УДК 519.688:004.8

Программная реализация иммунного алгоритма, применяемого для поиска проектных решений в электронном архиве проектной документации

Буткина А. А., Гущина О. А., Кузнецова О. А., Шамаев А. В.

Национальный исследовательский
Мордовский государственный университет им. Н.П. Огарева

Аннотация: В статье представлена разработка программного обеспечения, используемого для поиска проектных решений, хранящихся в электронном архиве проектной документации. Поиск осуществляется с помощью модифицированной версии иммунного алгоритма как по заданным характеристикам проектных решений, так и по характеристикам документов (текстовых и графических). Представлены диаграммы вариантов использования и классов, схема организации базы данных, стек средств и технологий разработки, результаты проверки работоспособности и эффективности разработанного программного обеспечения.

Ключевые слова: программное обеспечение, иммунный алгоритм, алгоритм клональной селекции, проектное решение, электронный архив, поиск.

На сегодняшний день большинство IT-компаний располагают хранилищами цифровых файлов, содержащими информацию о выполненных проектах и, как правило, имеющими значительный объем. Чаще всего реализация нового проекта или кейса основывается на принципах и методах, применяемых ранее в предыдущих работах. Использование готовых разработок компании позволяет сократить время выполнения нового пакета работ и избежать допущенных ранее ошибок и недочётов, обеспечив тем самым повышение качества создаваемого продукта.

В настоящее время на практике применяются разнообразные программные системы поиска необходимых сведений, содержащихся в электронных архивах проектной документации. В таких программах достаточно эффективно выполняется обнаружение искомой текстовой информации, однако зачастую не удается отыскать документы, содержащие данные, представленные в графическом формате. В IT компаниях это могут быть файлы презентаций, UML диаграммы, макеты интерфейса приложений, созданные в различных графических редакторах, и т. п. Таким образом, существует потребность в программном обеспечении, которое будет производить поиск документов любого типа, составляющих проектное решение, вне зависимости от формата хранящейся в них информации, что обуславливает актуальность данного исследования.

Целью данной работы является разработка программного обеспечения, позволяющего осуществлять поиск проектных решений, размещенных в электронном архиве проектной документации и представленных как в текстовом, так и в графическом форматах, с помощью иммунного алгоритма. Для достижения этой цели были поставлены и решены следующие задачи:

– изучение предметной области и постановка задачи исследования;

– анализ существующих аналогов разрабатываемого программного обеспечения и выявление особенностей их функционирования;

– выбор средств и технологий разработки;

– определение функциональных требований к разрабатываемому программному обеспечению;

– разработка архитектуры программного обеспечения;

– реализация и проверка работоспособности разработанного программного обеспечения.

Прежде всего, опишем постановку задачи исследования. Каждому проектному решению, хранящемуся в электронном архиве проектной документации, должна быть присвоена определённая бинарная последовательность, представляющая собой набор характеристик, соответствующих данному проектному решению. Указанная бинарная последовательность формируется следующим образом: в позицию, соответствующую определенной характеристике, записывается значение «1», если эта характеристика присуща данному проектному решению, и значение «0» – в противном случае.

Аналогично каждому файлу документа, входящего в состав каждого проектного решения, ставится в соответствие бинарная последовательность, отражающая набор характеристик данного документа и формируемая указанным выше способом.

Для реализации поиска проектного решения или документов, содержащихся в базе данных проектных решений, отвечающих заданному пользователем набору характеристик, должна быть сформирована соответствующая бинарная последовательность. Все бинарные последовательности хранящихся в архиве проектных решений и проектного решения, для которого выполняется поиск, имеют одинаковую размерность. Размерности бинарных последовательностей, отражающих характеристики искомого документа и документов, входящих в существующие проектные решения, также совпадают.

Таким образом, задачу исследования можно свести к нахождению заданной бинарной последовательности, соответствующей искомому проектному решению или документу, обладающему требуемым набором значений характеристик, в имеющейся базе бинарных последовательностей, построенных для хранящихся в базе данных проектных решений.

Эта задача относится к задачам линейного целочисленного программирования. Следует также учитывать, что целевая функция данной задачи является мультимодальной, поскольку в процессе поиска могут быть найдены несколько проектных решений (документов), отвечающих заданному пользователю набору характеристик.

Указанная особенность целевой функции приводит к тому, что задачи оптимизации таких функций плохо поддаются решению традиционными методами (метод ветвей и границ, метод градиентного спуска и др.), поэтому возникает необходимость в использовании альтернативных методов, среди которых следует выделить иммунный алгоритм клональной селекции, обеспечивающий параллельный поиск оптимального решения. Данный алгоритм наилучшим образом подходит для решения поставленной задачи, поскольку строит своё решение на основе перебора многочисленных вариантов последовательностей и отбора лучших из них, а также предоставляет более адаптивные механизмы, чем другие иммунные алгоритмы.

В настоящее время при работе с электронным архивом проектной документации применяется целый ряд специализированных программ. Наибольший интерес для данного исследования представляют программы YuKoSoft, E-Arch и Search. Далее рассмотрим каждую из них более подробно с точки зрения осуществления поиска проектных решений по заданным характеристикам.

1) YuKoSoft – созданная российскими разработчиками программа [1], используемая для ведения архива документов. Она включает систему электронного докумен-

тооборота, электронный архив документов с возможностью регистрации документов организации путем автоматического формирования и присвоения им уникальных номеров. Также в данной программе предусмотрена возможность поиска проектов и документов, который, однако, выполняется недостаточно корректно, в частности:

– в результатах поиска не учитывается дата создания документа;

– невозможно выполнить поиск отдельных файлов, входящих в состав проектного решения;

– невозможно производить поиск документов по типу файла;

– невозможно выполнить поиск графического файла по типу содержащейся в нем информации.

2) E-Arch – это защищённое структурированное централизованное хранилище электронных документов, предназначенное для организации оперативного многопользовательского доступа к содержащейся в них информации. Данная программа разработана российской компанией Редокс [2]. Основными выявленными недостатками данной программы являются:

– официальный сайт не предоставляет доступ для скачивания и установки бесплатной лицензионной версии;

– отсутствует возможность поиска проектного решения по нескольким файлам;

– отсутствует возможность поиска графических файлов по типу содержащейся в них информации.

3) Search – разработанная группой российских программистов программа, представляющая собой систему для работы с архивом технической документации [3]. Программа предоставляет возможность поиска по некоторым параметрам проекта, есть возможность просмотра файлов, входящих в состав проекта. Однако, в результате анализа её возможностей были обнаружены следующие недостатки:

– программа не имеет бесплатной лицензионной версии, доступной для использования;

– отсутствует возможность одновременного поиска нескольких файлов, входящих в проектное решение;

– невозможно выполнить поиск графических файлов по типу документа и внутренним параметрам (например, наличие видео в презентации).

В таблице 1 представлены результаты сравнительного анализа рассмотренных аналогов и разработанного программного обеспечения FindProject.

Как было описано ранее, каждое проектное решение (и каждый отдельный документ в его составе) имеет набор характеристик, значения которых должны быть заданы при его внесении в базу данных. Будем считать каждый добавленный в электронный архив набор характеристик проектного решения (или документа) – антителом, а искомое проектное решение (или документ), обладающее набором характеристик, по которым происходит поиск, – антигеном.

Обозначим через x некую сформированную пользователем совокупность проектных решений. В работе будут рассматриваться два вида аффинностей (термодинамическая характеристика, количественно описывающая силу взаимодействия веществ [4]):

– аффинность ax_v – выражает степень схожести антитела с антителом, т. е. одного архивного проектного решения (документа) с другим архивным проектным решением (документом);

– аффинность $ay_{v,w}$ – выражает степень схожести антитела с антигеном, т. е. архивного проектного решения (документа) с искомым проектным решением (доку-

Таблица 1. Сравнение систем

	YuKoSoft	E-Arch	Search	FindProject
Поиск проектного решения по нескольким документам	–	–	–	+
Поиск графических файлов по их характеристикам	–	–	–	+
Возможность привязывать документы к проектам	–	–	+	+
Разграниченный доступ к проектам	+	+	+	+
Разграниченный доступ к документам	–	+	+	+
Наличие бесплатной версии	+	–	–	+

ментом).

Для хранения бинарных последовательностей, соответствующих проектным решениям (документам), которые не отвечают критериям поиска и не будут рассматриваться, т. е. будут удалены, используется массив c_v.

Весь массив имеющихся в архиве бинарных последовательностей, участвующий в отборе решения, отвечающего критериям поиска, отражается величиной e_v с указанием того, какие претенденты будут удалены и какие останутся.

Таким образом, иммунный алгоритм параллельного поиска, основанный на клональном алгоритме, для решаемой задачи можно представить в следующем виде [5]:

Шаг 1. *Распознавание антигена.* На данном шаге происходит формирование антигена, т. е. бинарной последовательности, составленной по заданным пользователем значениям характеристикам проектного решения (документа).

Шаг 2. *Выработка антител.* На данном шаге происходит «вспоминание» – формирование массива бинарных последовательностей, отражающих характеристики проектных решений (документов), путем их извлечения из базы данных (архива проектных решений).

Шаг 3. *Вычисление аффинностей.* На данном этапе происходит выявление набора лимфоцитов, которые индуцируют наиболее соответствующие им антитела – проектные решения (документы). Формулы для вычисления аффинностей представлены ниже:

а) аффинность пары антител (пары архивных проектных решений (документов)):

$$ay_{v,w} = \frac{\sum\limits_{i=1}^{M} S_i}{\sum\limits_{j=1}^{M}\sum\limits_{i=1}^{S} p_{ij}}, \tag{1}$$

где M – количество бит в бинарной последовательности, S – количество архивных проектных решений (антител), S_i – вероятность равенства i-го бита (отдельной характеристики архивного проектного решения) текущего антигена единице; p_{ij} – ве-

роятность равенства i-го бита текущего антитела i-му биту j-го антитела;

б) аффинность антитела (архивного проектного решения (документа)) к антигену (искомому проектному решению (документу)):

$$ax_v = opt_v, \tag{2}$$

где opt_v – сила связи (соответствия, схожести) антигена (искомого проектного решения (документа)) с антителом (архивным проектным решением (документом)). Она равна количеству единиц в бинарной последовательности, полученной в результате логического умножения бинарной последовательности архивного проектного решения (документа) на бинарную последовательность искомого проектного решения (документа).

Шаг 4. *Разделение лимфоцитов.* На данном шаге происходит исключение из обработки не подходящих B-лимфоцитов. Решения, которые не будут рассматриваться, рассчитываются по формуле:

$$c_v = \frac{1}{N}\sum_{w=1}^{N} ac_{v,w} > T_c,$$

где

$$ac_{v,w} = \begin{cases} 1, \text{ если } ayv, w \geq Tac1; \\ 0, \text{ иначе } ayv, w < Tac1; \end{cases} \tag{3}$$

T_c – пороговое значение для B-лимфоцита; T_{ac1} – пороговое значение для антигена.

Шаг 5. *Размножение и подавление антител.* Ожидаемый масштаб e_v выработки антител рассчитывается по формуле:

$$e_v = \frac{ax_v \prod_{s=1}^{S}(1 - as_{v,s})}{c_v \sum_{i=1}^{N} ax_i},$$

где

$$as_{v,w} = \begin{cases} ay_{v,s}, \text{ если } ay_{v,s} \leq T_{ac2}; \\ 0, \text{ иначе } ay_{v,s} < T_{ac2}. \end{cases} \tag{4}$$

В популяции лимфоцитов разнообразие и концентрация антител архивных проектных решений (документов) также подчиняется отношению (4). Обширность направлений поиска определяется с учётом локальных экстремумов (4).

Шаг 6. *Размножение антител.* Для обработки бинарной последовательности искомого проектного решения (документа) происходит создание лимфоцитов, замещающих бинарные последовательности архивных проектных решений (документов) – антител. Это позволяет обеспечить разнообразие антител, достигаемое согласно теории иммунитета путем реализации процесса мутаций (скрещивания), что обеспечивает наибольшую эффективность алгоритма за счёт максимально широкого охвата вариантов решения.

Для разработки программного обеспечения были выбраны следующие средства:

– полнофункциональная и расширяемая интегрированная среда разработки Visual Studio Community;

– компилируемый, статически типизируемый язык программирования C++, поддерживающий объектно ориентированную, обобщённую и процедурную парадигмы программирования;

– интерфейс прикладного программирования Windows Forms для разработки графического интерфейса;

– платформа разработки .NET Framework взаимодействующая с операционными системами семейства Microsoft Windows и имеющая общеязыковую среду исполнения;

– надёжная, качественная и бесплатно распространяемая реляционная система управления базами данных MySQL.

Приведем описание функциональных требований к разработанному программному обеспечению в виде диаграммы вариантов использования (рис. 1). Стоит отметить, что данным программным обеспечением может пользоваться только зарегистрированный пользователь. Для регистрации ему необходимо обратиться к администратору.

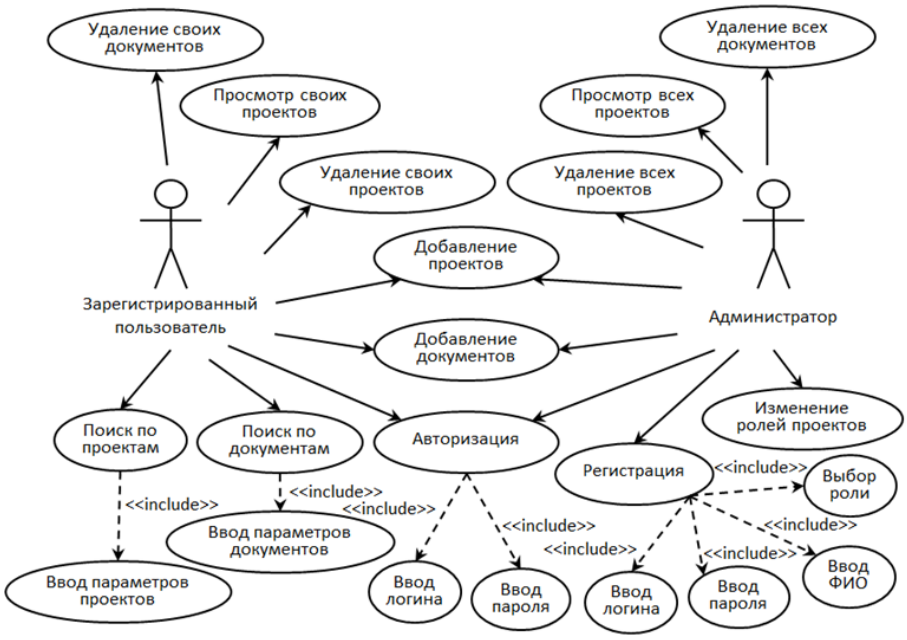

Рис. 1. Диаграмма вариантов использования

На основе сформулированных функциональных требований к программному обеспечению и применения объектно-ориентированного подхода к реализации описанного выше иммунного алгоритма была разработана логическая модель программного обеспечения, представленная в виде диаграммы классов (рис. 2).

Таким образом, структурно разработанное программное обеспечение включает три основных класса: Mainform, IaController и Antibody.

Класс MainForm обеспечивает:

– связь с пользователем (считывает заданные параметры поиска, выдаёт его конечный результат);

– подключение к имеющейся базе данных проектных решений и документов электронного архива;

– работу остальных классов и модулей разработанного программного обеспече-

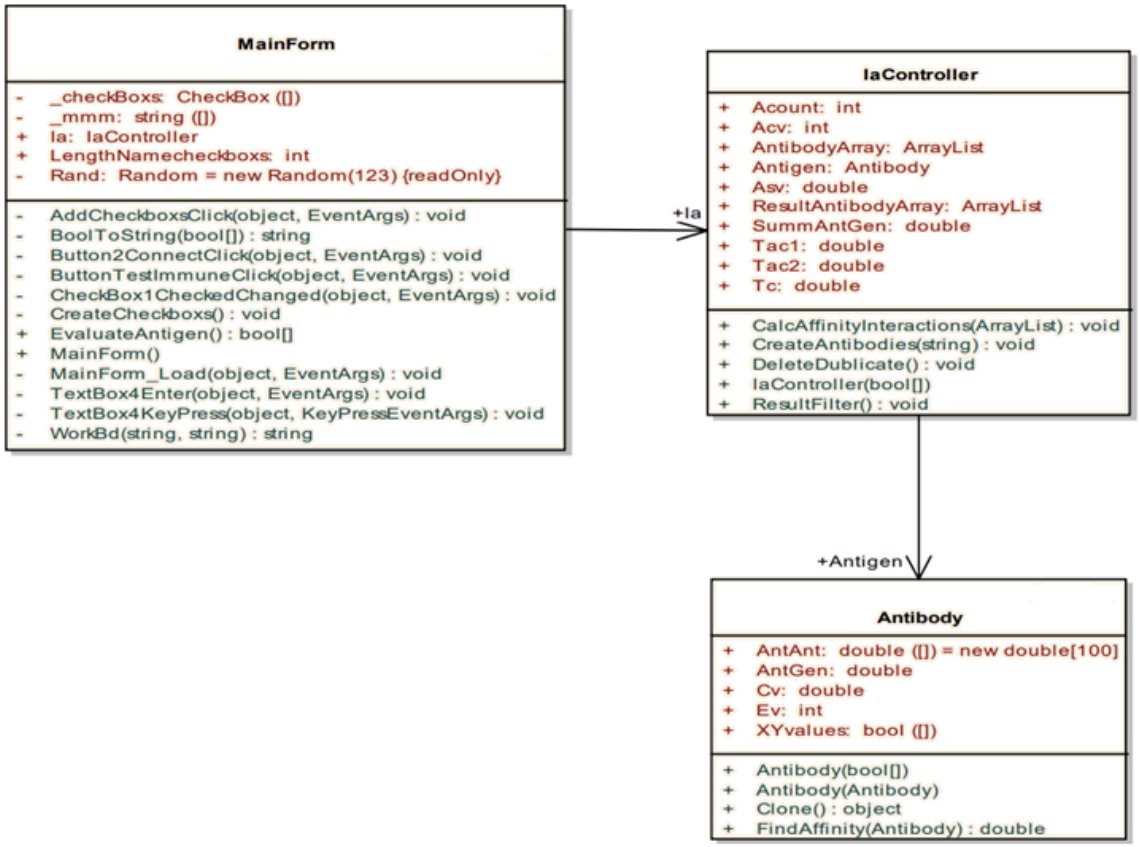

Рис. 2. Диаграмма классов

ния.

Класс Antibody описывает свойства (характеристики) антитела, а также выполняет вычисление различных видов аффинностей с помощью модуля FindAfinity.

В свою очередь модуль FindAfinity реализует вычисление аффинностей как между каждой парой антител, соответствующей архивным проектным решениям (документам), так и между каждой парой антитела с антигеном, соответствующей архивному проектному решению (документу) и искомому проектному решению (документу).

Класс IaController реализует алгоритм параллельного поиска оптимального решения и необходимые для его реализации вспомогательные методы. Данный класс содержит методы:

– CalcAffinityIterations реализует алгоритм параллельного поиска проектного решения, соответствующего заданным критериям;

– DeleteDublicate выполняет исключение одинаковых антител, которые образовались на шаге клонирования;

– ResultFilter необходим для объединения бинарных последовательностей, соответствующих отдельным проектным решениям (документам), полученным с применением параллельного поиска, в единый массив.

Для проверки работоспособности разработанного программного обеспечения была спроектирована база данных, структура которой представлена на рис. 3.

Она содержит 9 таблиц:

– login_password – для хранения логина и пароля пользователя;

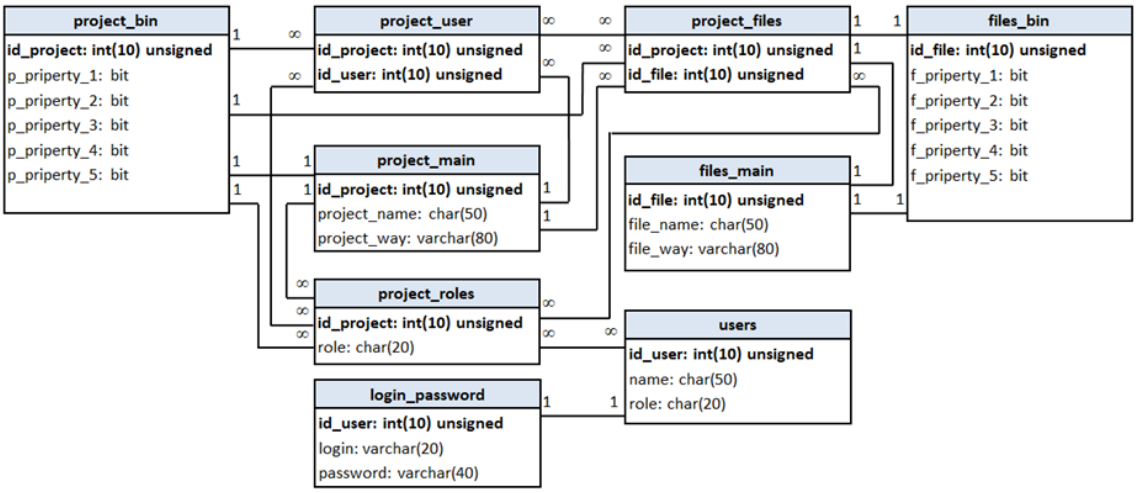

Рис. 3. Структура базы данных

– users – для хранения характеристик пользователя;
– project_roles – для соотнесения проектных решений и ролей;
– project_bin – для хранения бинарных характеристик проектных решений;
– project_user – для соотнесения проектных решений и пользователей;
– project_main – для хранения основных характеристик проектного решения;
– project_files – для соотнесения проектных решений и документов;
– files_main – для хранения основных характеристик документов;
– files_bin – для хранения бинарных характеристик документов.

Для использования разработанного приложения каждый пользователь должен войти в систему. Для входа необходимо ввести корректные логин и пароль и нажать кнопку «Войти». Администратор зарегистрирован в системе по умолчанию. Для того, чтобы обычный незарегистрированный пользователь получил возможность входа в систему по своему логину и паролю он должен обратиться к администратору.

После входа в систему открывается окно «Главное меню». Для администратора открывается главное меню с пометкой «(Администратор)» (рис. 4, слева), в котором он может перейти к регистрации новых пользователей или к настройке проектов. В главном меню обычного пользователя можно выполнить поиск проектных решений или документов, а также просмотреть свои проектные решения (рис. 4, справа).

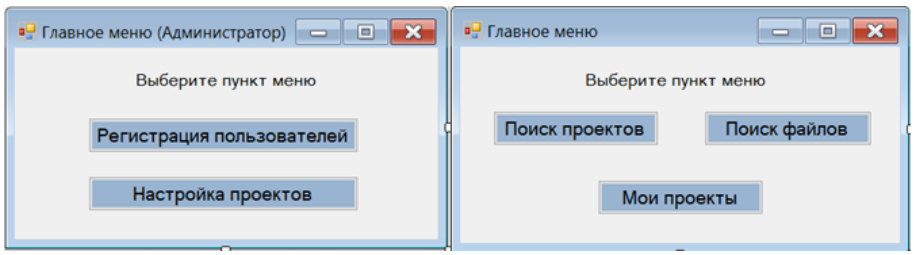

Рис. 4. Главное меню Администратора и обычного пользователя

Выбрав пункт главного меню «Регистрация пользователей», администратор может зарегистрировать нового пользователя. Для этого необходимо корректно запол-

нить поля «ФИО», «Логин», «Пароль» и выбрать роль для разграничения доступа к проектным решениям и документам. Выбрав в главном меню «Настройка проектов» (рис. 5), администратор может просмотреть роли, для которых доступен каждый проект, его расположение, а также просмотреть название и расположение каждого файла проектного решения. Кроме того, он может назначить роли на каждый проект, удалить или добавить проект, удалить или добавить файл.

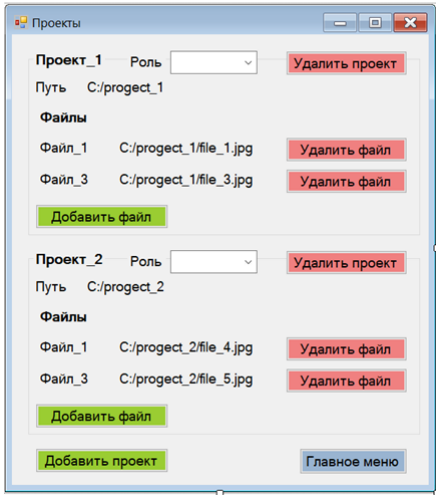

Рис. 5. Диалоговое окно «Настройка проектов»

Обычный пользователь может просматривать только свои проекты и не может назначать роли на проекты. Кроме того он может удалять только файлы документов и проекты, которые добавил он сам. Также пользователь может выполнить поиск проектных решений и файлов, задав соответствующие параметры поиска.

Теперь продемонстрируем проверку работоспособности разработанного программного обеспечения, для чего рассмотрим несколько сценариев действий пользователя.

Сценарий 1. Поиск проектного решения по заданным параметрам проектного решения.

Шаг 1. Ввод логина и пароля пользователя.

Шаг 2. Выбор пункта главного меню «Поиск проектов» в диалоговом окне, представленном на рис. 4.

Шаг 3. Ввод параметров поиска проекта в форму, представленную на рис. 6.

Шаг 4. Получение результатов поиска проектов по заданным параметрам (рис. 7).

Сценарий 2. Поиск проектного решения по заданным параметрам файла документа.

Шаг 1. Ввод логина и пароля пользователя.

Шаг 2. Выбор пункта главного меню «Поиск файлов» в диалоговом окне, представленном на рис. 4.

Шаг 3. Ввод параметров поиска файла в форму, представленную на рис. 8.

Шаг 4. Получение результатов поиска файлов, входящих в состав проектных решений, по их параметрам (рис. 9).

Оценка эффективности поисковой системы производилась по следующей методике построения паретовой границы в критериальном пространстве показателей эффективности поиска [6]. При этом были вычислены значения двух ключевых по-

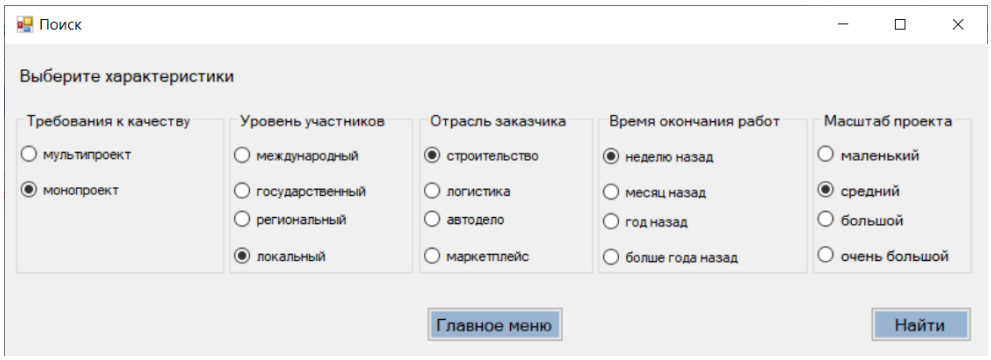

Рис. 6. Диалоговое окно «Поиск проектов»

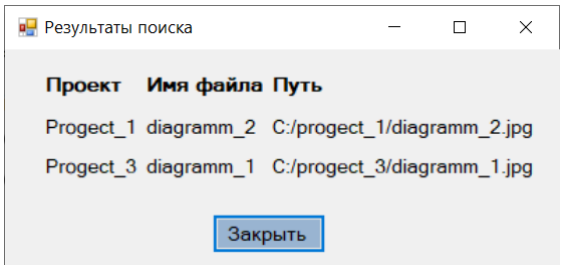

Рис. 7. Результат поиска проекта по заданным параметрам

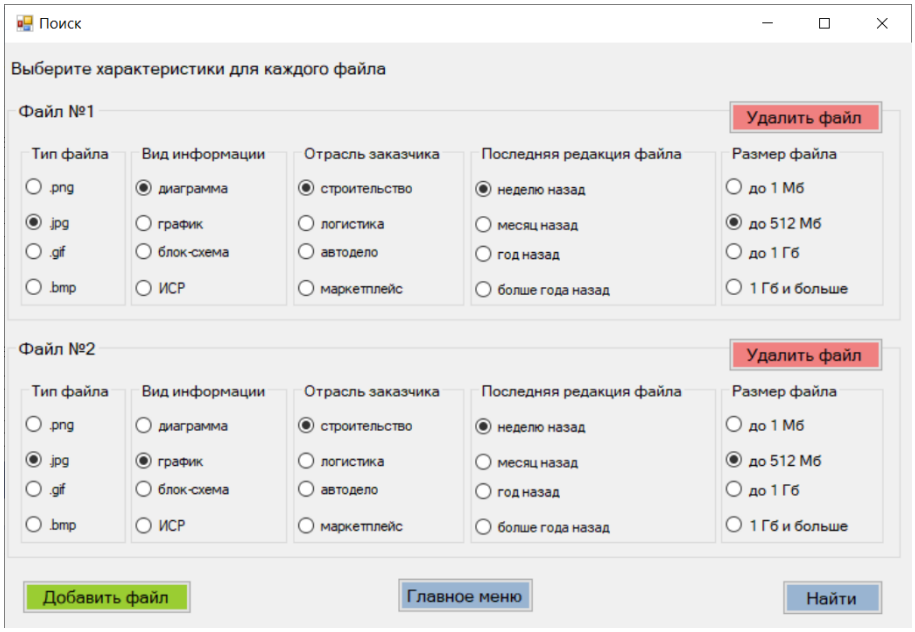

Рис. 8. Диалоговое окно «Поиск файлов»

казателей: коэффициента полноты $k_{\text{П}}$ и коэффициента точности $k_{\text{Т}}$ по следующим

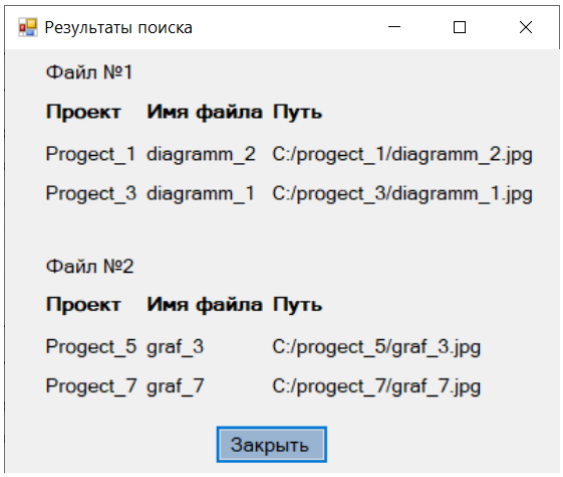

Рис. 9. Результат поиска файлов по заданным параметрам

формулам:

$$k_{\text{П}} = \frac{\sum\limits_{(d,s)\in M\times S} \eta\,(d,s)\,\varphi\,(d,s)}{\sum\limits_{(d,s)\in M\times S} \eta\,(d,s)}, \tag{5}$$

$$k_{\text{П}} = \frac{\sum\limits_{(d,s)\in M\times S} \eta\,(d,s)\,\varphi\,(d,s)}{\sum\limits_{(d,s)\in M\times S} \varphi\,(d,s)}, \tag{6}$$

где s – запрос, d – проектное решение, пара (d,s) называется релевантной парой, если проектное решение d релевантно запросу s, M – массив всех проектных решений, S – массив всех запросов, m – максимальное количество запросов, $\eta\,(d,s) = \frac{1}{m}\sum\limits_{j=1}^{m} \eta_j\,(d,s)$ — степень релевантности (d,s), $\varphi(d,s)$ – отношение выдачи (d,s):

$$\begin{cases} \varphi(d,s) = 1, \text{ если найден документ } d \text{ на запрос } s; \\ 0, \text{ в противном случае.} \end{cases}$$

Коэффициент полноты равен 1 только в том случае, когда все проектные решения, степень релевантности которых выше 0, входят в массив результатов поиска. Коэффициент точности будет единичным, если в массив результатов поиска входят только проектные решения со степенью релевантности, равной 1.

Изменяя состав выдачи, то есть функцию $\varphi(d,s)$, можно получать различные значения показателей полноты и точности поиска. Нахождением для каждого значения полноты наибольшего значения точности, строится паретова граница для множества значений полноты-точности поиска.

На основе проведенных экспериментов была построена кривая оптимальных характеристик (рис. 10).

По построенной кривой можно определить границы максимально достижимых значений полноты и точности для данной предметной области.

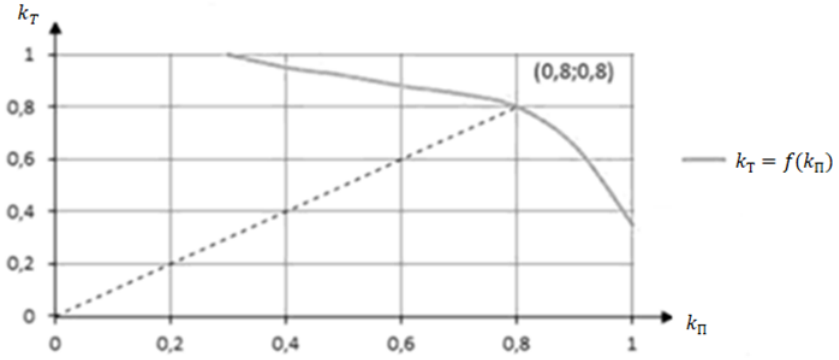

Рис. 10. Кривая оптимальных характеристик качества поиска

Так, если коэффициент полноты равен 1, то значение коэффициента точности не будет превышать 0.3. Это значит, что 70% выдачи будет состоять из «шума» – нерелевантной информации. Если же значение коэффициента точности равно 1 (т. е. в отсутствии «шума»), то значение коэффициента полноты не превысит 35%. Это значит, что 65% релевантной информации будет потеряно. Если требования пользователя к полноте и точности примерно одинаковы, то, проведя биссектрису координатного угла, получим $k_Т = k_П = 0.8$. Это значит, что выдача будет содержать 80% релевантной информации и 20% «шума». Потери также составят 20%.

В целях проверки работоспособности разработанного комплекса программ было выполнено тестирование всех вариантов его использования, представленных на рис. 1, с применением соответствующих тестовых сценариев. Успешность выполнения всех рассмотренных авторами сценариев позволяет сделать вывод о работоспособности разработанного программного обеспечения.

Таким образом, разработанный комплекс программ может быть успешно применено для организации поиска проектных решений и входящих в их состав документов в электронных архивах проектной документации IT-компаний с достаточным для практического использования уровнем полноты и точности выдачи релевантной информации, что подтверждает практическую значимость работы.

Литература

1. Учет клиентов [Электронный ресурс] `http://yukosoft.ru` (дата обращения 12.03.2022).

2. Электронный архив E-Arch [Электронный ресурс] `http://e-arch.ru` (дата обращения 15.03.2022).

3. ИНТЕРМЕХ – Комплексная автоматизация технической подготовки производства [Электронный ресурс] `https://intermech.ru` (дата обращения 12.03.2022).

4. De Castro L.N., Timmis J.I. Artificial immune systems as a novel soft computing paradigm // Soft Computing - A Fusion of Foundations, Methodologies and Applications. 2003. 7. P. 526–544.

5. de Castro L.N., Von Zuben F.J. Artificial Immune Systems: Part II – A Survey of Application. Technical Report. 2000.

6. Соколов А.В. Методика оценки максимально возможных значений показателей эффективности поиска текстовой информации: Информационные технологии // Новые технологии. 2009. №5. С. 18–24.

MSC 68T20 68-04

Software implementation of the immune algorithm used to search for design solutions in the electronic archive of design documentation

A. A. Butkina, O. A. Gushina, O. A. Kuznetsova, A. V. Shamaev

National Research Ogarev Mordovia State University

Abstract: The article discusses the process of developing software used to search for design solutions stored in the electronic archive of design documentation. The search is carried out using a modified version of the immune algorithm, both according to the given characteristics of design solutions, and according to the characteristics of documents (both text and graphic). Diagrams of use cases and classes, a database organization scheme, a stack of development tools and technologies, the results of testing the functionality and efficiency of the developed software are presented.

Keywords: software, immune algorithm, clonal selection algorithms, design solution, electronic archive, search.

References

1. Customer accounting [Electronic resource] `http://yukosoft.ru` (date of access 12.03.2022).

2. Electronic archive E-Arch [Electronic resource] `http://e-arch.ru` (date of access 15.03.2022).

3. INTERMECH – Complex automation of technical preparation of production [Electronic resource] `https://intermech.ru` (date of access 12.03.2022).

4. L.N. De Castro, J.I. Timmis, Artificial immune systems as a novel soft computing paradigm, *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 2003, 7, P. 526–544.

5. L.N. de Castro, F.J. Von Zuben, Artificial Immune Systems: Part II – A Survey of Application, Technical Report, 2000.

6. A.V. Sokolov, Methodology for estimating the maximum possible values of indicators of the efficiency of text information retrieval: Information Technology, *New technologies*, 2009, 5, P. 18–24.