

УДК 519.6

ПРОГРАММНАЯ СИСТЕМА “ВИГОЛТ” ДЛЯ РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ, ВОЗНИКАЮЩИХ В ТРАНСПОРТНОЙ ЛОГИСТИКЕД. С. Бухаров¹, А. Л. Казаков²

Представлено описание программной системы “ВИГОЛТ”, предназначенной для решения ряда логистических задач в непрерывной постановке: об оптимальном маршруте, об оптимальном размещении объектов логистической инфраструктуры и о сегментации логистических зон. В рамках системы реализованы вычислительные алгоритмы, основанные на оптико-геометрической аналогии. Предложен параллельный алгоритм вычисления задачи об оптимальном размещении нескольких логистических объектов. Проведено тестирование параллельного алгоритма на ЭВМ с различным числом процессоров. Работа выполнена при частичной финансовой поддержке РФФИ (код проекта 11-07-00245).

Ключевые слова: математическое моделирование, программная система, вычислительные методы, параллельные вычисления, транспортная логистика.

1. Введение. Широта и разнообразие оптимизационных задач, возникающих при математическом моделировании работы транспортно-логистических систем, приводит к необходимости разработки методики их численного решения в ситуациях, когда построение точного решения является проблематичным. Для исследования ряда таких задач авторами был предложен подход [1, 2], основанный на аналогии с геометрической оптикой и отысканием экстремума интегрального функционала.

Ранее в работах чл.-корр. РАН В.Н. Ушакова и его учеников [3, 4] подобный подход и его численная реализация применялись при решении задач управления подвижными объектами на плоскости при наличии фазовых ограничений. Особенностью данных работ является построение множеств достижимости и применение методов вычислительной геометрии.

В работах Вл. Вит. Башурова [5] оптико-геометрическая аналогия использовалась при решении задач безопасности:

- а) задача оптимизации системы физической защиты охраняемого объекта;
- б) задача обезвреживания нарушителя;
- в) задача безопасности передвижения.

Эти задачи сведены к поиску оптимального маршрута, и поэтому при построении решения используются полярные координаты.

В настоящей статье представлено описание разрабатываемой авторами программной системы “ВИГОЛТ”, предназначенной для проведения вычислительного эксперимента при исследовании транспортно-логистических задач. Все реализованные в системе алгоритмы объединяет единый подход [1, 2]. Опишем программную систему более подробно.

Программная система реализована на языке программирования Pascal в среде разработки Delphi. Система имеет следующую структуру: набор настроек, повышающих гибкость системы, и четыре взаимосвязанных модуля:

— модуль “Среда” предназначен для задания проницаемости среды для световой волны; изменение размерности среды влияет как на режим отображения графических элементов, так и на точность получаемого решения (чем меньше шаг дискретизации, тем корректнее получаемый результат);

— модуль “Волна” предназначен для реализации распространения световой волны в заданной среде и хранения списка волн, выпущенных из различных источников; построение волны можно производить с помощью различных алгоритмов, отличающихся точностью вычисления световых фронтов и, соответственно, скоростью построения решения;

— модули “Алгоритмы” и “Волна с границы” предназначены для построения двумерных изображений, реализуют различные алгоритмы решения логистических задач (об оптимальном маршруте, об оптимальном размещении объектов логистической инфраструктуры, о сегментации логистических зон).

¹ Иркутский государственный технический университет (ИрГТУ), ул. Лермонтова, 83, 664074, г. Иркутск; аспирант, e-mail: bukharovds@gmail.com

² Институт динамики систем и теории управления СО РАН (ИДСТУ СО РАН), ул. Лермонтова, 134, 664033, г. Иркутск; гл. науч. сотр., e-mail: kazakov@icc.ru

В рамках модуля “Алгоритмы” реализован алгоритм поиска решения задачи о размещении нескольких логистических центров, основанный на многопоточной реализации основного вычислителя. Применение параллельного вычисления позволяет существенно ускорить процесс построения решения.

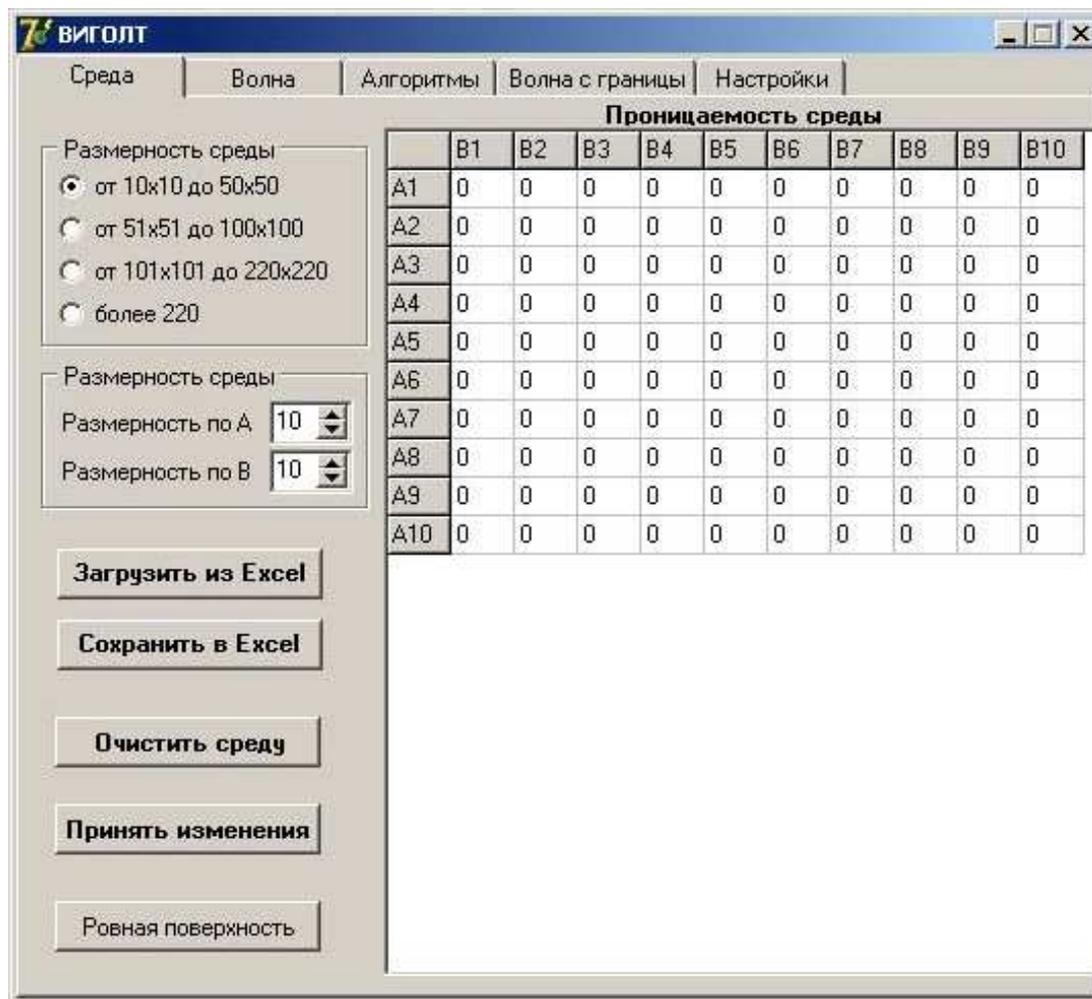


Рис. 1. Вкладка “Среда”

2. Модуль “Среда”. Данный модуль представлен на рис. 1 в виде вкладки “Среда”. Проницаемость среды можно задать вручную, указав конкретное число узлов, в которых устанавливается значение проницаемости (где значение, равное “0”, определяет минимальную проницаемость, а значение “-1” — непроходимый для света барьер). Заданную среду можно загрузить из Excel-файла и сохранить в Excel-файл. От выбранной размерности зависит величина элементов построения изображений в подсистеме “Алгоритмы”. Чем больше размерность, тем меньше элементы изображения. Если размер элементов меньше одного пикселя, то построение изображения не производится.

3. Модуль “Волна”. Данный модуль представлен на рис. 2 в виде вкладки “Волна”. Для распространения волны необходимо указать расположение источника в таблице, дважды щелкнув на определенную ячейку. “Пуск” волны осуществляется после нажатия кнопки “Пустить волну”. Кроме того, реализована возможность сохранения волны в Excel-файл или в специальный список. Из этого списка волну можно удалить, вывести в таблицу (например, для построения фронтов световой волны в модуле “Алгоритмы”) или временно отключить (что полезно при исследовании некоторых задач в модуле “Алгоритмы”). Удаление, вывод и отключение осуществляются нажатием указателя мыши на необходимую строку из списка и выбором определенного действия в выпадающем меню.

“Пуск” волны осуществляется из указанного источника света в соответствии с принципом Гюйгенса [6]: все точки, до которых доходит фронт световой волны, становятся вторичными источниками волн. В каждой точке области фиксируется следующая информация: время, за которое она достигнута, и координата источника, из которого выпущена волна. Из вторичных источников производится аналогичный

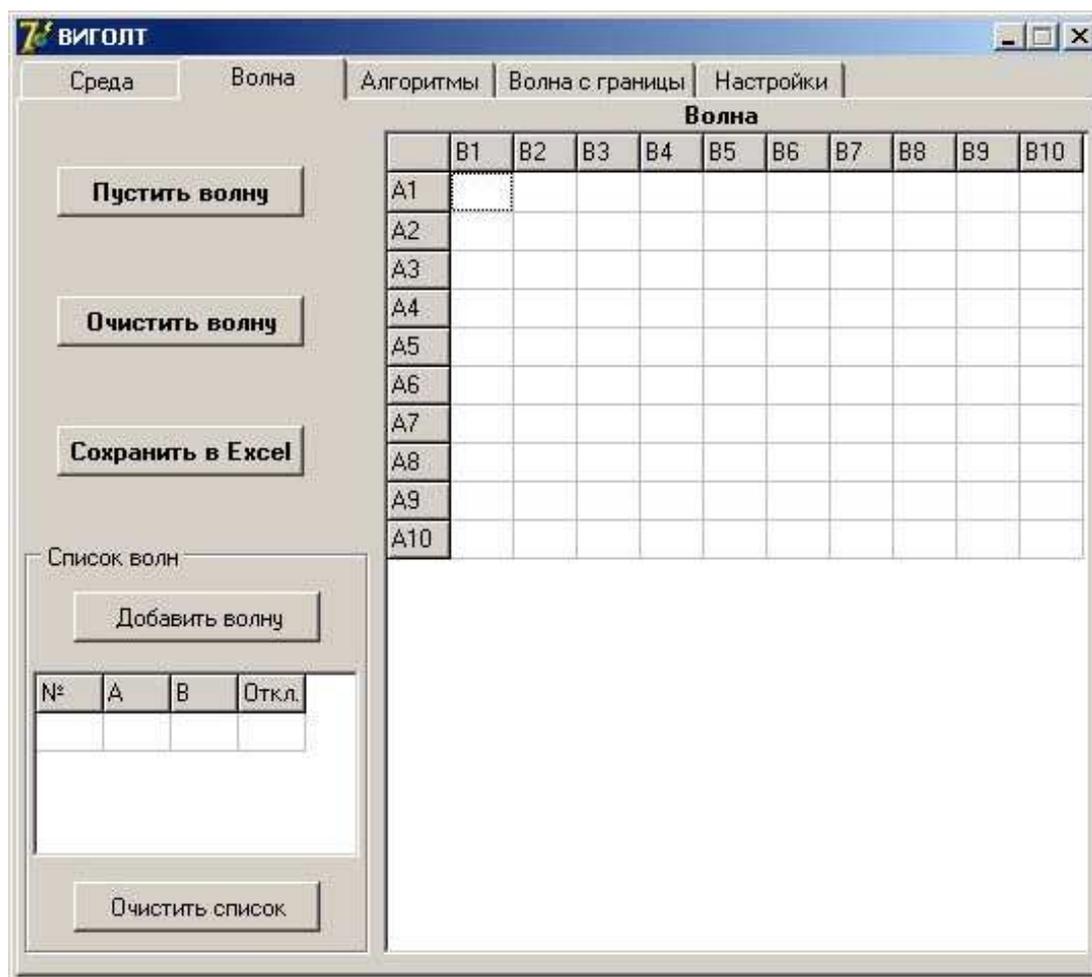


Рис. 2. Вкладка “Волна”

пуск световой волны. Процесс распространения волны продолжается до тех пор, пока не будет заполнена вся рассматриваемая область. Имеется возможность построить фронт световой волны в заданный момент времени, для этого строится огибающая вторичных источников.

В общем случае построить аналитическое выражение для фронта не представляется возможным, поэтому проводится численная реализация описанного алгоритма. В заданной области строится прямоугольная сетка, по которой и движется волна. Этот метод прост в реализации и позволяет определить для каждой точки, достиг ли ее фронт волны к данному моменту времени. Однако для получения достаточной точности необходимо построить очень мелкую сетку, что соответственно ведет к увеличению вычислительных затрат.

Для построения световой волны в программной системе реализованы волновой алгоритм Ли и разработанные авторами алгоритмы однослойной и трехслойной аппроксимации.

Волновой алгоритм Ли — это метод трассировки печатных плат [2], который эффективен только при построении маршрутов в однородной среде (на плоскости), так как в каждом узле сетки не производится фиксация времени достижения.

При однослойной аппроксимации распространение волны осуществляется по восьми направлениям (по восьми соседним точкам), поэтому фронт волны в однородной среде имеет форму восьмиугольника. Согласно принципу Ферма свет в однородной среде распространяется прямолинейно [6] и фронты имеют форму окружности.

Так как однослойная аппроксимация дает достаточно “грубое” приближение, то разработана модификация алгоритма, основанная на трехслойной аппроксимации распространения волны (аппроксимация по 32 соседним точкам). На рис. 3 представлены фронты волны, построенные при помощи трехслойной аппроксимации в однородной среде.

4. Модуль “Алгоритмы”. Данный модуль представлен на рис. 3 в виде вкладки “Алгоритмы”.

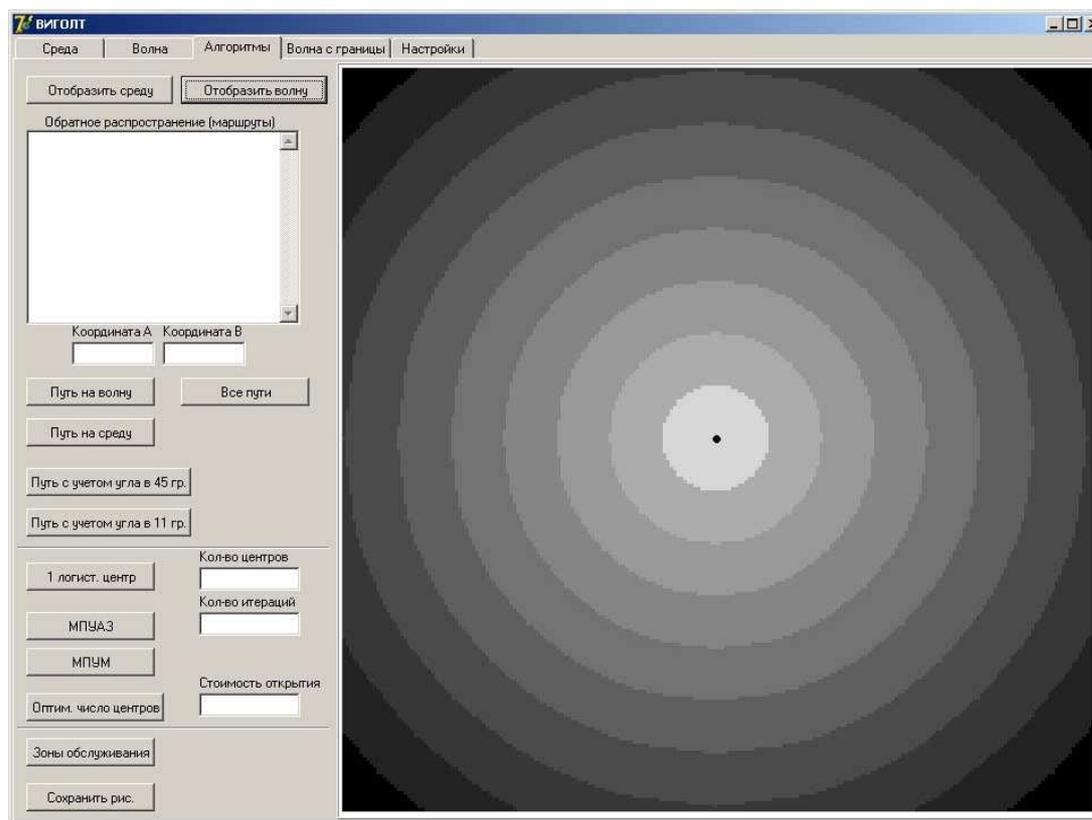


Рис. 3. Вкладка “Алгоритмы” с построенными фронтами световой волны

Модуль “Алгоритмы” взаимодействует с модулями “Волна” и “Среда”, получая данные о времени распространения волны и проницаемости среды в узлах.

Для пользователя доступны следующие функции: вывод линий уровня ландшафта (кнопка “Отобразить ландшафт”), построение фронтов световой волны (кнопка “Отобразить волну”), вывод оптимального маршрута поверх ландшафта или волны, вывод всех возможных оптимальных маршрутов (кнопка “Все пути”), сохранение изображения (кнопка “Сохранить рис.”).

Для построения маршрута от источника волны до некоторой точки, во-первых, необходимо указать расположение источника света и произвести “пуск” волны в модуле “Волна”, во-вторых, необходимо указать координаты точки назначения в полях “Координата А” и “Координата В” и нажать одну из кнопок построения маршрута. Если требуется построить маршрут от ранее сохраненного источника (список волн — вкладка “Волна”), то для этого необходимо вывести уже просчитанную волну в таблицу “Волна”, после этого все процедуры будут выполняться относительно данной волны.

Возможность сохранения изображения также имеется в модуле “Волна с границы”, поэтому с целью оптимизации программного кода реализована единая функция `saveImages()`. В качестве входного параметра функция `saveImages()` получает объект типа `Image`, на области рисования (`Canvas`) которого и строится изображение. Данное улучшение программного кода не является единственным и произведено со всеми однотипными процедурами и функциями.

В модуле “Алгоритмы” реализованы численные алгоритмы решения задач:

- об оптимальном маршруте;
- об оптимальном маршруте с ограничением на кривизну;
- об оптимальном размещении одного логистического центра;
- об оптимальной идентификации и сегментации логистических зон;
- об оптимальном размещении нескольких логистических центров;
- об оптимальном количестве логистических центров.

Дадим общее описание методов решения задач.

4.1. Метод решения задачи об оптимальном маршруте. Из начальной точки (источник света) выпускается волна, и для каждого узла сетки определяется значение времени, за которое волна достигла

данной точки. Кроме того, на каждом шаге фиксируется координата предыдущего узла сетки, что позволяет, двигаясь в обратном направлении по времени из любой точки $A(x, y)$, которой достигла световая волна, построить экстремаль в виде ломаной. Возможно несколько решений при одном и том же наборе данных.

Таким образом определяется искомый маршрут Γ , преодолеваемый за минимальное время (принцип Ферма):

$$T = \min_{\Gamma} \int_{\Gamma} \frac{d\Gamma}{v(x, y)},$$

где $v(x, y)$ — мгновенное значение скорости в точке (x, y) .

4.2. Метод решения задачи об оптимальном маршруте с ограничением на кривизну. Метод является обобщением вышеописанного алгоритма, который на каждом шаге построения маршрута Γ учитывает перепады высот и необходимый радиус кривизны

$$R = \frac{\sqrt{([x'(t)]^2 + [y'(t)]^2)^3}}{|x'(t)y''(t) - y'(t)x''(t)|} \geq C,$$

где R — радиус кривизны кривой Γ и C — минимальный разрешенный радиус кривизны.

Учет радиуса кривизны осуществляется при построении маршрута, где для каждого узла сетки проверяется не только время достижения, но и угол поворота строящегося пути. Максимальное значение угла поворота ограничено минимальным радиусом кривизны.

4.3. Метод решения задачи об оптимальном размещении одного логистического центра. В начальный момент времени из источников выпускаются световые волны. Для каждой точки области фиксируется максимальное значение времени, за которое возможно добраться до данной точки. Затем находится точка с наименьшим значением времени, которая и будет определять оптимальное расположение логистического центра:

$$T = \min_A \min_i \int_{\Gamma} \frac{d\Gamma}{v_i(x, y)}, \quad i = \overline{1, k},$$

где $v_i(x, y)$ — скорость распространения волны из i -го источника.

4.4. Метод решения задачи об оптимальной идентификации и сегментации логистических зон. Из k источников выпускаются световые волны. Для каждой точки $A(x, y)$ фиксируется минимальное значение времени $T = \min_i \int_{\Gamma} \frac{d\Gamma}{v_i(x, y)}$, $i = \overline{1, k}$, за которое возможно добраться до данной точки из одного из источников света. Кроме того, для данной точки фиксируется номер источника света, который и устанавливает принадлежность точки $A(x, y)$ к определенной логистической зоне. Если данную точку достигают две или более волн одновременно, то эта точка находится на границе логистических зон.

4.5. Методы решения задачи об оптимальном размещении нескольких логистических центров. Разработаны два численных метода [7], основанных на итеративном принципе поиска решения: метод последовательного улучшения с использованием мультистарта (МПУМ) и метод последовательного улучшения с выделением активных зон (МПУАЗ). Данные методы позволяют определить расположение нескольких логистических центров (например, складов или магазинов), при котором обеспечиваются минимальные суммарные затраты на обслуживание клиентов.

Идея метода МПУМ заключается в итеративном улучшении получаемого решения при помощи последовательной сегментации области на зоны, обслуживаемые одним центром, и отыскании оптимального расположения данного центра в соответствующей зоне. Генерация начальных положений повторяется многократно с целью поиска различных локальных минимумов. Среди полученных решений выбирается наилучшее.

Идея метода МПУАЗ заключается в выделении на полигоне обслуживания двух зон: “активной” и “неактивной”. Данные области строятся относительно общего центра тяжести O . Размер “неактивной” области варьируется в зависимости от числа складов, которые необходимо разместить, и от числа потребителей, попавших в “активную” зону. Если в “активной” зоне число потребителей n_a меньше числа складов m , то радиус r уменьшается так, чтобы $n_a \geq m$. Выделение зон необходимо для размещения складов в наибольшей удаленности от центра тяжести O . После определения размеров зон перебираются некоторые комбинации начальных положений складов относительно потребителей в “активной” зоне.

4.6. Метод решения задачи об оптимальном количестве логистических центров. Идея метода заключается в последовательном увеличении числа складов, начиная с двух, с отысканием для них оптимального расположения. Число складов увеличивается до тех пор, пока не будет определен минимум затрат на обслуживание клиентов. Поиск оптимального расположения логистических центров осуществляется методом МПУАЗ. Очевидно, что при необходимости разместить n центров решение будет найдено не более чем за $n - 1$ итерацию.

4.7. Многопоточная реализация вычислителя для метода МПУАЗ. На рис. 4 представлена модельная задача, в которой 4 склада (белые круги) размещены среди 16 потребителей (черно-белые точки) так, что обеспечивается минимальное суммарное время доставки. В [7] отмечено, что методу МПУМ не удается построить решение при 6 и более складах. Метод МПУАЗ позволяет решить данные задачи, однако для этого требуется использовать больше вычислительных ресурсов. На размещение 5 складов методом МПУАЗ требуется 66 минут (расчет проведен на двухъядерном процессоре Genuine Intel CPU T2300, частота процессора 1.66 ГГц).

Данная проблема решена при помощи распараллеливания процедуры поиска координат складов на уровне данных (рис. 5). Каждый поток производит решение одной и той же задачи для различных начальных данных, получаемых из глобального массива. Безопасный доступ к массиву координат начального расположения складов обеспечивается посредством защищенного блока CriticalSection (критическая секция). Данный блок является стандартным механизмом синхронизации потоков в среде Delphi. Внутри критической секции программный код выполняется последовательно, и выполнять заданные функции может только тот поток, который первым достиг блока CriticalSection и осуществил захват секции вызовом метода Enter.

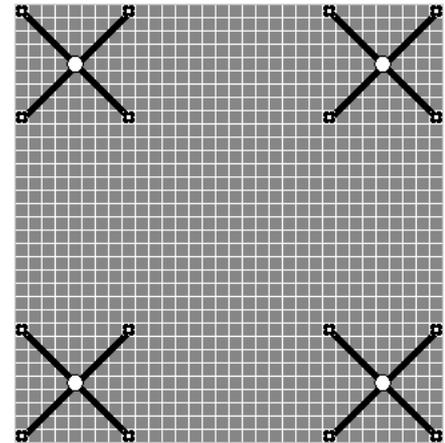


Рис. 4. Модельная задача: 4 склада, 16 потребителей

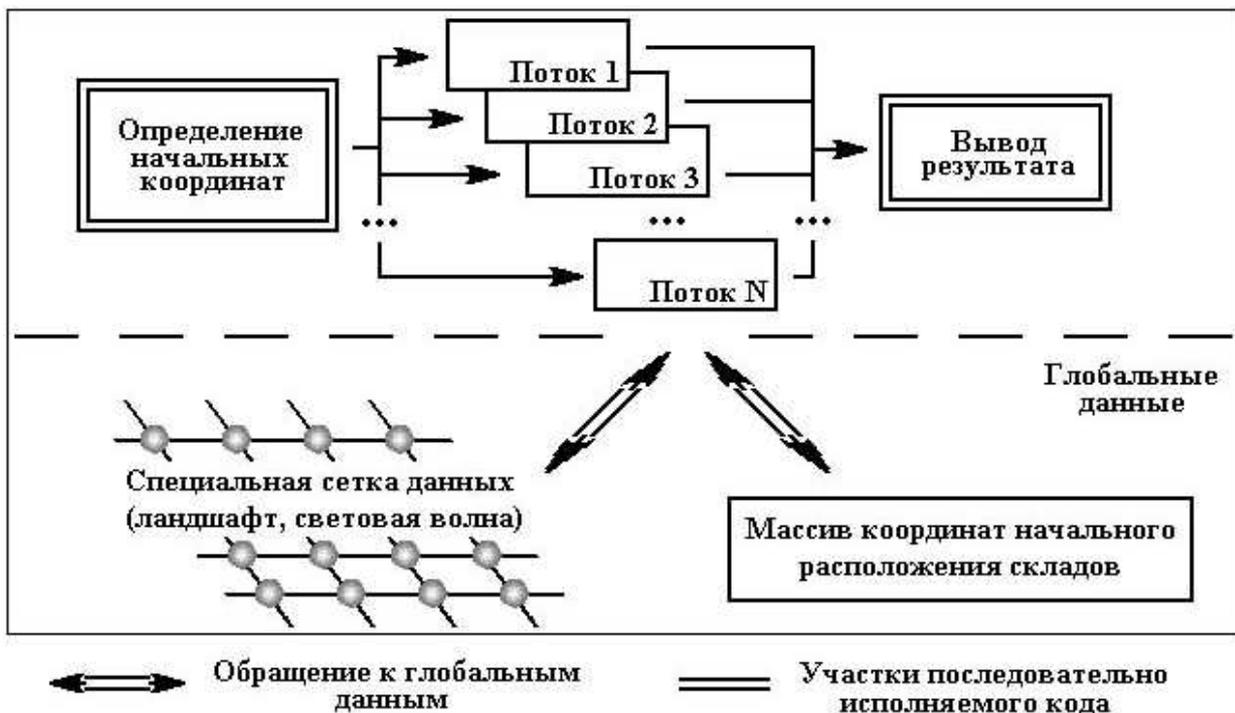


Рис. 5. Схема вычислительного процесса

При такой организации вычисления (рис. 5) каждый поток выполняет единый программный код. Определение начальных координат складов, вывод результата и обращение к глобальным данным осуществляется последовательно. Специальная сетка характеризует собой массив данных, в узлах которого

хранится информация о проникаемости среды и времени достижения световой волной данного узла. Кроме того, каждый узел данной сетки хранит координату того вторичного источника, из которого волна в данный узел приходит раньше.

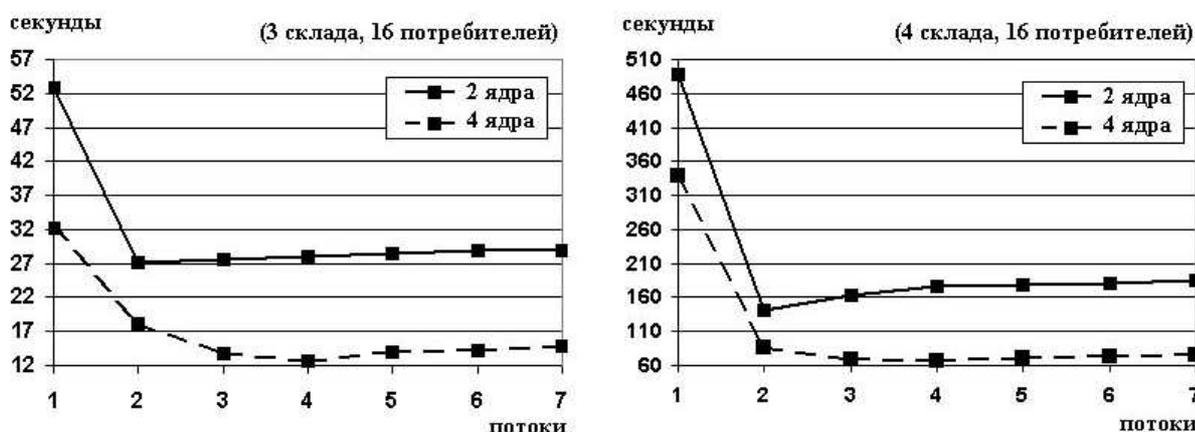


Рис. 6. Зависимость времени вычисления от количества потоков

Количество параллельно исполняемого программного кода составляет 50%. На рис. 6 представлены графики зависимостей времени вычисления от количества потоков для ЭВМ с различным числом процессоров: 2-ядерный AMD Athlon II Dual-Core M300 (2.00 ГГц) и 4-ядерный Intel Core(TM)2 Quad CPU (2.33 ГГц). Решались модельные задачи о размещении трех и четырех складов относительно шестнадцати потребителей (рис. 4). Из графиков можно видеть, что наилучшая скорость вычисления на 2-ядерном процессоре обеспечивается при двух потоках, а на 4-ядерном процессоре — при четырех потоках. Данный факт объясняется тем, что каждый поток обеспечивает решение трудно вычисляемой задачи и практически полностью загружает работой определенный процессор [8].

В табл. 1 и 2 приведены зависимости времени вычисления от числа потоков для задачи оптимального размещения 3 и 4 складов среди 16 потребителей.

Таблица 1
Зависимость времени вычисления от количества потоков
(3 склада, 16 потребителей)

Время вычисления (сек.) на определенном процессоре	Число потоков						
	1	2	3	4	5	6	7
AMD Athlon II Dual-Core	52.7	27.2	27.5	27.9	28.6	28.8	28.9
Intel Core(TM)2 Quad CPU	32.3	18.2	13.9	12.6	13.9	14.2	14.9

Таблица 2
Зависимость времени вычисления от количества потоков
(4 склада, 16 потребителей)

Время вычисления (сек.) на определенном процессоре	Число потоков						
	1	2	3	4	5	6	7
AMD Athlon II Dual-Core	488.4	141.1	162.5	176.3	178.5	181.4	186.2
Intel Core(TM)2 Quad CPU	338.5	86.5	68.2	66.0	70.8	74.1	76.4

Наименьшее время поиска оптимального расположения трех складов на 2-ядерном процессоре (2 потока) составляет 27.16 секунд, а на 4-ядерном процессоре (4 потока) — 12.61 секунд. Таким образом, увеличение количества потоков в 2 раза (при соответствующем числе процессоров) уменьшило время вычисления в 2.15 раза. На решение задачи о размещении пяти складов (при наилучших соотношениях числа процессоров и потоков) было затрачено следующее количество времени: как упоминалось ранее, при одном потоке — 66 минут, при двух потоках — 8.12 минуты, при четырех потоках — 3.68 минуты (что в 2.2 раза быстрее, чем при двух потоках).

5. Модуль “Волна с границы”. Данный модуль представлен на рис. 7 в виде вкладки “Волна с границы”. Построение фронтов волны осуществляется внутри выделенной области. Источником волны выступает не одна точка, а заданное многообразие. Из каждой точки выпускается свет, и через некоторое время Δt строится огибающая световых волн (принцип Гюйгенса), которая является искомым фронтом.

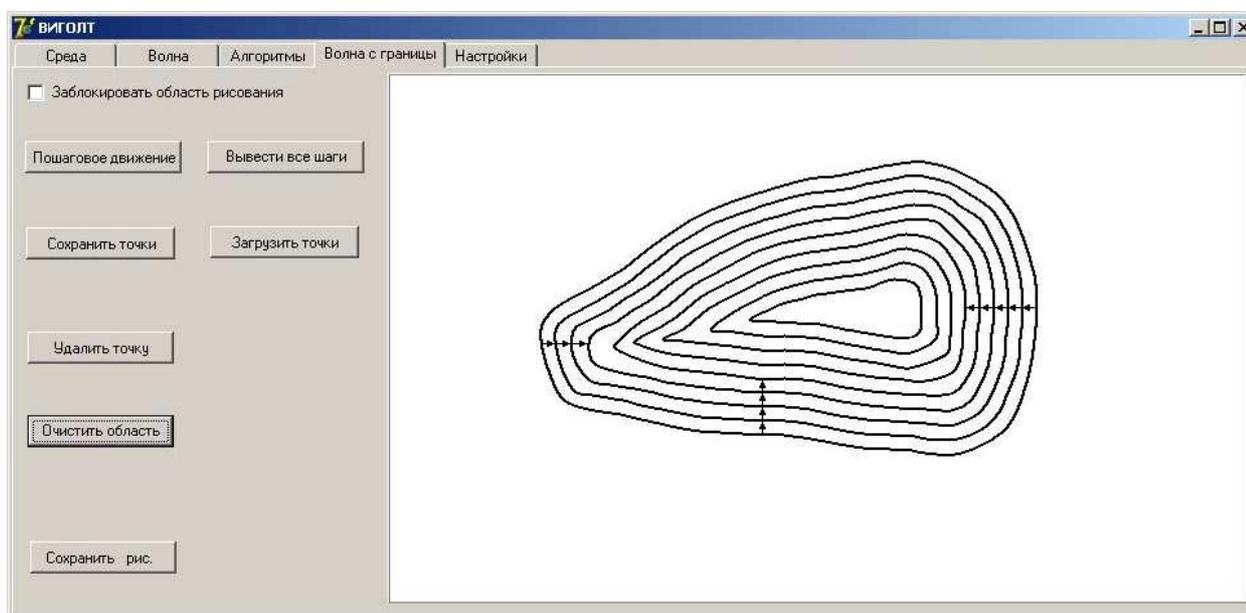


Рис. 7. Вкладка “Волна с границы” с построенными фронтами световой волны

Данную область можно задать вручную, последовательно выделяя контур нажатием указателя мыши на области рисования. Если при выделении области некоторые точки установлены в несоответствующем месте, то их можно удалить нажатием кнопки “Удалить точку”. Точки удаляются по одной, начиная с последней установленной.

Выделенные точки являются опорными точками, которые необходимы при построении замкнутой линии на основе кривых Безье. Для построения кривых Безье используется стандартная функция *PolyBezier* в среде Delphi.

Кроме того, доступны следующие функции: загрузка (и сохранение) точек из Excel-файла, сохранение изображения (кнопка “Сохранить рис.”), очистка области рисования и построение фронтов световой волны — пошаговое (кнопка “Пошаговое движение”) и вывод всех шагов (кнопка “Вывести все шаги”).

В рамках алгоритма построения фронта световой волны реализована функция, обеспечивающая решение проблемы возможной потери гладкости фронта [9] (разрыва производной) при удалении от начального многообразия. На рис. 7 представлено построение фронтов волны с учетом описанной особенности. Как можно видеть, через несколько шагов образуется излом фронта волны в форме “ласточкиного хвоста”.

Данным алгоритмом обеспечивается решение, во-первых, задачи о размещении логистических объектов (таких как, например, центры захоронения радиоактивных отходов), при котором обеспечивается их минимальная доступность для населения, или, иначе говоря, “максимальная недоступность”.

Во-вторых, задачи о размещении логистических объектов (например, спасательных пунктов) в местах, из которых обслуживание прилегающей территории будет производиться за минимально возможное время.

6. Настройки системы. На рис. 8 представлена вкладка “Настройки”, на которой доступны следующие действия:

- переключение с цветного на черно-белое рисование;
- переключение со сплошного рисования на вкладке “Алгоритмы” на рисование с выделением каждого элемента области рисования;
- переключение на многопоточное исполнение метода МПУАЗ;
- изменение режима прорисовки маршрута: точки, сплошная линия;
- выбор алгоритма распространения волны: волновой алгоритм Ли (метод трассировки печатных плат), однослойная аппроксимация (распространение по восьми соседним точкам), трехслойная аппроксимация (распространение по тридцати двум соседним точкам);

- смена способа нахождения оптимального расположения одного логистического центра: максимальное время доставки минимально, суммарное время доставки минимально;
- включение функции учета перепада высот в задаче об оптимальном маршруте с ограничением на кривизну.

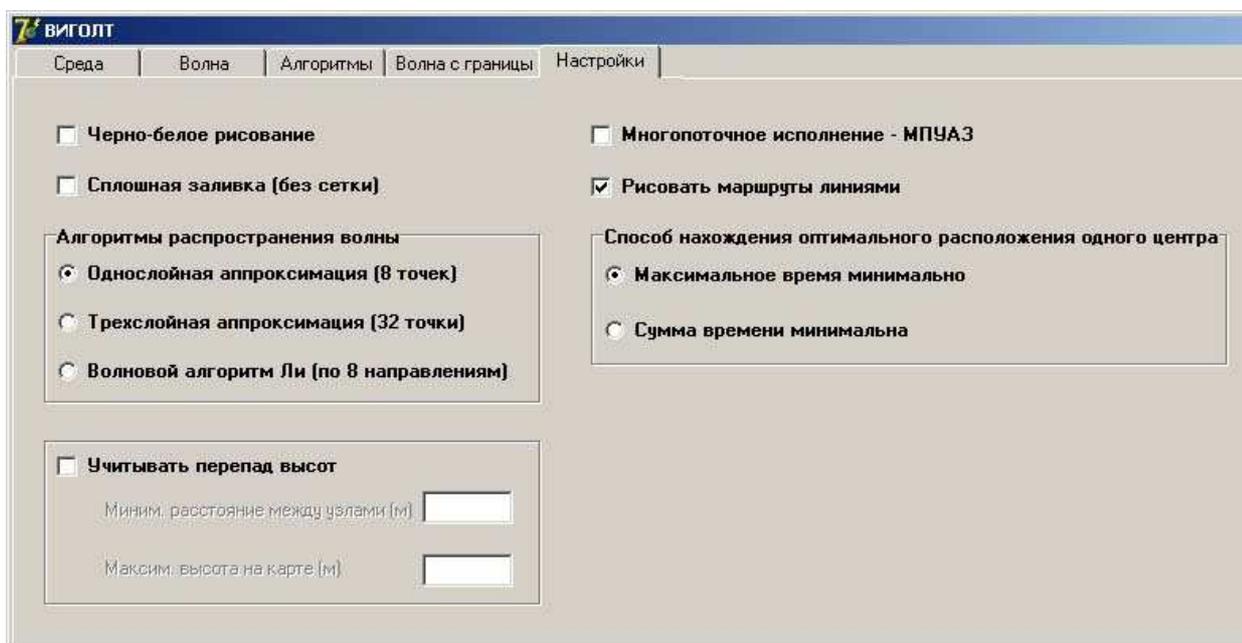


Рис. 8. Вкладка “Настройки”

7. Заключение. Программная система “ВИГОЛТ” позволяет эффективно находить численное решение оптимизационных задач транспортной логистики, математическое описание которых приводит к задачам вариационного исчисления специального вида. Спектр решаемых задач достаточно широк: от классической задачи о прокладке оптимального маршрута между двумя пунктами до задачи о размещении нескольких логистических центров с сегментацией зон обслуживания в непрерывной постановке. Программная система применялась при решении следующих прикладных задач: идентификация и сегментация логистических зон утилизации старых автомобилей в Иркутской [2] и Свердловской областях [10]; оптимальная прокладка высокоскоростной железнодорожной магистрали Екатеринбург–Челябинск [11].

Решаемые задачи являются сложными с вычислительной точки зрения и требуют больших затрат времени и ресурсов, что в ряде случаев представляет серьезную проблему. Для ее преодоления реализован параллельный алгоритм вычисления для метода МПУАЗ, позволяющий в несколько раз сократить время поиска решения, при этом количество параллельно исполняемого программного кода составляет 50%.

Разработанная программная система “ВИГОЛТ” позволяет пользователю изменять размерность оптической среды, алгоритмы аппроксимации распространения волны, проницаемость среды в определенных точках рассматриваемой области, что необходимо для установления необходимой точности решения. Доступно редактирование данных непосредственно в программной системе, а также возможна загрузка (и сохранение) данных из Excel-файлов. Предусмотрено задание дополнительных ограничений, например при построении маршрутов и поиске оптимального места расположения логистических объектов. Реализовано графическое представление данных и результатов решения задач.

В дальнейшем планируется модернизация некоторых “медленных” алгоритмов, например алгоритма размещения нескольких логистических объектов (в непрерывной постановке) с целью сокращения времени вычисления, для чего, вероятно, потребуется организация многопоточного вычислителя. Кроме того, планируется разработка графического представления в трехмерном формате, что, возможно, позволит заранее выявить специфические особенности исследуемых задач.

СПИСОК ЛИТЕРАТУРЫ

1. Казаков А.Л., Лемперт А.А. Об одном подходе к решению задач оптимизации, возникающих в транспортной логистике // Автоматика и телемеханика. 2011. № 7. 50–57.

2. Казаков А.Л., Лемперт А.А., Бухаров Д.С. Об одном численном методе решения некоторых задач оптимизации, возникающих в транспортной логистике // Вестник ИргТУ. 2011. № 6. 6–12.
3. Матвейчук А.Р., Ушаков В.Н. О построении разрешающих управлений в задачах управления с фазовыми ограничениями // Известия РАН. Теория и системы управления. 2006. № 1. 5–20.
4. Ушаков В.Н., Матвейчук А.Р. Один метод решения задач управления протяженными объектами на конечном промежутке времени // Тр. IX Междунар. Четаевской конференции “Аналитическая механика, устойчивость и управление движением”. Т. 3. Иркутск: ИДСТУ СО РАН, 2007. 253–261.
5. Башуров В.В., Фильмоненкова Т.И. Математические модели безопасности. Новосибирск: Наука, 2009.
6. Ланцош К. Вариационные принципы механики. М.: Физматгиз, 1965.
7. Бухаров Д.С. Определение оптимального количества и расположения логистических центров: математическая модель и численный метод // Вестник ИргТУ. 2012. № 4. 8–14.
8. Таненбаум Э. Современные операционные системы. СПб.: Питер, 2010.
9. Арнольд В.И. Особенности каустик и волновых фронтов. М.: ФАЗИС, 1996.
10. Казаков А.Л., Журавская М.А., Лемперт А.А. Вопросы сегментации логистических платформ в условиях становления региональной логистики // Транспорт Урала. 2010. № 4. 17–20.
11. Журавская М.А., Казаков А.Л., Лемперт А.А., Бухаров Д.С. О методе решения задачи оптимальной прокладки высокоскоростных железнодорожных магистралей с учетом региональных особенностей // Транспорт: наука, техника, управление. 2012. № 2. 41–44.

Поступила в редакцию
23.05.2012
