Зокрема, клітинний генетичний алгоритм на самому початку роботи, частіше за все, дає краще значення цільової функції у порівнянні з простим, але у подальшому процесі роботи в цілому генетичні алгоритми дають майже однаковий результат. Алгоритм імітації відпалу на початку процесу поступається генетичним алгоритмам, але під час подальшої роботи цей алгоритм в більшості випадків показує краще значення цільової функції, ніж генетичні алгоритми.

Висновки. Чисельні дослідження, проведені в даній роботі, показали, що алгоритм імітації відпалу приблизно на 5% ефективніший, ніж генетичні алгоритми для задачі про «Розумну мураху». В даній роботі метод відпалу був застосований та показав також свою ефективність для розв'язання деяких модифікацій задачі про «Розумну мураху», зокрема, «Мураха – 2» та «Мураха – 3».

Перевагою методу відпалу є властивість уникати "пасток" в локальних максимумах (мінімумах) функції, що оптимізується, і продовжувати пошук глобального максимуму (мінімуму). Ще однією перевагою є те, що навіть в умовах браку обчислювальних ресурсів для знаходження глобального максимуму (мінімуму) метод відпалу, як правило, дозволяє отримати досить непогане рішення. Метод відпалу і його модифікації є одним з найбільш ефективних методів випадкового пошуку оптимального рішення для великого класу задач. Результати роботи алгоритмів показали, що метод відпалу не програє генетичним алгоритмам, а для багатьох задач і виграє.

ЛІТЕРАТУРА

1. Бедный Ю.Д. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей» / Бедный Ю.Д., Шалыто А.А. // СПбГУ: ИТМО. – 2007. – [Електронний ресурс]. – Режим доступу: http://is.ifmo.ru/works/ant
2. А.С.Лопатин. Метод отжига в задачах оптимизации / А.С.Лопатин. – 2004. – [Електронний ресурс] – Режим доступу: http://www.math.spbu.ru/user/gran/students/cothesis.pdf

Надійшла до редколегії 06.06.2012.

УДК 519.8, 004.4

КАДОЧНИКОВА Я.Е., к.ф.-м.н, доцент
ЛЕОНОВ А.В., студент

Днепродзержинский государственный технический университет

ИСПОЛЬЗОВАНИЕ ВОЗМОЖНОСТЕЙ ASP.NET ДЛЯ ВИЗУАЛИЗАЦИИ РЕШЕНИЯ ОДНОЙ ЗАДАЧИ ОПТИМАЛЬНОГО РАЗБИЕНИЯ МНОЖЕСТВ

Введение. Интерес к непрерывным задачам оптимального разбиения множества (ОРМ) на непересекающиеся подмножества с целью минимизации некоторого критерия качества разбиения вызван тем, что к ним сводится достаточно широкий класс как теоретически, так и практически важных задач оптимизации, а также задач из других разделов прикладных наук. Отметим, что на протяжении последних тридцати лет Днепропетровской научной школой под руководством Е.М.Киселёвой получены существенные результаты в направлении развития теории ОРМ. Широкий обзор задач теории и практики, сводящихся к математическим моделям оптимального разбиения множеств, а также алгоритмы их решения приведены в монографии [1].

За все время исследований, связанных с решением задач ОРМ, было создано большое количество программных приложений, которые оставались недоступными широкому кругу пользователей. Поскольку задачи ОРМ имеют практическое значение, необходимым шагом является разработка web-сервиса, который смогут использовать сторонние клиенты для решения конкретных прикладных задач.

Постановка задачи. Необходимо разработать web-сервис и клиентское приложение, демонстрирующее работу web-сервиса, для решения непрерывной задачи оптимального разбиения множества на подмножества без ограничений с заданными координатами "центров" подмножеств. Математическая постановка данной задачи приведена в [1] и имеет следующий вид.

Найти

$$\min_{\{\Omega_1,...,\Omega_N\}\in\Sigma_\Omega^N} F\left(\{\Omega_1,...,\Omega_N\}\right),$$

где

$$F\left(\{\Omega_1,...,\Omega_N\}\right)=\sum_{i=1}^{N}\int_{\Omega_i}\left[c(x,\tau_i)+a_i\right]\rho(x)dx,$$

$$\Sigma_\Omega^N=\{\ \{\Omega_1,...,\Omega_N\}:\ \bigcup_{i=1}^{N}\Omega_i=\Omega, mes\left(\Omega_i\bigcap\Omega_k\right)=0,\ i\neq k,\ \ i,k=1,...,N\},$$

функции $c(x,\tau_i)$ – действительные, ограниченные, измеримые по аргументу x, $x=(x^{(1)},...,x^{(n)})\in\Omega$, при любом фиксированном τ_i из Ω для всех $i=1,...,N$; функция $\rho(x)$ – действительная, ограниченная, измеримая и неотрицательная на Ω; $a_1,...,a_N$ – заданные действительные неотрицательные числа.

Результаты роботы. Для решения поставленной задачи было решено использовать популярную платформу .NET Framework [2]. Разработан проект WCF-сервиса Partitioning_Sets_Service и развернут на консольном приложении. На рис.1 приведена структура проекта:

- app.config – файл конфигураций веб-сервиса в формате XML;
- point.cs – класс для описания центра;
- area.cs – класс, который реализует решение задачи ОРМ;
- IPartitioningSetsService.cs – интерфейс, содержащий контракты для организации взаимодействия клиент-сервис;
- PartitioningSetsService.cs – класс, реализующий интерфейс IPartitioningSetsService;
- Program.cs – класс, который отвечает за создание экземпляра веб-сервиса, его запуск и остановку.

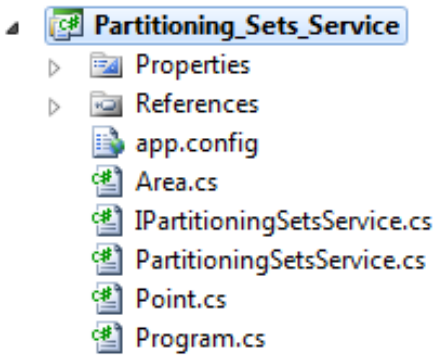

Рисунок 1 – Структура проекта WCF-сервиса

Для создания экземпляра web-сервиса выполняется запуск консольного приложения. Когда служба готова к обработке запросов, появляется сообщение «Service is running».

Когда с клиентского приложения приходит запрос, web-сервис его обрабатывает, вызывает нужные методы и отправляет ответ клиенту. Результаты по обработке каждого запроса отображаются в консольном приложении. Так на рис.2 можно увидеть время получения запроса, входные параметры и результирующую матрицу.

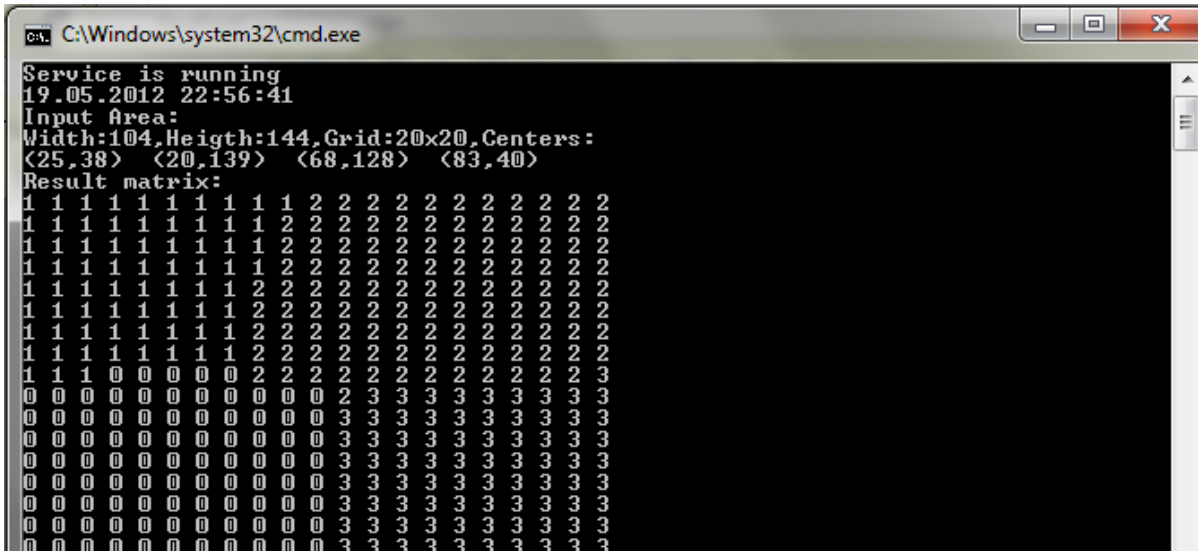

Рисунок 2 – Пример результата выполнения запроса

Проект Service_Client – клиентское приложение, которое разработано для наглядной демонстрации результатов работы web-сервиса. Для разработки клиентского приложения была выбрана технология ASP.NET Web Forms, которая позволила достичь следующих целей:

- интуитивно-понятный интерфейс;
- удобный ввод входных данных решаемой задачи;
- формирование результатов в понятном для пользователей виде;
- корректная обработка исключительных ситуаций, вызванных ошибочными действиями пользователя.

На рис.3 представлена структура проекта Service_Client.

Рассмотрим структуру проекта более детально:

- Partitioning_Sets_Service – ссылка на web-сервис. При добавлении ссылки к проекту автоматически формируются классы для технической реализации взаимодействия клиент-сервис;
- Resources – каталог для хранения дополнительных файлов проекта (изображений, анимации и т.п);
- Scripts – каталог для хранения JScript библиотек;
- Styles – каталог для хранения CSS-стилей;
- DrawImageClass.cs – класс, реализующий работу с изображениями;
- Global.asax – содержит глобальные события приложения;
- Web.config –файл конфигураций приложения в формате XML;

- Task.aspx – страница ASP.NET Web Forms, которая содержит HTML разметку и классы с реализацией серверной логики;
- Task.aspx.cs – часть страницы ASP.NET Web Forms, которая реализует обработку событий страницы и ее элементов управления.

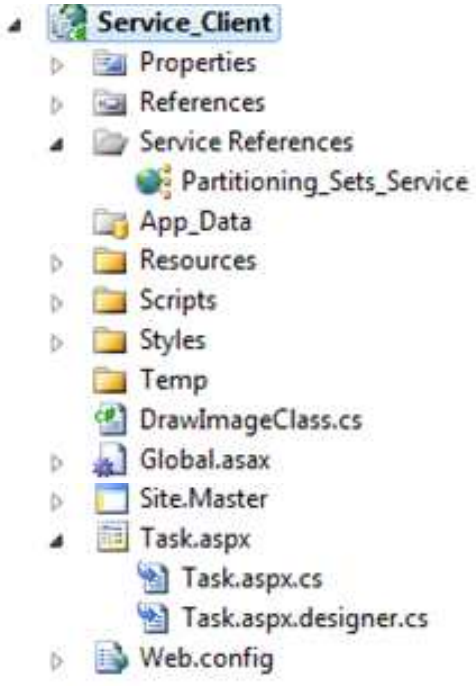

Рисунок 3 – Структура проекта клиентского приложения

Стартовая страница клиентского приложения в web-браузере выглядит, как показано на рис.4. Здесь пользователю предлагается загрузить карту, на которой будет проводиться решение задачи ОРМ. Если выбранный файл имеет некорректный формат, то пользователь увидит сообщение об ошибке.

Рисунок 4 – Стартовая страница консольного приложения

После удачной загрузки карты пользователю необходимо указать допустимую область на ней. Выбор области осуществляется мышью (рис.5), после чего на сервере выполняется обработка изображения и генерируется web-страница, содержащая выбранную область с поддержкой ввода центров двумя способами (рис.6):

- кликом мыши в нужной точке изображения;
- нажатием «Add Center» с последующим вводом координат центра.

После ввода всех параметров нажатием кнопки «Next» инициируется отправка данных разработанному web-сервису, который получает решение сформулированной

Рисунок 5 – Вид web-страницы при выборе области

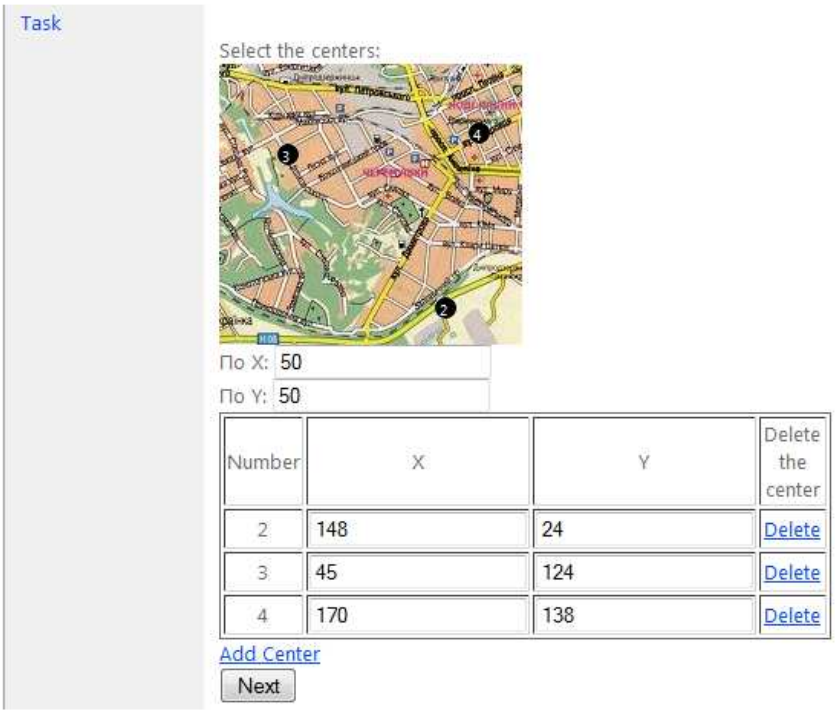

Рисунок 6 – Вид web-страницы после ввода параметров

задачи и возвращает результат её решения в виде, представленном на рис.7. Полученные результаты могут быть сохранены в виде изображения или текста.

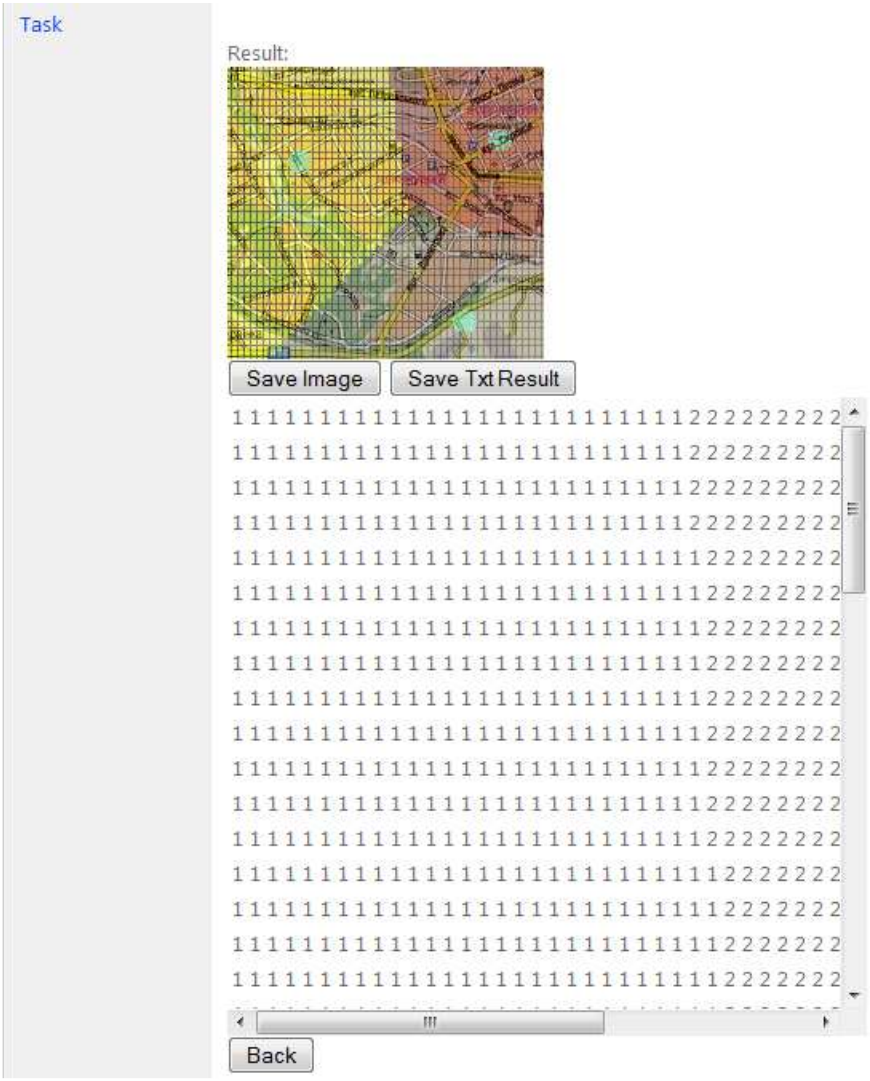

Рисунок 7 – Вид web-страницы с результатами решения задачи

Выводы. Разработан web-сервис и клиентское приложение, которое демонстрирует работу web-сервиса, для решения непрерывной задачи оптимального разбиения некоторого заданного множества на непересекающиеся подмножества без ограничений с заданными координатами "центров" этих подмножеств. Для реализации сервис-ориентированного проекта использовалась платформа .NET Framework 4.0.

ЛИТЕРАТУРА

1. Киселёва Е.М. Непрерывные задачи оптимального разбиения множеств: теория, алгоритмы, приложения: Монография / Е.М.Киселёва, Н.З.Шор. – К.: Наук. думка, 2005. – 564с.
2. Макки А. Ведение в .NET 4.0 и Visual Studio 2010 для профессионалов / А.Макки. – Вильямс, 2010. – 415с.

Поступила в редколлегию 18.06.2012.