alba. – Springer Berlin Heidelberg, 2011. – 367 p.
10. Dawid, H. Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economic Models / Herbert Dawid. - Springer Berlin Heidelberg, 1999. – 200 p.
11. Pearson, D.W. Artificial Neural Nets and Genetic Algorithms / D.W. Pearson, N.C. Steele, R.F. Albrecht. - Springer Vienna, 2003. – 266 p.
12. Chen, Y. Extending the Scalability of Linkage Learning Genetic Algorithms / Ying-ping Chen. - Springer Berlin Heidelberg, 2006. – 108 p.

13. Cantú-Paz, E. Efficient and Accurate Parallel Genetic Algorithms / Erick Cantú-Paz. - Springer US, 2001. – 162 p.
14. Reeves, C. Genetic Algorithms—Principles and Perspectives / C.R. Reeves, J.E. Rowe. - Springer US, 2002. – 332 p.
15. Gen, M. Network Models and Optimization / M. Gen, R. Cheng, L. Lin. - Springer London, 2008. – 692 p.
16. Хайкин С. Нейронные сети. Полный курс. 2-е издание / С. Хайкин; пер. с англ. Н.Н. Куссуль и А.Ю. Шелестова, под ред. Н.Н. Куссуль. – 2-е изд. – М.: «Вильямс», 2006. – 1104 с.
17. Koprinkova-Hristova, P. Artificial Neural Networks / Petia Koprinkova-Hristova, Valeri Mladenov, Nikola K. Kasabov. – Springer International Publishing, 2015. – 488 p.
18. Li, S.Z. Hamming Distance / S.Z. Li, A. Jain. – Encyclopedia of Biometrics, 2009. – 668 p.
19. Rigatos, G. G. Advanced Models of Neural Networks / Gerasimos G. Rigatos. – Springer Berlin Heidelberg, 2015. – 396 p.
20. Galushkin, A. I. Neural Networks Theory / Alexander I. Galushkin. – Springer Berlin Heidelberg, 2007. – 396 p.

UML диаграммы классов

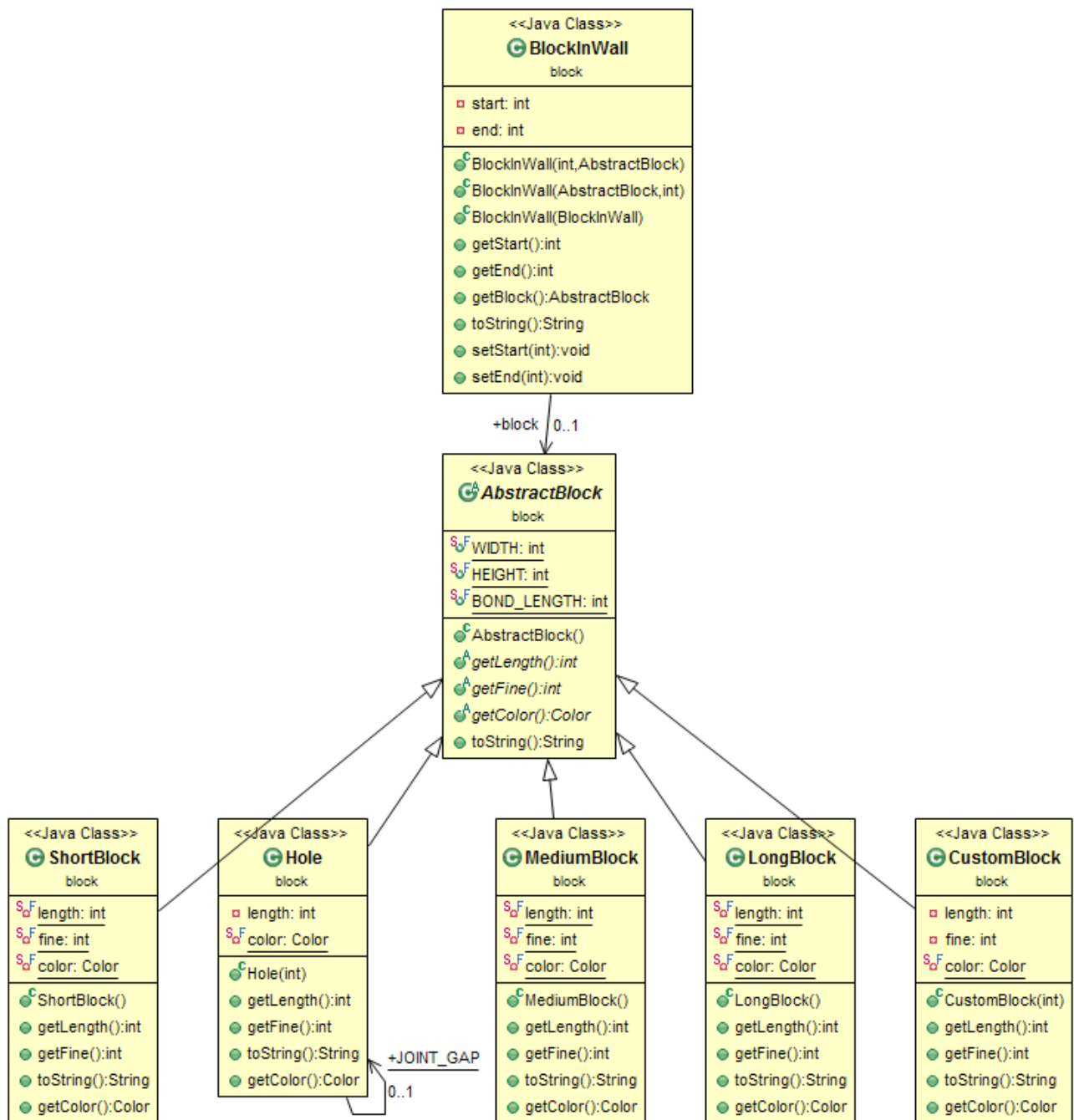


Рисунок А.1 – UML диаграмма классов в пространстве имен block проекта

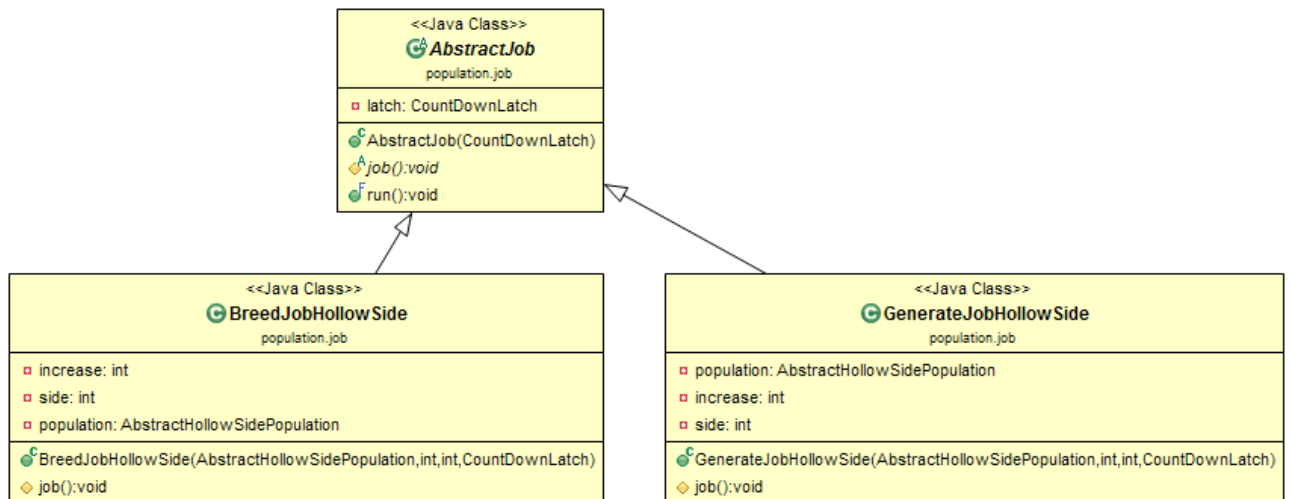


Рисунок А.2 – UML диаграмма классов в пространстве имен population.job проекта

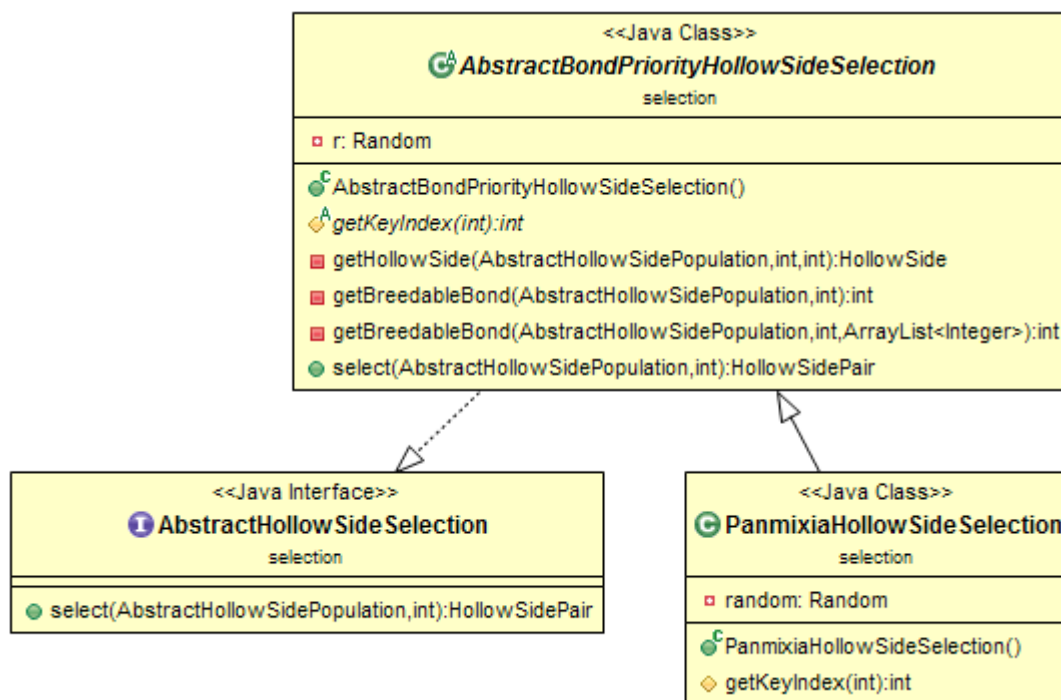


Рисунок А.3 – UML диаграмма классов в пространстве имен selection проекта

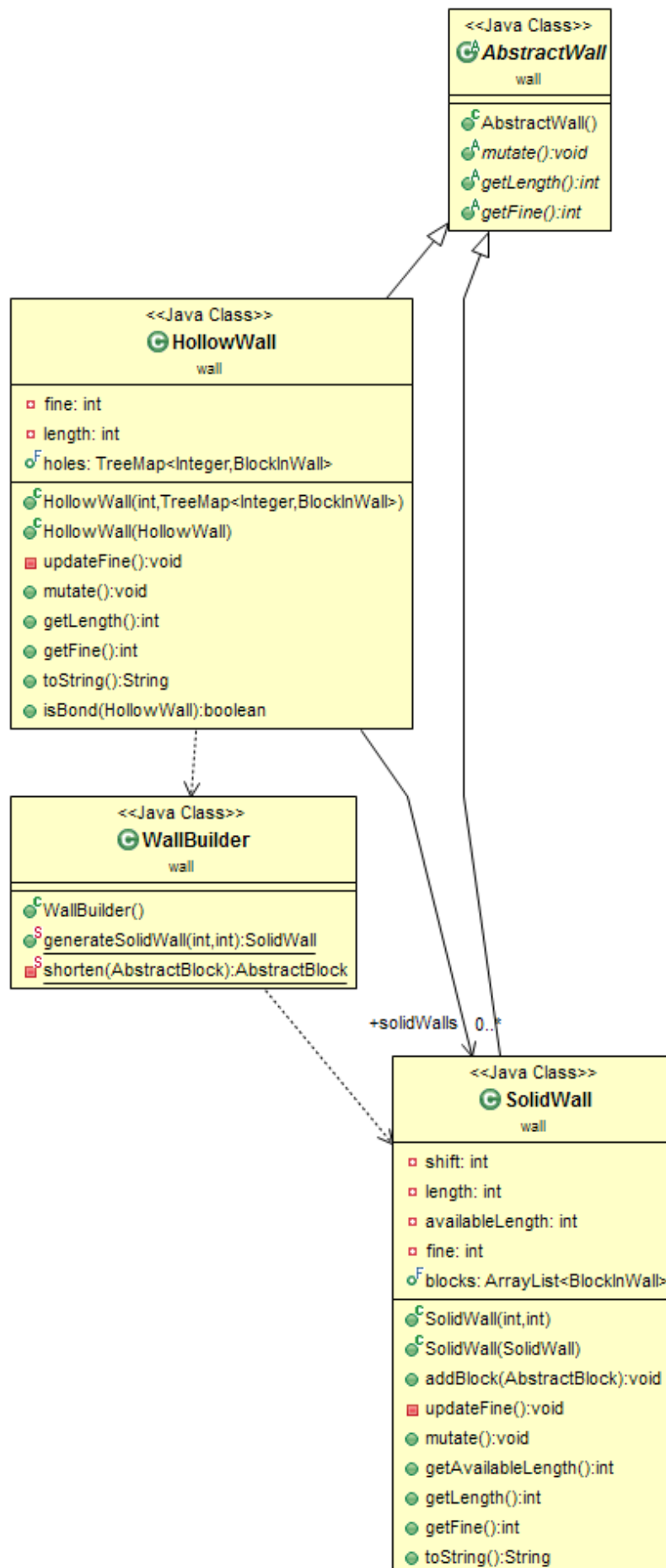


Рисунок А.4 – UML диаграмма классов в пространстве имен wall проекта

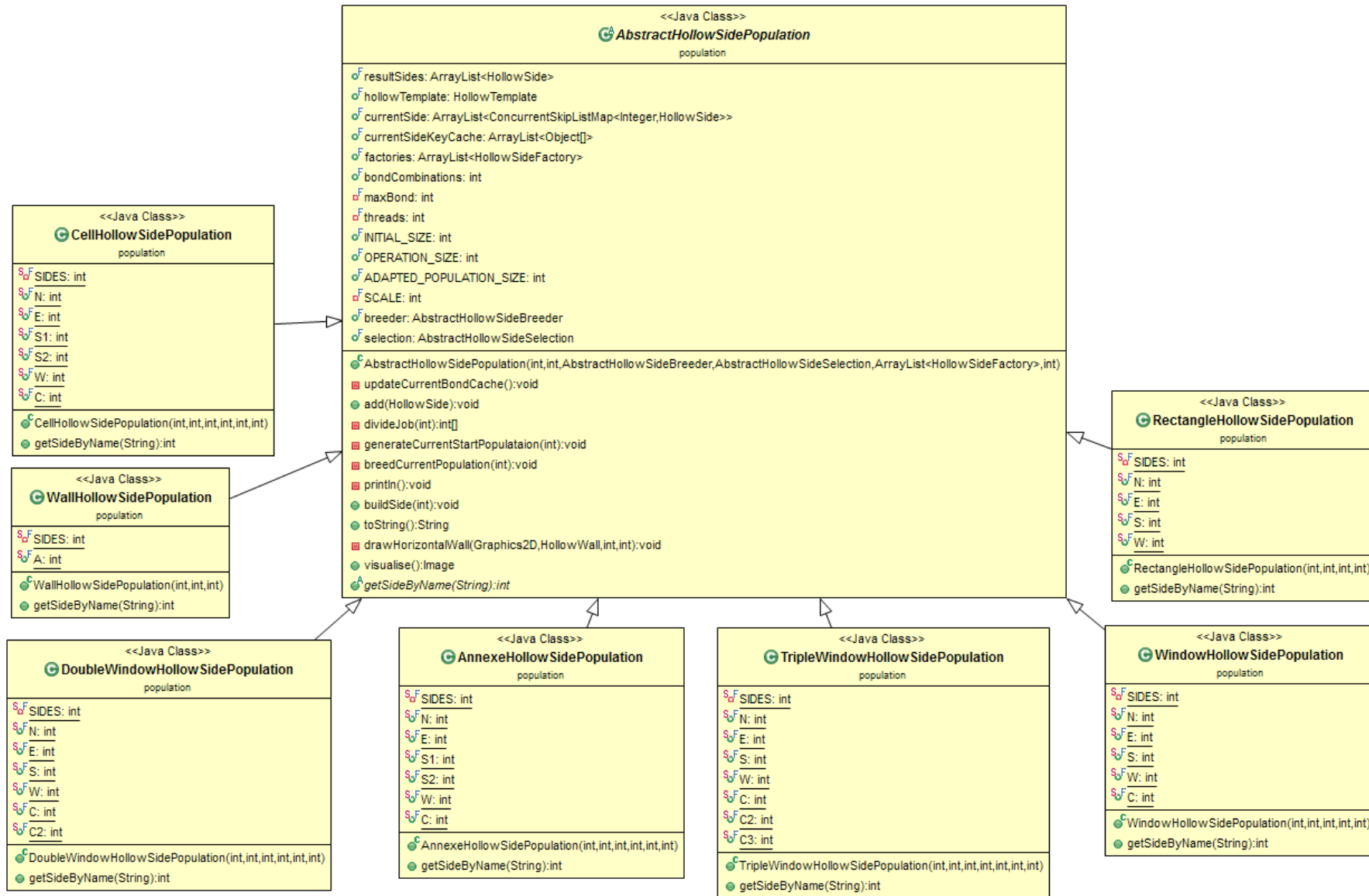


Рисунок А.5 – UML диаграмма классов в пространстве имен population проекта

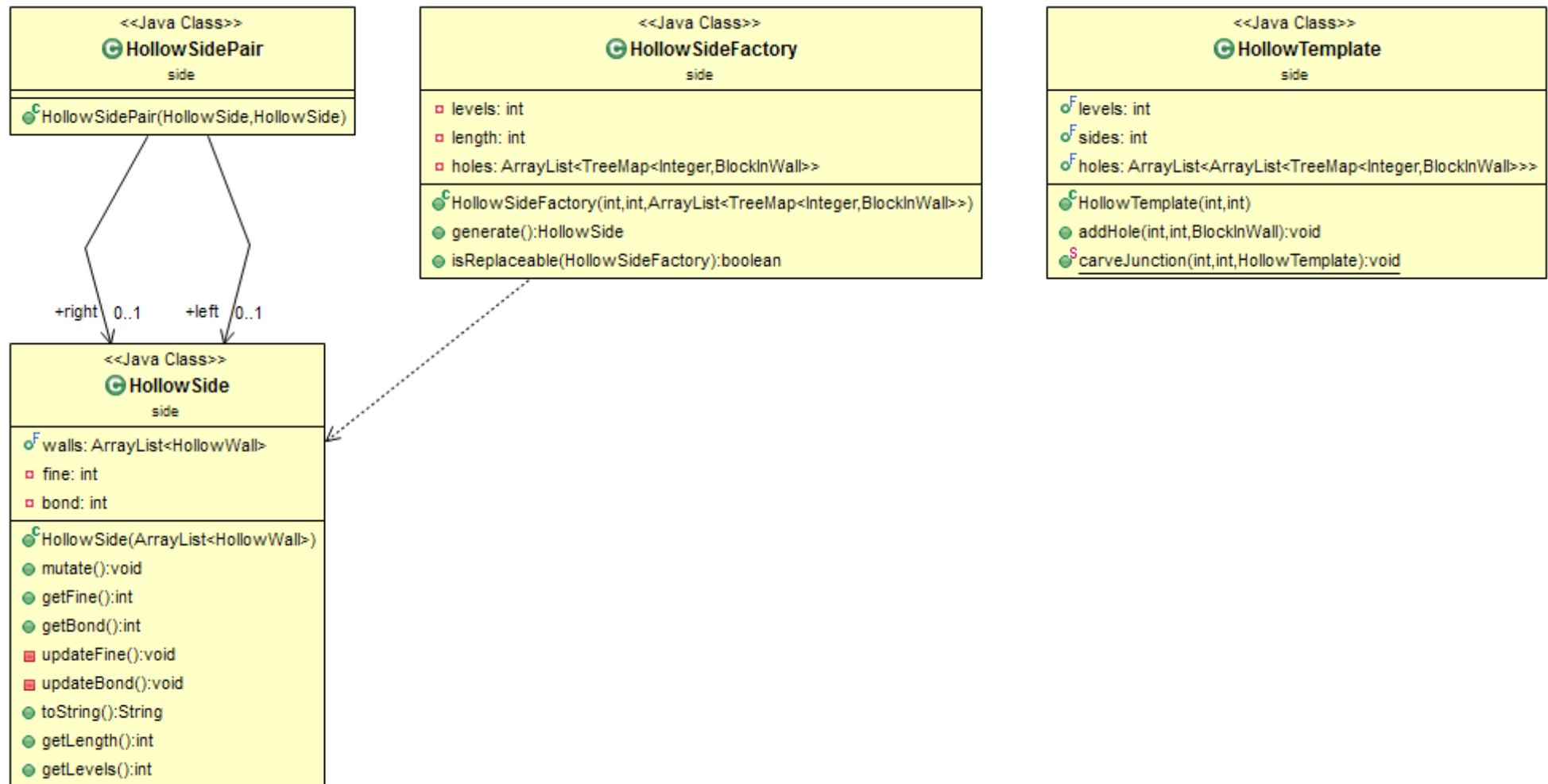


Рисунок А.6 – UML диаграмма классов в пространстве имен side проекта