

- Barnes, R. Neural networks for ischemic stroke / R. Barnes, J. Toole, J. Nelson, V. Howard // Journal of stroke and cerebrovascular diseases. – Vol. 15. – No. 5. – 2006. – P. 223–227.
- Shanthi, D. Input feature selection using hybrid neuro-genetic approach in the diagnosis of stroke disease / D. Shanthi, Dr.G. Sahoo, Dr.N. Saravanan // IJCSNS International Journal of Computer Science and Network Security. – Vol. 8. – No. 12. – 2008. – P. 99–107.
- Shanthi, D. Designing an artificial neural network model for the prediction of thrombo-embolic stroke / D. Shanthi, Dr.G. Sahoo, Dr.N. Saravanan. // International journal of biometric and bioinformatics. – Vol. 8. – No.1. – 2008. – P. 10–18.
- McNeill, A. How Accurate Are Primary Care Referral Letters For Presumed Acute Stroke? // Scottish Medical Journal. – V. 53. – № 4. – 2008. – P. 11–12.
- Мастыкин, А.С. Нейросетевой подход к решению проблемы предотвращения атак на нормальное кровоснабжение мозга // А.С. Мастыкин, В.В. Евстигнеев, В.А. Головко, Е.Н. Апанель, Г.Ю. Войцехович // Доклады Академии наук Беларуси. – 2010. – Т.54 – № 5. – С. 81–90.

Материал поступил в редакцию 30.01.12

GOLOVKO V.A., VAITSEKHOVICH H.U., MASTYKIN A.S., APANEL E.N. Neural networks for diagnostics transient ischemic attack

In this paper the neural network model for transient ischemic attacks recognition have been addressed. The proposed approach is based on integration of the NPCA neural network and multilayer perceptron. The dataset from clinic have been used for experiments performing. Combining two different neural networks (NPCA and MLP) it is possible to produce efficient performance in terms of detection and recognition transient ischemic attacks. The main advantages of using neural network techniques are the ability to recognize "novel" TIA attack instances, quickness and ability to assist the doctor in making decision.

УДК 004.89

Гречка А.В., Головко В.А.

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ФОРМИРОВАНИЯ РАСПИСАНИЯ РАБОЧИХ СМЕН ДЛЯ ОПЕРАТОРОВ ЦЕНТРА ОБСЛУЖИВАНИЯ ВЫЗОВОВ

Введение. Основной задачей теории расписаний (ТР) является разработка методов синтеза расписаний работы обслуживающих систем. Одно из наиболее актуальных применений теории расписаний – календарное планирование и регулирование производственных процессов.

Теория расписаний довольно специфична и имеет ряд особенностей, в отличие от других наук, занимающихся сходными проблемами. Во-первых, для неё характерно огромное разнообразие теоретических моделей, вытекающее из разнообразия реальных моделируемых процессов, так как проблема оптимальной организации протекающих во времени процессов имеет глобальный характер и возникает практически во всех сферах человеческой деятельности. Второй особенностью является комбинаторная природа исследуемых моделей и решаемых оптимизационных задач, которая обуславливает высокую комбинаторную сложность их решения. И, в-третьих, для теории расписаний характерна высокая актуальность решаемых задач, ввиду их ярко выраженной прикладной направленности.

При всём этом теория расписаний занимается вполне теоретическими проблемами, разработкой наиболее общих и универсальных методов решения возникающих в ней оптимизационных задач.

В большинстве ситуаций оптимальное расписание может быть найдено в результате перебора конечного множества возможных вариантов. Основное затруднение состоит в том, что число таких вариантов обычно оказывается исключительно большим и растёт, по меньшей мере, экспоненциально с ростом размерности задачи. Среди задач теории расписаний можно выделить полиномиально разрешимые и NP-сложные. Для каждой полиномиально разрешимой задачи известен по крайней мере один эффективный алгоритм ее решения, трудоёмкость которого ограничена сверху некоторым полиномом от длины записи исходной информации задачи. Для NP-сложных задач такие алгоритмы неизвестны и, по всей видимости, не существуют [4].

Алгоритмы построения расписаний без проведения полного или частичного перебора вариантов являются решающими эвристическими правилами и играют важную роль в прикладной теории расписаний. Однако эвристические алгоритмы основаны на приеме, который называется «снижением требований». Он заключается в отказе от поиска оптимального решения и нахождения вместо этого «хорошего решения» за приемлемое время. Методы, применяемые для построения алгоритмов такого типа, сильно зависят от специфики задачи [3]. Универсального алгоритма не существует.

Использование только математических методов теории

расписаний в большинстве случаев неприменимы для прикладных задач, так как имеют ряд недостатков: такие модели не позволяют учитывать многие факторы, влияющие на ход производства, которые для разных предприятий могут быть индивидуальны; использование только математических методов ограничивается невозможностью быстрого реагирования на возникающие ситуации, требующие немедленной корректировки планов и т.д. [2].

Для исключения недостатков подобных разработок возможно использовать эволюционные методы, которые позволили бы получать оптимальные решения проблем реальных производственных ситуаций за малое время. При решении такие методы рассматривают систему планирования как чёрный ящик, когда на входе задаются различные значения параметров планирования, после чего оценивается эффективность получаемых расписаний с точки зрения ключевых показателей эффективности. На основе анализа существующих разработок в области эволюционных методов перспективным решением сложных комбинаторных задач оптимизации является гибридное использование генетического и муравьиного алгоритмов. Это позволит существенно улучшить систему оперативного планирования, тем самым сократив время получения оптимальных или приемлемых производственных расписаний [2].

Описание работы. Основная идея работы – создание интеллектуальной системы формирования расписания рабочих смен для операторов центра обслуживания вызовов (ЦОВ). При составлении расписания необходимо решить 2 проблемы: с одной стороны, оптимизировать работу ЦОВа как системы массового обслуживания, с другой стороны расписание рабочих смен как минимум должно удовлетворять нормам трудового законодательства, кроме того, необходимо учесть ряд субъективных факторов, которые определяют «хорошее» для операторов расписание.

Данная система состоит из 3-х подсистем, поэтапно решающих последовательность задач:

- Прогноз нагрузки на ЦОВ: определяем предполагаемое количество вызовов в каждые полчаса указанного месяца.
- Формирование карты смен: упорядочиваем во времени определенное количество смен таким образом, чтобы наиболее оптимально покрыть предполагаемую нагрузку.
- Составление расписания: назначаем операторов (телефонистов) на смены так, чтобы минимизировать сумму и дисперсию численно выраженных нежелательных факторов (для телефонистов) расписания.

Гречка А.В., магистрант кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

Физика, математика, информатика

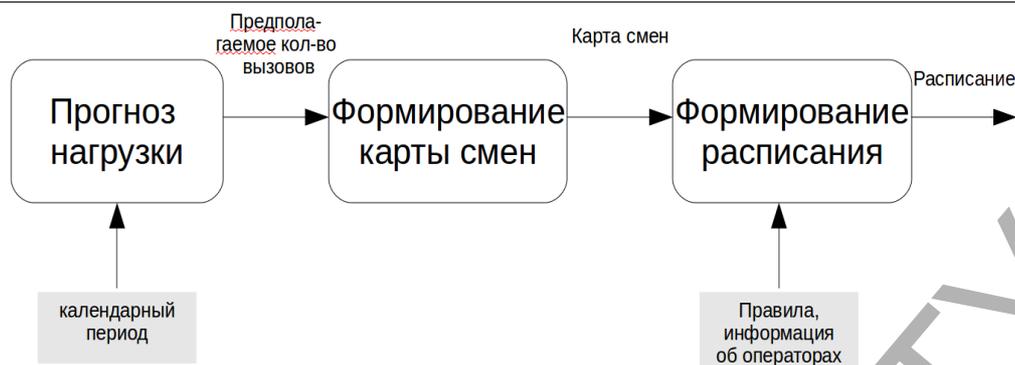


Рис. 1. Схема работы систем

Прогноз нагрузки. На первом этапе необходимо спрогнозировать нагрузку на ЦОВ, т.е. получить предполагаемое количество вызовов в каждые полчаса заданного месяца. Интенсивность заявок (вызовов) зависит от следующих факторов: время суток, день недели, месяц, является ли день выходным или рабочим и т.д. Для решения этой задачи можно использовать нейронную сеть как метод, доказавший свою эффективность для решения подобных задач. Многослойная нейронная сеть способна осуществлять любое отображение входных векторов в выходные. Архитектура такой сети состоит из множества слоев нейронных элементов[1].

Структура НС

Для прогнозирования нагрузки используется многослойный персептрон с одним скрытым слоем. На вход подается вектор нормализованных значений:

1. Месяц, $x_1 = X_1/12$, где $X_1 = (1, \dots, 12)$
2. День месяца, $x_2 = X_2/31$, где $X_2 = (1, \dots, 31)$
3. День недели, $x_3 = X_3/7$, где $X_3 = (1, \dots, 7)$
4. Время суток (номер получаса) $x_4 = X_4/48$, где $X_4 = (1, \dots, 47)$
5. Рабочий или выходной $x_5 = \{0, 1\}$; $x_5 = 1$ – рабочий день, $x_5 = 0$ – выходной (или праздничный) день, для субботы $x_5 = 0.6$.

Скрытый слой содержит 20 нейронов, используется сигмоидная функция активации.

На выходе НС получаем предполагаемое количество вызовов в данный получас месяца (используется линейная функция активации). Выходные значения также нормализованы: $Y = \{0, \dots, 1\}$, где Y – выходное значение НС; $calls = Y * 1500$.

Значение «1500» обусловлено статистическими данными и принято как максимально возможное количество обработанных вызовов в полчаса (максимальное количество поступивших вызовов на ЦОВ с 2005 по 2010 год = 1380 вызовов, и в данном случае учтена тенденция к росту количества обслуженных вызовов из года в год).

Для обучения использовался метод градиентного спуска (алгоритм обратного распространения ошибки). В качестве обучающей выборки были использованы статистические данные за 2010 год, т.е. размер обучающей выборки равен количеству получасов в год ≈ 17520 эталонных образов.

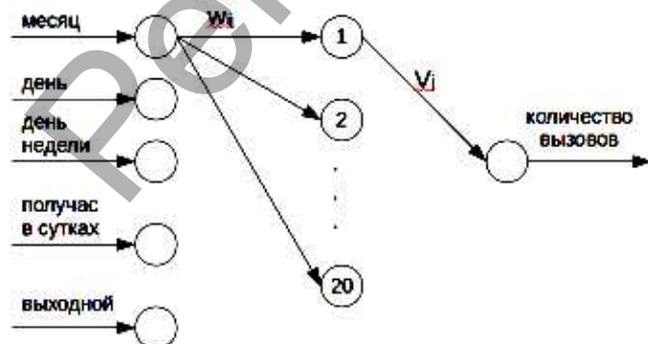


Рис. 2. Нейронная сеть

Результаты работы подсистемы прогнозирования нагрузки изображены на рисунках 3, 4. На графиках изображены 5 видов кривых: forecast – спрогнозированная нагрузка с помощью искусственной нейронной сети; остальные кривые получены путем анализа статистических данных:

- a) last year – данные за прошлый год (в ту же дату в прошлом году);
- b) adjusted – данные за прошлый год с учетом дней недели (в ближайшую дату в прошлом году в такой же день недели);
- c) stats – среднестатистические данные с учетом дней недели за 3 предыдущих года;
- d) fact – фактические данные за месяц (день), на который прогнозируем нагрузку.

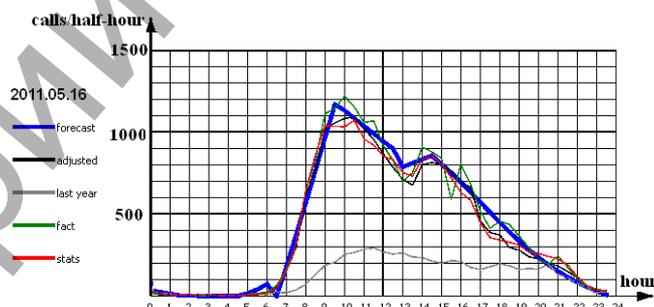


Рис. 3. Прогноз нагрузки на ЦОВ на 16.05.2011 (понедельник) E=47,636

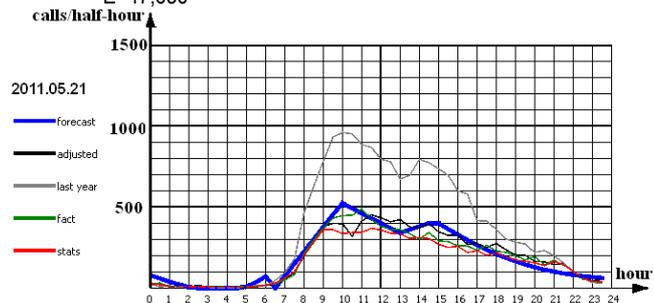


Рис. 4. Прогноз нагрузки на ЦОВ на 21.05.2011 (суббота) E=52,749

Формирование карты смен. На следующем этапе необходимо сформировать карту смен таким образом, чтобы наиболее оптимально покрыть предполагаемую нагрузку. Необходимо, чтобы среднее время ожидания вызовов в каждые полчаса находилось в рамках заданного (нормативного) диапазона 32–37 сек.

Существует нелинейная зависимость между количеством вызовов и количеством обрабатывающих узлов (операторов, работающих в данные полчаса) при заданном диапазоне времени ожидания вызовов. Эту функциональную зависимость можно получить, проанализировав статистические данные.

Ниже на рисунке 5 изображена выборка статистических данных за 2007–2010 гг. – количество телефонистов и количество обработанных вызовов в полчаса при среднем времени ожидания вызовов (32; 37) сек.

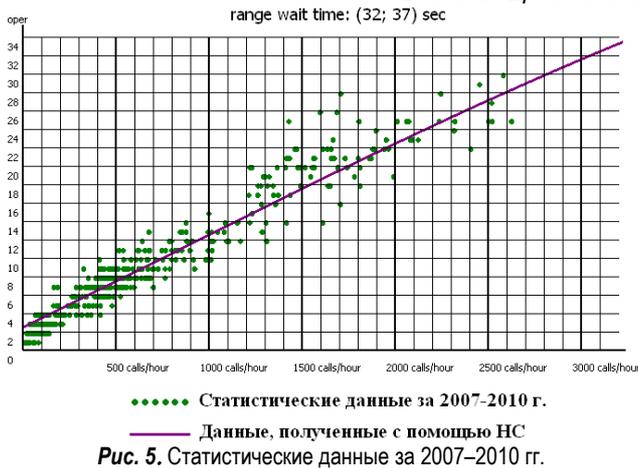


Рис. 5. Статистические данные за 2007–2010 гг.

Для аппроксимации данных была использована НС с одним скрытым слоем. На вход такой сети подается количество вызовов, на выходе получаем количество операторов, необходимое для их обработки. Структура НС изображена на рисунке 6.

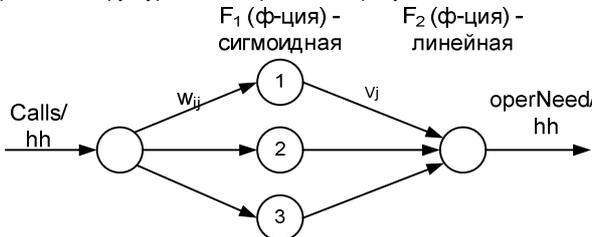


Рис. 6. Структура НС для аппроксимации данных

Таким образом, производим своего рода маппинг – пересчет одних входных данных в другие, т.е. пересчитываем предполагаемое количество вызовов в необходимое количество операторов в каждые полчаса.

Теперь задача сводится к тому, что необходимо упорядочить смены с временем дискретизации полчаса (т.е. каждые полчаса может начинаться одна или несколько новых смен) таким образом, чтобы значение количества операторов, работающих в каждые полчаса, было приближено к рассчитанным значениям.

Здесь необходимо учитывать, что существует статическая или константная (заданная пользователем) часть расписания. В эту часть расписания должны входить ночные и разрывные (перерыв равен 4 часа) смены. Оставшееся количество смен должно быть упорядочено наиболее оптимальным образом с 6.30 до 23.00 часов (с 23.00 до 6.30 длится ночная смена). Таким образом, диапазон значений начала смен, которые необходимо упорядочить, равен (13, 31) получасам, т.е. они должны начинаться с 6.30 до 15.30 часов (т.к. для того, чтобы семичасовая смена с перерывом полчаса закончилась в 23.00, она должна начаться в 15.30). В таком случае, изменение параметров карты смен в одном дне не влияет на характеристики карты смен в другом дне.

Задача формирования карты смен на один день относится к комбинаторным (NP-сложным) задачам. Для ее решения используем генетический алгоритм как метод адаптивного перебора в пространстве поиска решений.

В данном случае особью будет являться определенная карта смен на один день. Хромосома (генетический код) – числовой массив, однозначно характеризующий данную карту смен – массив блоков, содержащих значения начала смен. Начало смены в данном контексте – это номер получаса в сутках, в котором она начинается, остальные параметры – длительность, перерыв и т.д. – одинаковые для всех смен.

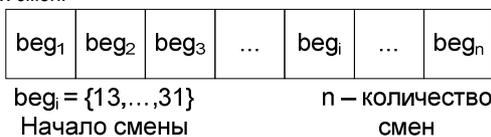


Рис. 7. Хромосома

Функция приспособленности – дисперсия коэффициентов пропорциональности необходимого и получившегося количества операторов, работающих в каждые полчаса.

$$fitness = \sigma = \frac{1}{n} \sum_{i=1}^n (K_i - \bar{K})^2, \quad (1)$$

где *fitness* – функция приспособленности данной особи (хромосомы);

σ – дисперсия искомым величин;

n – количество получасов в сутках;

i – номер получаса в сутках;

K_i см. формулу 2;

\bar{K} – среднее значение K_i ;

$$K_i = \frac{OpNeed_i}{Op_i}, \quad (2)$$

где K_i – коэффициент пропорциональности необходимого и получившегося количества операторов в *i*-ый полчаса;

$OpNeed_i$ – необходимое (рассчитанное) количество операторов для нагрузки в *i*-ый полчаса;

Op_i – количество операторов, работающих в *i*-тый полчаса при данном расписании.

Сам генетический алгоритм работает стандартным образом:

1. Случайная инициализация популяции особей. Вычисление функций приспособленности.
2. Репродукция.
3. Мутация.
4. Селекция.

Схема работы данной подсистемы выглядит следующим образом.

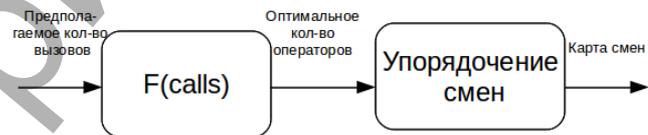


Рис. 8. Схема работы подсистемы формирования карты смен

Для более наглядного представления результатов работы генетического алгоритма на рисунках приведены графики значений (рассчитанного для минимизации среднего времени ожидания в сети и получившегося) количества операторов в единицу времени.

Результатом работы подсистемы является карта смен. Пример карты смен на один день изображен на рисунке 11.

Расписание. Полноценное расписание кроме карты смен подразумевает расстановку операторов по сменам.

Как отмечалось выше, при составлении расписания для операторов кроме трудового законодательства учитываются другие субъективные факторы. Например, желательно, чтобы у оператора было не больше одной ночной смены в месяц, после ночной смены, как правило, должно стоять два выходных или хотя бы один; желательно, чтобы у каждого оператора в расписании хотя бы одна пара выходных совпала с субботой, воскресеньем и т.д.

Таких субъективных факторов, определяющих «хорошее» расписание довольно большое количество, более того, они могут конфликтовать друг с другом, и степень их важности может быть четко выявлена только в процессе эксплуатации данной. Поэтому хорошим решением было описать эти факторы в виде правил на специально разработанном метаязыке и использовать их (правила) в качестве входных данных для подсистемы формирования расписания.

Правило – арифметическое выражение, вычисляющее метрику на основании параметров смены или смен. У каждого правила есть свой весовой коэффициент, который определяет степень значимости правила при формировании расписания. Правила состоят из констант, переменных и операторов. Переменными описываются



Рис. 9. Данные при случайной инициализации расписания



Рис. 10. Данные после работы генетического алгоритма 14.05.2011 суббота

Кол-во смен: 24 (всего 688)

	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23																				
23.00-6.30	6																			273																		
6.30-14.00		7	8	9	10	11	12	13	14											630																		
7.00-14.30		7	8	9	10	11	12	13	14											678																		
7.00-14.30		7	8	9	10	11	12	13	14											678																		
7.30-15.00			8	9	10	11	12	13	14	15										683																		
7.30-15.00			8	9	10	11	12	13	14	15										683																		
7.30-15.00			8	9	10	11	12	13	14	15										703																		
8.00-15.30			8	9	10	11	12	13	14	15										703																		
8.00-15.30			8	9	10	11	12	13	14	15										703																		
8.30-16.00				9	10	11	12	13	14	15	16									676																		
8.30-16.00				9	10	11	12	13	14	15	16									676																		
8.30-19.30				9	10	11	12	13	14	15	16	17	18	19						708																		
8.30-19.30				9	10	11	12	13	14	15	16	17	18	19						666																		
9.00-16.30				9	10	11	12	13	14	15	16	17	18	19	20					666																		
9.00-20.00				9	10	11	12	13	14	15	16	17	18	19	20					706																		
9.00-20.00				9	10	11	12	13	14	15	16	17	18	19	20					706																		
9.30-17.00				9	10	11	12	13	14	15	16	17	18	19	20					704																		
9.30-17.00				9	10	11	12	13	14	15	16	17	18	19	20					702																		
10.00-17.30				10	11	12	13	14	15	16	17	18	19	20						668																		
10.30-18.00				10	11	12	13	14	15	16	17	18	19	20	21					643																		
14.00-21.30										15	16	17	18	19	20	21	22			617																		
15.30-23.00										16	17	18	19	20	21	22	23			617																		
15.30-23.00										16	17	18	19	20	21	22	23			617																		
23.00-6.30																			23	279																		
Количество вызовов:	58	2	143	265	428	572	716	862	992	860	819	778	738	698	622	644	666	689	673	627	576	525	473	422	373	326	283	243	206	173	143	116	93	72	54	38		
Количество	2	1	4	7	10	13	16	19	20	19	18	17	16	14	15	15	16	15	14	13	12	11	10	9	8	7	6	5	5	4	3	3	2	2	2			
Количество	1	1	3	6	8	12	15	17	17	16	17	15	14	13	14	14	15	11	11	11	11	12	10	8	7	6	6	4	4	3	3	2	1	1	1			
Удельная нагрузка	58	2	47	47	53	47	47	50	53	50	51	45	49	49	47	46	47	53	61	57	52	43	47	52	46	46	47	40	51	43	35	38	31	36	58	38		

Рис. 11. Пример карты смен на один день

характеристики смены: начало, перерыв, день и т.д. У каждого оператора (телефониста) свой массив смен, в котором указание элемента (в данном случае смены) осуществляется как по абсолютному индексу (день), так и по относительному (0:сегодня, -1:вчера и т.д.). Если смена в данный день отсутствует, храним специальную смену «выходной», это такая смена, которая начинается в 0.00 и ее длительность равна 0 часов. Ниже приведен пример правила.

$$1000 * (\text{смена}[0].\text{начало} - \text{смена}[-1].\text{конец} < 14)$$

Правило означает, что между сменами должно быть не менее 14 часов, и если это условие не выполняется, увеличиваем метрику текущего оператора на 1000. В данном случае число «1000» играет роль весового коэффициента правила, т.е. определяет степень его значимости при составлении расписания.

При составлении расписания для каждой смены оператора применяем набор правил и суммируем метрики, нормализуем суммар-

ные метрики, т.е. делим на количество часов оператора. Целью будет являться такое расписание, которое обладает минимальной суммой и дисперсией метрик.

Оптимальное расписание формируется на основании карты смен, информации об операторах и правил.

Решение такой задачи можно разделить на 2 этапа: расстановку операторов по дням и расстановку операторов по сменам.

1 этап: расстановка людей по дням. Необходимо определить, какие операторы в какие дни будут работать, не указывая конкретные смены. На этом этапе учитываем только правила, в которых не указаны конкретные параметры смены, т.е. используем понятия «выходной день», «рабочий день», «ночная смена» (ночная смена на этом этапе отличается от обычной тем, что занимает два календарных дня).

Для того, чтобы начать процесс оптимизации, т.е. минимизации суммы и дисперсии метрик, необходимо сформировать первоначальную

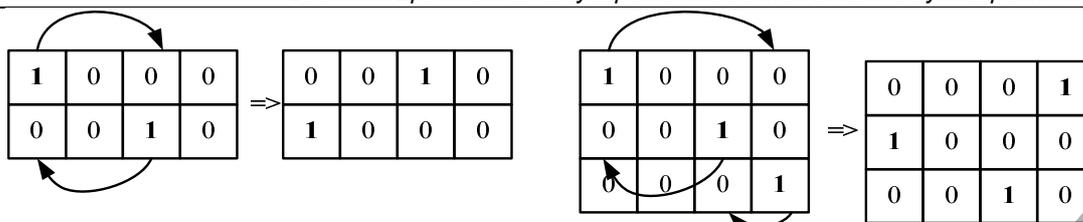


Рис. 12. Замена смен

расстановку по дням, без учета правил. Иначе можно сказать, формируем бинарную матрицу M с ограничениями размером $n \times m$, где n – количество дней в месяце, m – количество операторов. $M_{ij}=1$ означает, что у i -того оператора есть смена в день j . Ограничениями являются: сумма единиц в строке (количество смен в месяц у оператора), сумма единиц в столбце (количество смен в день), маска отпусков или желаемых выходных (элементы матрицы, которые не могут принимать единичные значения), маска фиксированных смен (смены, заданные пользователем, элементы матрицы, которые не могут принимать нулевое значение) и др. Сам процесс оптимизации основан на заменах рабочих смен операторов, т.е. меняем операторов местами, если это возможно. Замена возможна, если она не меняет заданные параметры матрицы. Примеры таких замен изображены на рисунке 12.

Для того, чтобы ускорить алгоритм поиска оптимальной расстановки по дням подобными заменами, находим для каждого оператора смену (смены) – факторы метрик, т.е. смены, которые максимально влияют (или просто влияют) на метрику. И подобными свопами (см. рисунок 12) меняем ее.

Описание алгоритма:

- 1) считаем метрики для каждого оператора;
- 2) находим оператора с максимальной метрикой;
- 3) находим смену-фактор оператора, т.е. такую смену, без которой метрика уменьшилась наиболее существенно;
- 4) находим все возможные варианты замен данной смены;
- 5) находим наилучшую замену (при которой сумма метрик будет минимальной);
- 6) если при наилучшей замене сумма метрик меньше исходной, меняем операторов местами;
- 7) если есть хотя бы одна замена, при которой сумма метрик уменьшится, переходим к пункту 1, иначе алгоритм завершается.

2 этап: расстановка операторов по сменам. Аналогично первому этапу применяем правила, вычисляющие метрики. На этом этапе учитываем все правила, в том числе те, в которых указаны конкретные параметры смены.

Т.к. мы уже имеем предварительную расстановку операторов по дням, то задачу можно представить следующим образом: в каждый день месяца, на который составляем расписание, есть n операторов и n смен, существует $n!$ (количество перестановок из n элементов) возможных решений составления расписания на один день. Необходимо из всех возможных комбинаций расписаний на каждый день при возможных перестановках операторов по сменам найти такую, при которой сумма и дисперсия метрик была бы минимальной.

Алгоритм оптимизации на 2-ом этапе работает примерно аналогичным образом как и на 1-ом: формируется первоначальная раскладка операторов по сменам, затем методом замен (но операторы меняются сменами в пределах одного дня) минимизируем метрики.

Ниже перечислены правила, которые были использованы для тестирования.

1) Правила первого порядка (учитываются только при предварительной расстановке по дням):

$$1000 * (\text{hasWeekend} == 0)$$

Правило обозначает, если у оператора отсутствует хотя бы одна пара выходных суббота-воскресенье, увеличиваем метрику на 1000.

$$10000 * ((\text{shift}[0].\text{isDayOff} + \text{shift}[-1].\text{isDayOff} + \text{shift}[-2].\text{isDayOff} + \text{shift}[-3].\text{isDayOff} + \text{shift}[-4].\text{isDayOff} + \text{shift}[-5].\text{isDayOff}) == 0)$$

Если у оператора шесть подряд рабочих дней, увеличиваем метрику на 10000.

$$4000 * (\text{shift}[-2].\text{isNightShift} * (\text{shift}[0].\text{isDayOff} != 1))$$

Если у оператора после ночной смены сразу рабочий день, увеличиваем метрику на 4000.

$$10 * (\text{shift}[-3].\text{isNightShift} * \text{shift}[-1].\text{isDayOff} * (\text{shift}[0].\text{isDayOff} != 1));$$

Если у оператора после ночной смены только один выходной день, увеличиваем метрику на 10.

2) Правила второго порядка (учитываются при формировании готового расписания):

$$255 * (\text{shiftsCount}(13, 13) > 1) * (\text{shiftsCount}(13, 13) - 1);$$

Правило обозначает: если у оператора в расписании на данный месяц количество смен, которые начинаются в 13 полчаса в сутках (т.е. в 6.30 утра) больше одного, увеличиваем метрику на 255, умноженное на разность количества таких смен и 1.

$$266 * (\text{shiftsCount}(31, 31) > 2) * (\text{shiftsCount}(31, 31) - 2);$$

Правило обозначает: если у оператора в расписании на данный месяц количество смен, которые начинаются в 31 полчаса в сутках (т.е. в 15.30) больше двух, увеличиваем метрику на 266, умноженное на разность количество таких смен и 2.

$$500 * (\text{Max}(14, 30) > 3) * (\text{Max}(14, 30) - 3);$$

Правило обозначает: если у оператора в расписании на данный месяц количество одинаковых смен (с одинаковым началом), начало которых лежит в диапазоне от 14 до 30 полукасов (т.е. 7.00- 15.00), больше 3, увеличиваем метрику на 500, умноженное на разность количество таких смен и 3. Данное правило формализует понятие: «желательно, чтобы у операторов в расписании разных смен было примерно поровну»

$$999 * (\text{shift}[0].\text{hhBegin} - \text{shift}[-1].\text{hhEnd} < 28) * (28 - (\text{shift}[0].\text{hhBegin} - \text{shift}[-1].\text{hhEnd})) * (\text{shift}[0].\text{len} != 0) * (\text{shift}[-1].\text{len} != 0);$$

Правило обозначает: если у оператора от начала текущей смены до конца предыдущей прошло меньше 28 полукасов (14 часов), увеличиваем метрику на 999, умноженное на разность 28 и межсменного перерыва (в полукасах), при условии что обе смены не являются «выходными». Напомним, что, если у оператора смена в определенный день отсутствует, то храним специальную смену, длительность которой равна 0.

В результате работы подсистемы формирования расписания на май 2011 года при первоначальной раскладки по дням сумма метрик была равна 880020; после алгоритма оптимизации – 0. На втором этапе формирования расписания рабочих смен при первоначальной раскладке смен сумма метрик равна 6453, после алгоритма оптимизации – 255.

Часть готового расписания изображена ниже на рисунке 4.7 (красным выделены смены, которые увеличивают метрику, т.е. отрицательно влияют на расписание).

Результаты приведены на рисунке 13.

Более подробная схема работы всей системы изображена ниже на рисунке 14.

Заключение. Данная работа была направлена на решение конкретной прикладной задачи – автоматизация процесса составления расписания рабочих смен для операторов ЦОВа брестского филиала РУП «Белтелеком».

Май 2011	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	смен	часов	metric1 0	metric2 255					
МАРЦИШКО В.М.	В	В	9:00	9:00	15:30	В	7:30	9:00	9:00	13:30	В	6:30	15:30	23:00	0:00	В	В	8:30	8:00	9:00	В	В	9:00	7:00	8:00	7:30	14:00	В	В	10:30	8:00	20(20)	139(139)	0	0					
НАЗАРЧУК А.Б.	15:30	В	7:00	7:00	6:30	8:00	14:30	В	8:00	8:00	9:00	10:30	В	23:00	0:00	В	В	9:30	10:30	В	В	7:00	7:00	7:00	8:00	10:00	В	В	9:30	8:00	20(20)	139(139)	0	0						
НИКОЛАЕНКО Т.П.	В	В	6:30	7:30	7:30	7:30	9:30	В	7:30	8:00	8:30	9:30	В	15:30	23:00	0:00	В	В	12:30	В	В	7:30	7:00	8:00	7:30	10:30	В	В	10:30	8:00	20(20)	139(139)	0	0						
НОВИК Л.П.	В	В	10:00	14:00	14:00	15:30	В	8:30	9:00	14:00	В	7:30	9:00	15:30	23:00	0:00	В	В	9:30	8:00	6:30	8:00	6:30	8:00	7:30	13:30	В	В	9:00	7:00	7:30	7:30	20(20)	139(139)	0	0				
ГРИЦУК О.А.	В	В	7:30	7:30	7:30	7:30	10:00	В	8:30	8:00	8:30	10:00	В	7:30	7:30	15:30	23:00	0:00	В	В	7:00	7:00	8:30	8:00	10:00	В	В	15:30	15:30	20(20)	139(139)	0	0							
ОСИПЕНКО Н.В.	В	В	9:00	15:00	15:00	8:00	14:30	15:00	В	8:00	8:00	8:00	9:30	В	7:00	7:30	15:30	23:00	0:00	В	В	15:00	8:30	9:30	15:00	16:00	В	В	6:30	7:00	7:00	7:00	20(20)	139(139)	0	0				
ПАНАСЮК О.В.	В	В	12:00	8:00	15:30	В	7:30	9:00	15:00	В	7:00	9:00	15:30	В	15:30	15:00	23:00	0:00	В	В	15:30	23:00	0:00	В	В	8:30	7:00	8:00	12:00	14:00	20(20)	139(139)	0	0						
ПЛЕЧКО Д.Л.	В	В	7:30	15:30	13:00	15:00	В	7:30	7:00	8:00	7:00	9:30	В	7:00	7:30	7:00	7:30	8:00	В	В	23:00	0:00	9:30	В	В	7:00	6:30	7:00	7:00	6:30	20(20)	139(139)	0	0						
РОМАНЮК В.В.	В	В	16:00	19:00	В	16:00	15:00	16:30	В	В	В	В	16:30	17:30	22:30	В	14:30	17:00	23:00	21:30	В	17:00	17:30	В	23:00	0:00	5:30	В	В	14:00	15:30	22:00	22:00	20(20)	139(139)	0	0			
РЮБАШЕВСКАЯ Е.В.	8:30	8:30	8:00	8:00	10:30	В	7:30	8:00	8:00	8:30	10:30	В	7:30	7:30	9:00	15:30	В	9:00	8:30	9:00	15:30	В	15:30	23:00	0:00	16:30	В	В	В	В	18(18)	125(125)	0	255						
САМУСИК Л.Г.	В	В	9:00	12:00	13:00	В	12:30	В	В	В	В	6:30	7:30	7:30	7:30	9:00	В	11:30	15:30	15:00	16:30	В	19:30	23:00	В	16:30	21:30	В	В	15:00	15:30	23:00	0:00	6:30	В	В	20(20)	139(139)	0	0
СЕМЕНЮК Е.А.	15:30	23:00	В	6:30	7:00	7:00	8:30	13:30	В	В	В	15:00	В	В	В	В	В	В	В	В	В	В	В	В	В	В	В	В	В	В	В	В	В	В	8(8)	55(55)	0	0		
СКОМОРОХ М.И.	В	В	16:00	13:00	В	15:30	14:30	19:00	В	В	В	В	14:30	14:30	15:00	14:00	16:00	В	20:30	22:30	В	17:30	17:00	В	23:00	20:30	В	В	17:30	16:00	20(20)	139(139)	0	0						
СОБОЛЕВСКАЯ И.И.	9:00	8:30	В	8:00	8:00	12:30	В	7:30	6:30	7:30	8:00	8:30	В	7:00	7:30	7:30	8:00	8:30	В	14:30	15:00	15:00	14:30	16:00	В	21:30	В	В	15:00	23:00	23:00	0:00	6:30	В	В	20(20)	139(139)	0	0	
СТЕЦКАЯ Ж.В.	В	В	9:00	9:30	15:00	В	6:30	8:30	9:00	9:30	10:30	В	7:30	7:30	12:30	14:00	В	15:00	15:00	20:00	21:30	В	В	В	В	В	В	В	В	В	В	В	В	В	В	13(13)	90(90)	0	0	

Рис. 13. Готовое расписание

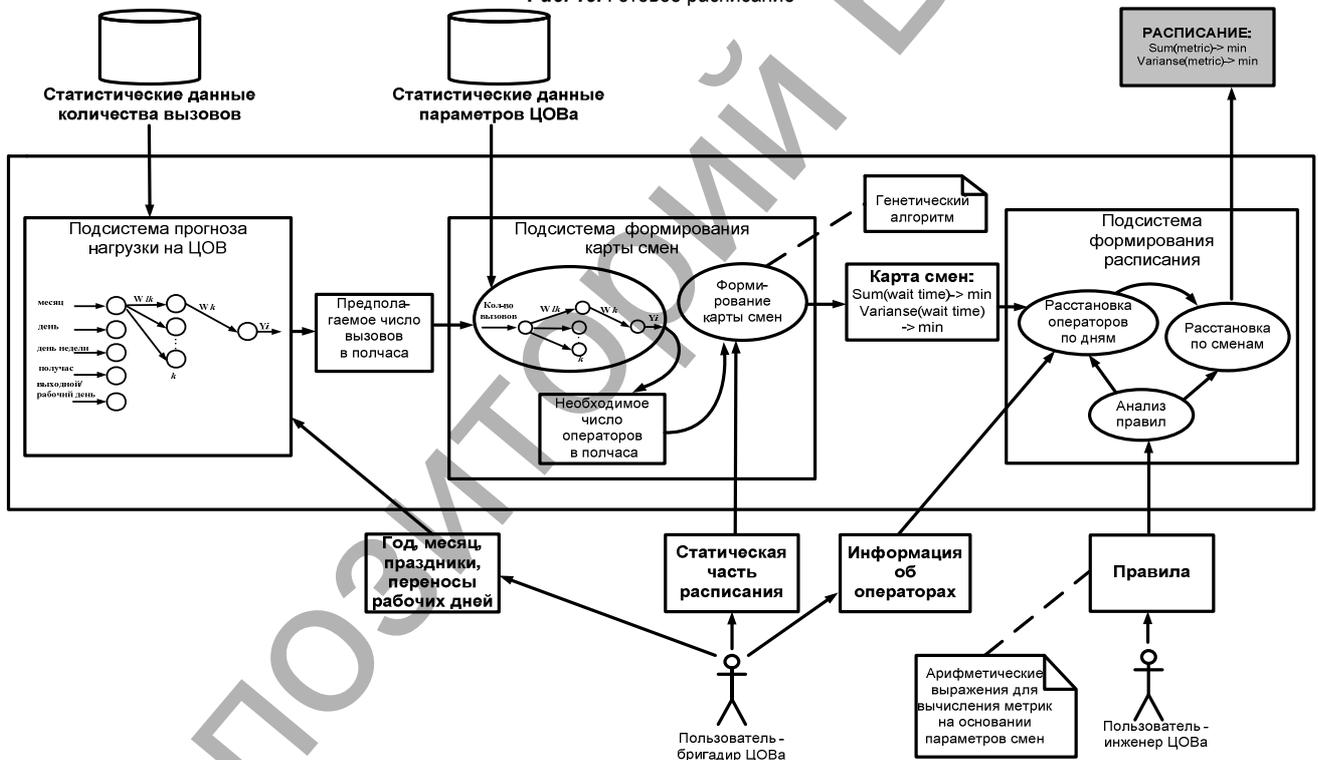


Рис. 14. Подробная схема работы системы

В результате работы над проектом была спроектирована и реализована интеллектуальная система формирования расписания. При проектировании были учтены все особенности и специфика задачи. Были изучены особенности и параметры функционирования центра обслуживания вызовов как системы массового обслуживания, с одной стороны, и как объекта функционирования трудовых ресурсов, с другой стороны, а также была изучена система ограничений и правил составления расписаний такого типа.

Данная система позволяет значительно ускорить и оптимизировать процесс составления расписания, при этом она является довольно «гибкой» к условиям ограничений, которые необходимо учесть при составлении расписания.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Головкин, В.А. Нейроинтеллект: теория и применение. Книга 1 Организация и обучение нейронных сетей с прямыми и обратными связями. – Брест, 1999.
2. Кузьменко, В.М. Анализ современных методов искусственного интеллекта применительно к задачам календарного планирования единичного производства / В.М. Кузьменко, С.В. Таран. – М., 1999.
3. Люгер, Дж. Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем. – 4-е изд. – М.: Вильямс, 2005. – 9 с.
4. Танаев, В.С. Теория расписаний. Многостадийные системы / В.С. Танаев, Ю.Н. Сотсков, В.А. Струевич – М.: Наука, 1989. – 328 с.

Материал поступил в редакцию 15.12.11

HRECHKA A.V., GOLOVKO V.A. The intelligent system to creation schedule for operators of Calls Service Center

In this paper the intelligent system to creation schedule for operators of Calls Service Center have been addressed. This system forecasts load (estimated amount of calls per time scale) based on input data (calendar period), calculates optimal number of operators and creates shift map. Then it is created resulting schedule according to the information about shifts and operators.

УДК 004.5;621.38

Бутов А.А.

ПРОСТОЙ МЕТОД НАХОЖДЕНИЯ БУЛЕВОЙ ФОРМУЛЫ МНОГОУГОЛЬНИКА В ДИЗЬЮНКТИВНОЙ НОРМАЛЬНОЙ ФОРМЕ

Введение. Традиционные способы описания геометрических объектов основаны на использовании методов вычислительной геометрии [1] и имеют практическое применение, например, в системах автоматизированного проектирования (САПР) топологии интегральных схем [2, 3]. Однако в последнее время появились альтернативные способы описания, основанные на использовании булевых формул: в работе [4] предложен метод нахождения булевой формулы многоугольника в скобочной форме, в работе [5] – в более привычной дизъюнктивной нормальной форме. Однако для алгоритма из [5], как и для многих алгоритмов вычислительной геометрии, характерно наличие проблемы вычислительной точности. Она заключается в том, что хотя теоретически строго обоснована правильность работы алгоритма, однако на практике встречаются задачи, для которых алгоритм не работает или работает некорректно в силу ограниченной точности представления вещественных чисел в памяти компьютера и потери точности в промежуточных вычислениях. Практика разработки САПР в области проектирования топологии интегральных схем показала, что такой эффект возникает, в частности, если в алгоритме используется операция разбиения стороны топологического объекта на отрезки, которые далее используются в качестве операндов.

В настоящей работе предлагается достаточно простой и приемлемый на практике метод нахождения булевой формулы многоугольника в дизъюнктивной нормальной форме, который основан на использовании двух простых операций: вычисления угла между прямыми и проверки факта принадлежности вершин многоугольника его выпуклой компоненте, что снимает указанную выше проблему вычислительной точности.

Основные определения, постановка задачи. Многоугольник, расположенный на плоскости, задается своей *границей* – замкнутой не пересекающейся ломаной линией, состоящей из отрезков прямых или *сторон* многоугольника. Эту границу можно определить последовательностью *угловых точек* или *вершин* многоугольника, получаемых при обходе его по границе справа: p_1, p_2, \dots, p_n (рис. 1, где $n = 10$).

Так как каждая пара соседних угловых точек ограничивает соответствующую сторону многоугольника, то его границу можно задать также последовательностью сторон многоугольника: s_1, s_2, \dots, s_n , где $s_1 = (p_1, p_2), s_2 = (p_2, p_3), \dots, s_n = (p_n, p_1)$.

Вершина p_1 , которая служит начальной точкой для последовательного обозначения отрезков, образующих границу многоугольника, называется *начальной*. В качестве начальной будем выбирать вершину, наиболее удаленную от координатной оси X (если таких вершин несколько, то среди них выбирается вершина, наиболее удаленная от координатной оси Y).

Вершина многоугольника называется *крайней*, если через нее можно провести прямую, не пересекающуюся ни с одним из отрезков границы за исключением двух соседних отрезков, на стыке которых эта вершина располагается. На рисунке 1 таких вершин шесть, и они отмечены жирными кружками. Начальная вершина p_1 , как следует

из указанного выше способа ее выбора, всегда принадлежит множеству крайних вершин многоугольника.

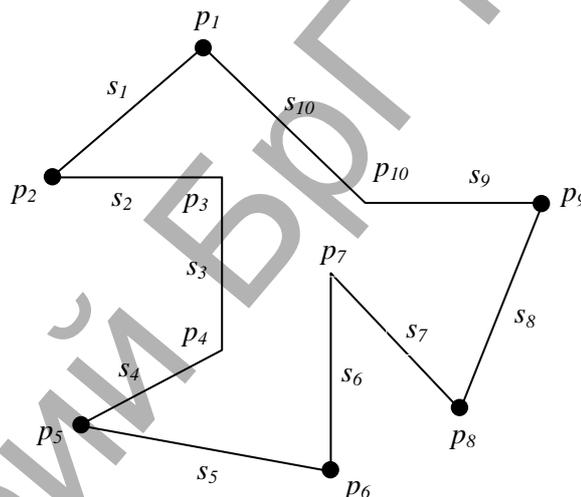


Рис. 1. Угловые точки и стороны многоугольника

Каждой стороне s_i многоугольника поставим в соответствие *ориентированную прямую* v_i , содержащую точки p_i и p_{i+1} . Положим, что она ориентирована от p_i к p_{i+1} .

Рассмотрим некоторую произвольную точку плоскости p , заданную парой декартовых координат (X, Y) . Будем считать, что точка p расположена *слева от прямой* v_i если она принадлежит полуплоскости, расположенной слева от ориентированной прямой v_i , или лежит на прямой v_i . Все возможные варианты левостороннего расположения точки p относительно ориентированной прямой v_i представлены на рисунке 2 (последние два варианта соответствуют случаю, когда прямая v_i параллельна координатной оси X).

Как и в работе [4], будем в дальнейшем обозначать отрезки ломаной буквами a, b, c, \dots , границу многоугольника – как $abc\dots$, а полуплоскости, расположенные слева от соответствующих ориентированных прямых – буквами A, B, C, \dots (считая, что каждая из этих полуплоскостей включает в себя еще и все точки порождающей ее ориентированной прямой). Введем также предикаты a, b, c, \dots для описания положения некоторой точки p на плоскости, полагая, что $a(p) = 1$, если и только если $p \in A$.

Основываясь на таких предикатных переменных, в работе [4] описан метод построения скобочной булевой формулы F , представляющей многоугольник и обладающей следующим свойством: если выполнить подстановку предикатных координат произвольной точки плоскости, то формула F примет значение 1 в случае, когда точка принадлежит данному многоугольнику, и значение 0 – в противном случае.

Аналогичная задача решается и в работе [5], однако булева формула F там строится в дизъюнктивной нормальной форме (ДНФ).

Бутов А.А., к.т.н., доцент кафедры экономической кибернетики Белорусского государственного университета информатики и радиоэлектроники.

Беларусь, БГУИР, 220013, г. Минск, ул. П. Бровки, 6.

Физика, математика, информатика