

Оборудование, инструменты, программы: при уборке снега используется снегоуборочная техника. Снег вывозят и плавят в снеготаялках.

Пользователи, лица принимающие решения, исполнители решений, эксплуатационный персонал:

Руководитель – получает информацию от диспетчера и определяется масштабы работы, т.е. определяет какую и в каком количестве отправить технику на определенные маршруты.

Диспетчер – принимает информацию от метеорологических служб и отправляет её руководителю.

Водитель-механик – управляет снегоуборочной техникой.

**Вывод.** Была разработана функциональная система автоматизированной информационной системы планирования уборки снега на городских улицах. В результате применения разработанной автоматизированной системы, предприятию по уборке снега удастся извлечь максимальную прибыль без существенных затрат.

*Работа выполнена под научным руководством доцента кафедры АСОИУ КНИТУ-КАИ, кандидата технических наук И.С. Ризаева.*

## ЛИТЕРАТУРА

1. Ризаев И.С. , Теория принятия решений. Учебное пособие. – Казань, Изд-во «Мастер Лайн», 2014. – 132с.
2. Ризаев И.С., Яхина З.Т. Базы данных/Лабораторный практикум. – К., «Мастер Лайн», 2003.
3. Суздальцев В.А., Осипова А.Л., Зарайский С.А., Проектирование информационных систем. Учебное пособие. Казань: Изд-во Казан. гос. техн. ун-та, 2007. – 86 с.

УДК 007.51

## ПЕРСОНАЛИЗИРОВАННЫЙ ИНТЕЛЛЕКТУАЛЬНЫЙ МЕТОД ОПТИМИЗАЦИИ ИСПОЛЬЗУЕМЫХ СИСТЕМНЫХ РЕСУРСОВ ГЕОИНФОРМАЦИОННОЙ СИСТЕМЫ

**А.А. Глушков<sup>1</sup>**

<sup>1</sup>аспирант кафедры информационно-аналитических систем безопасности, Южный федеральный университет, г. Таганрог, Российская Федерация, e-mail: [andrey@glushkov.net](mailto:andrey@glushkov.net)

**Аннотация.** В данной работе приводится алгоритм решения проблемы оптимизации используемых системных ресурсов геоинформационной системы, с помощью выявления приложений, которые не являются востребованными в данный момент на устройстве пользователя путем анализа исторических данных использования устройства в среде геоинформационной системы для определенного пользователя.

*Ключевые слова: интеллектуальная система, оптимизация памяти, кластеризация, адаптация, геоинформационные системы.*

## **GEOGRAPHICAL SYSTEM RESOURCES USAGE PERSONALIZED INTELLECTUAL OPTIMIZATION METHOD**

**Andrey Glushkov<sup>1</sup>**

<sup>1</sup>Postgraduate of Information Analytical Systems of Safety of Technical Sciences Department, Southern Federal University, Taganrog, Russian Federation, e-mail: [andrey@glushkov.net](mailto:andrey@glushkov.net)

**Abstract.** The algorithm for geographical system memory consumption optimization problem solved by revealing unused applications are considered. It is achieved by analyzing a specific user device usage historical data inside geographical system environment.

*Keywords: intellectual system, memory optimization, clusterization, adaptation, geographical systems.*

**Введение.** Современные геоинформационные системы (ГИС) представляют из себя системы, решающие множество задач в зависимости от типа, поставщика данных, целевой группы пользователей и других факторов [1]. Объединяет их стремление запуска на мобильных и других устройствах с ограниченными системными ресурсами: память, процессор, дисковое пространство. Мобильные устройства более требовательны к ресурсам так как это напрямую влияет на сохранение заряда батареи, а в некоторых случаях и на пользовательский опыт использования [2]. В данной статье мы предлагаем метод, который можно использовать для построения модуля для ГИС, который позволяет следить за использованием внутренних подпрограмм ГИС (далее просто приложений) и классифицировать их на две группы: «нужные», «ненужные» на основании исторических данных использования всех приложений и текущем окружении рабочей среды. Предполагается, что ГИС должна иметь модульную архитектуру и существует возможность выгружать из памяти приложения независимо от других. Также следует отметить, что в системе должен присутствовать модуль, который сохраняет исторические данные, к которым будет осуществляться доступ метода.

Определение предпочтений пользователя в определенном окружении рабочей среды, которое далее мы будем называть контекстом, является одной из проблем создания электронных персональных ассистентов и рекомендательных и других подобных систем [3]. Добыча в той или иной форме пользовательских предпочтений могут сделать программы электронных

персональных ассистентов ближе к пользователю и более персонализированными, в то время как рекомендательные системы могут быть более адаптивными к конкретному пользователю. Конечным результатом также может служить выборка более релевантной информации, включая рекламу, адаптацию пользовательского интерфейса, изменения информационных потоков в зависимости от контекста и других.

В результате, такого рода персонализация применима к решению проблемы высвобождения ресурсов устройства путем выгрузки из памяти (закрытию) приложений, которые были прежде установлены в ГИС, но не используются пользователем в достаточной степени. Следует отметить, что в большинство распространенных операционных систем уже встроены те или иные механизмы [4] решения данной проблемы, но абсолютное большинство из них основаны на различных методах (например, вытесняющая многозадачность) [5], но не учитывают предпочтения самого пользователя.

**Цель работы.** Создание метода, основанного на сборе статистики использования приложений в рабочей среде ГИС и последующего определения набора приложений, которые наиболее важны для пользователя в определенном контексте с целью дальнейшей выгрузки их из памяти и высвобождению ресурсов системы тех, которые не являются релевантными для конкретного пользователя в определенном контексте.

**Общий алгоритм работы модуля оптимизации.** Общий алгоритм функционирования модуля оптимизации памяти можно разбить на следующие этапы:

1. Начало работы модуля;
2. Определение параметров рабочей области;
3. Пересчет параметров модели на основе исторических данных;
4. Вычисление ранга для каждого из приложений на основе предложенной далее модели;
5. Отделение множества (классификация) «ненужных» приложений от «нужных» в данный момент путем описанным далее;
6. Принятие решения о выгрузке из памяти приложений, которые вошли в множество «ненужных»;
7. Завершение работы модуля;

**Рабочая область.** Будем считать, что рабочая область строится в соответствии со следующей моделью. Обозначим через  $J = \{j_1, j_2, \dots, j_k\}$  множество приложений ГИС [6]. Множество пользователей  $U = \{u_1, u_2, \dots, u_c\}$ . Тогда рабочая область есть подмножество приложений, отобранных последовательностью запросов  $Q_h^u, h = \overline{0, N}$  клиентом  $u$ . Каждый запрос  $Q_h^u$  - это вызов приложения пользователем.

**Параметры модели.** Каждый параметр модели очевидно имеет свою собственную, отличную от других параметров размерность и природу. Поэтому все параметры модели приводятся к одной размерной шкале при помощи функции softmax (1).

$$v_j^* = \frac{e^{v_j}}{\sum_k e^{v_k}} \quad (1)$$

Где  $v_j$  - значение параметра для  $j$ -того приложения,  $k$  – количество приложений.  $v^*$  - исправленное значение переменной, которая может принимать значение любой переменной из множества  $L$  (2).

$$L_j = \{d_j, f_j, m_j\} \quad (2)$$

Где  $d_j$ - отражает коэффициент длительности работы пользователя с  $j$ -ым приложением,  $f_j$  - коэффициент частоты запуска приложения пользователем,  $m_j$  - коэффициент потребления памяти приложением.

**Модель вычисления ранга приложения.** Искомый ранг приложения обозначим как  $w$ . Данный ранг будет отражать насколько важным является приложение в определенном контексте: чем больше число, тем выше вероятность того, что пользователь будет использовать это приложение. Мы можем предположить, что на повышение ранга может положительно повлиять фактор интереса пользователя к данному приложению [7], а именно частота запуска и количество проведенного в нем времени, что можно выразить как  $d_j \times f_j$ . В тоже время нельзя забывать о системных ресурсах, которые потребляет приложение. В некоторых случаях будет оптимальней оставить в памяти небольшие приложения [8], чтобы получить более быстрый доступ к ним, что является спорным моментом. В итоге интерес пользователя очевидно, должен повышать рейтинг приложения для текущего контекста, а совокупный коэффициент потребления памяти является значительным понижающим фактором и модель  $w_j$  будет иметь вид (3).

$$w_j = \frac{d_j \times f_j}{m_j} \quad (3)$$

**Классификация приложений на «нужные» и «ненужные».** После того как мы получили множество  $W$ , которое будет содержать набор рангов для каждого приложения, мы можем классифицировать приложения на два класса:  $A$  – «нужные приложения»,  $B$  – «ненужные». Эти же обозначения будем использовать в качестве отображения соответствующих подмножеств.

Для классификации существует множество методов, но в данном случае мы будем использовать кластеризацию с помощью метода k-means – так как  $W$  является одномерным множеством можно обойтись максимально простым алгоритмом вычисления. Предварительно мы можем сделать подмножества более отделимыми друг друга (4).

$$w_j^* = \sqrt{\log_e w_j + 1} \quad (4)$$

После применения алгоритма кластеризации к множеству  $W^*$  мы получим два множества (кластера)  $X$  и  $Y$ . Для определения принадлежности этих множеств к классам  $A$  и  $B$  воспользуемся простым алгоритмом:

1. *Начало алгоритма*

2. Если  $\sum_n w_i^{*X} > \sum_m w_l^{*Y}$ ,  $i = \overline{0 \dots n}$ ,  $l = \overline{0 \dots m}$ , то  $A = X$ ,  $B = Y$ ;

3. Если  $\sum_n w_i^{*X} < \sum_m w_l^{*Y}$ ,  $i = \overline{0 \dots n}$ ,  $l = \overline{0 \dots m}$ , то  $A = Y$ ,  $B = X$ ;

4. Если  $\sum_n w_i^{*X} = \sum_m w_l^{*Y}$ ,  $i = \overline{0 \dots n}$ ,  $l = \overline{0 \dots m}$ , то множества считаются не отделимыми;

5. *Конец алгоритма.*

**Вывод.** Данный метод позволяет оптимизировать системные ресурсы, используемые ГИС путем выявления не используемых приложений (модулей) и последующей их деактивации или активацией в зависимости от контекста. Также повышается полезность рабочей области конкретного пользователя путем минимизации избыточности, обзор и оценка которой является предметом для последующей работы.

## ЛИТЕРАТУРА

1. Беляков С.Л., Боженьюк А.В., Гинис Л.А., Герасименко Е.М. Нечеткие методы управления потоками в геоинформационных системах. – Таганрог: Изд-во ЮФУ. – 2013. – 176 с.

2. Aitor Agirre, Jorge Parra, Aintzane Armentia, Elisabet Estévez, Marga Marcos, "QoS Aware Middleware Support for Dynamically Reconfigurable Component Based IoT Applications", International Journal of Distributed Sensor Networks, vol. 2016, pp. 1, 2016, ISSN 1550-1329.

3. Розенберг И.Н. Спутниковые и геоинформационные технологии в интеллектуальных системах управления // Железнодорожный транспорт. – 2013. – No 3. – С. 28-32.

4. S. H. Bokhari "The linux operating system" Computer vol. 28 no. 8 pp. 74-79 Aug. 1995.

5. Cheng-Zen Yang Bo-Shiung Chi "Design of an Intelligent Memory Reclamation Service on Android" Technologies and Applications of Artificial Intelligence (TAAI) 2013 Conference on pp. 97 6–8 Dec. 2013.

6. Belyakov S.L., Belykova M.L., Glushkov A.A. Visualization of Spatial Data Adaptive to Changes in the Database Structure // Proc. 2nd International Conference on Information Science and Security ICISS 2015, pp. 185-187, 2015.

7. Ковалев С.М., Гуда А.Н., Бутакова М.А. Гибридная стохастическая модель обнаружения особых типов паттернов в темпоральных данных // Вестник Ростовского государственного университета путей сообщения. – 2013. – No 3. – С. 36-42.

8. Miklós Ajtai et al. "A sieve algorithm for the shortest lattice vector problem" STOC pp. 601-610 2001.