

внимания, что, в свою очередь, снижает психологическую нагрузку от необходимости частого переключения контекста, а также повысит своевременность получения нужных сведений – от прогноза погоды и загруженности транспортных магистралей до любых отслеживаемых пользователем событий.

Список цитированных источников

1. Echo & Alexa - Amazon Devices [Электронный ресурс]. – Режим доступа: <https://www.amazon.com/Amazon-Echo-And-Alexa-Devices/b?ie=UTF8&node=9818047011>. – Дата доступа: 08.04.2018.
2. Low, J. Two-Way Mirrors [Электронный ресурс] – Режим доступа: <http://www.jimloy.com/physics/mirror0.htm> – Дата доступа: 11.07.2012.
2. Калачев, А. Высокопроизводительные многоядерные процессоры для встраиваемых систем // Компоненты и технологии. – № 2. – 2010. – С. 92–102.
3. Справочник по OpenCV с примерами кода [Электронный ресурс] – Режим доступа: <https://opencv-tutorial.ru/> – Дата доступа: 08.04.2018.
4. Viola, P. Robust real-time face detection / P. Viola, M. J. Jones // International Journal of Computer Vision. – Vol. 57. – No. 2. – 2004. – P.137–154.
5. Viola, P. Rapid Object Detection using a Boosted Cascade of Simple Features / P. Viola, M. J. Jones // Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001.
6. Буслюк, В. В. О возможностях применения в вузах аутентификации на основе персональных устройств / В. В. Буслюк, Д. А. Костюк, Д. И. Кульбеда, Н. А. Терешкевич, В. А. Юхно. // Свободное программное обеспечение в высшей школе : материалы XIII конференции / Переславль, 26–28 января 2018 года. – М.: Альт Линукс, 2018. – С. 45–49.

УДК 519.863 + 004.588

Николаев М. В.

Научные руководители: ст. преподаватель Хомицкая Т. Г., ст. преподаватель Лизун Л. В.

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ О НАЗНАЧЕНИЯХ С ИСПОЛЬЗОВАНИЕМ ТЕОРЕМЫ ХОЛЛА

Введение. Нередко в реальной жизни возникает проблема, когда необходимо сопоставить объекты одного типа объектам другого типа. Например, будучи лидером некоторой группы людей, необходимо назначать по одному человеку, ответственному за определённую работу. При этом известна оценка, как каждый человек из группы справляется с различными работами. Речь идёт про задачу о назначениях. Автором разработано приложение, которое призвано обучить пользователя решению этой задачи на базе теоремы Холла. Приложение демонстрирует работу алгоритма, предоставляя информацию об этапах и порядке выполнения решения, позволяет быстро найти все решения задачи и проверить качество усвоения пользователем метода решения.

Постановка задачи о назначениях. Группа из n станков A_1, A_2, \dots, A_n должна выполнять n видов работ B_1, B_2, \dots, B_n . Эффективность использования i -го станка на j -й работе определяется мерой эффективности c_{ij} , где $i, j=1, 2, \dots, n$. Найти оптимальную расстановку станков по видам работ (т. е. такую расстановку, при которой суммарная эффективность назначения окажется наибольшей).

Алгоритм решения и демонстрация работы программы

Теорема Ф. Холла о существовании системы различных представителей. Система $M(S) = \{S_1, S_2, \dots, S_n\}$ имеет систему различных представителей

тогда и только тогда, когда для любой подсистемы $\{S_{i_1}, S_{i_2}, \dots, S_{i_k}\} \subseteq M(S)$ выполняется неравенство: $|\bigcup_{j=1}^k S_{i_j}| \geq k$, т. е. количество элементов в объединении любых подмножеств должно быть не менее k [1].

Алгоритм состоит в подборе чисел u_i и v_j , и опирается на следующее утверждение:

Пусть $A = [a_{ij}]$, $i, j=1, \dots, n$ – матрица целых чисел. Тогда

$$\max_{\pi} \sum_{i=1}^n a_{i\pi(i)} = \min_{u_i + v_j \geq a_{ij}} \left(\sum_{i=1}^n u_i + \sum_{j=1}^n v_j \right).$$

Минимум суммы $(\sum_{i=1}^n u_i + \sum_{j=1}^n v_j)$ достигается на перестановке π такой, что $u_i + v_{\pi(i)} = a_{i\pi(i)}$, $\forall i=1, \dots, n$.

Начинаем решение задачи с поиска чисел v_j , u_i по следующему правилу:

u_i выбирается как максимальный элемент по строке i . v_j заполняется нулями. Будем выполнять построение u_i и v_j в виде векторов, присоединённых справа и снизу к матрице эффективностей соответственно (рисунок 1).

	A1	A2	A3	A4	A5	U_i
B1	6	4	3	3	2	6
B2	8	8	5	8	6	8
B3	16	14	16	17	9	17
B4	9	8	7	7	3	9
B5	14	11	10	12	6	14
V_j	0	0	0	0	0	

Рисунок 1 – Присоединение векторов U_i и V_j к матрице эффективностей

Затем заполняется матрица совпадений по следующему правилу: ставим «*» в те позиции, где для элементов матрицы ценностей и найденных u_i и v_j выполняется равенство: $u_i + v_j = a_{ij}$ (рисунок 2).

Рисунок 2 – Заполнение матрицы совпадений

	A1	A2	A3	A4	A5
B1	*				
B2	*	*		*	
B3				*	
B4	*				
B5	*				

По матрице совпадений строятся множества совпадений. В множество совпадений S_i добавляем номер столбца j , если элемент матрицы совпадений в i -й строке и j -м столбце отмечен знаком «*» (рисунок 3).

	Сет 1	Сет 2	Сет 3
S[1]	1		
S[2]	1	2	4
S[3]	4		
S[4]	1		
S[5]	1		

Рисунок 3 – Заполнение множества совпадений

Затем выясняется, имеют ли множества совпадений систему различных представителей. Для этого проверяется выполнимость условий теоремы Холла. Чтобы система множеств имела систему различных представителей, необходимо (по теореме Холла), чтобы для любого объединения, которое можно составить из этих множеств, количество элементов, попавших в объединение, было не меньше, чем число множеств, образующих объединение.

При этом построение объединений начинается с самого наибольшего, и при каждом построении проверяется условие теоремы Холла. Если ни на одном из объединений условия теоремы Холла не нарушаются, то система различных представителей найдена (она определяется по матрице совпадений) и работа алгоритма заканчивается.

Если же найдётся объединение, где условие теоремы Холла нарушается, дальнейшее объединение строить не нужно. В этом случае необходимо на одну и ту же величину (например, на единицу) уменьшить числа u_i и увеличить числа v_j по следующему правилу: если множество S_i участвовало в объединении, уменьшаем число u_i ; если элемент j входит в объединение, то увеличиваем число v_j .

После изменения чисел u_i и v_j повторяются этапы построения матрицы совпадений, множеств совпадений, проверки условия теоремы Холла и изменения чисел u_i и v_j до тех пор, пока условия теоремы Холла не будут выполнены.

Выполнение условий теоремы Холла означает, что система множеств имеет систему различных представителей, и работа алгоритма закончена. По

	B1	B2	B3	B4	B5
1 Combination	5	4	3	2	1
2 Combination	5	2	4	3	1
3 Combination	3	5	4	2	1
4 Combination	1	5	3	2	4
5 Combination	2	5	4	3	1

последней матрице совпадений можно записать все возможные назначения станков на выполнение работ и подсчитать суммарную эффективность этих назначений (рисунок 4).

Рисунок 4 – Результат работы программы: назначение механизмов на работы

Процесс реализации приложения. Приложение разрабатывалось при помощи Qt – кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++. Разработка приложения проходила в рамках совмещения процедурного и объектно-ориентированного подхода. Основной алгоритм программы вынесен в отдельный модуль и представляет собой набор функций для обработки данных в визуальных формах. Пользовательский интерфейс реализован в виде набора классов, каждый из которых представляет одно или несколько окон и реализует обработку данных только для этих окон, однако может обмениваться данными с остальными окнами (классами).

Приложение имеет следующие окна:

- Главное окно программы;
- Окно инициализации исходных данных;
- Окно результата;
- Окно сохранения данных;
- Окно загрузки данных;
- Окно вопросов (для режима контроля пользователя).

Приложение имеет 3 режима работы. Обычный режим работы предназначен для быстрого и эффективного решения задачи и демонстрации решения пользователю. Режим обучения пользователя предназначен для пошагового выполнения алгоритма с последовательным появлением таблиц с решением задачи (рисунки 1 – 4). Каждый этап алгоритма выполняется в главном окне программы и сопровождается пояснениями, появляющимися в текстовом поле в левой части окна. Таблицы также появляются на главном окне, дополняя друг друга но не перекрывая, поэтому у пользователя нет необходимости воз-

вращаться назад по алгоритму. Режим «контроль» предназначен для проверки знаний пользователя и включает в себя как теоретические вопросы по алгоритму, так и его практическое выполнение (заполнение таблиц, описанных ранее, вручную), и дальнейшую проверку введенных данных. По окончании режима контроля пользователю предоставляется окно результата, содержащее информацию о количестве вопросов, ошибок, попыток ввода и времени прохождения режима контроля.

Кроме основных функций, программа даёт возможность сохранить исходные данные в виде файла в выбранной директории или загрузить предварительно сохранённый файл с данными.

Описание используемых алгоритмов. Данные хранятся в виде объекта (модели), содержащего двумерный массив. Простая обработка на первом этапе имеет сложность $O(n^2)$ (при поиске максимального элемента, построении матрицы и множества совпадений).

Для проверки выполнимости теоремы Холла используется построение всех возможных сочетаний без повторений. Сложность этого участка алгоритма определяется формулой:

$$C_n^k = \frac{n!}{(n-k)! * k!}$$

где n – размерность квадратной матрицы, а k – число объединяемых множеств, в худшем случае изменяется от n до 2 (этот этап алгоритма прерывается, как только находится объединение множеств, которое нарушает условия теоремы Холла). Отсюда видно, что сложность $O(n!)$.

Для поиска элементов на каждой итерации используется бинарный массив, где 1 в позиции i означает, что множество с индексом i будет участвовать в объединении, а 0 – не будет.

Последний этап алгоритма заключается в поиске среди отмеченных элементов множеств всех возможных назначений механизмов на работы и заключается в рекурсивном вызове функции назначения.

Пусть n – размерность матрицы эффективности, а индекс i содержится во множестве совпадений n . Это означает, что механизм под номером n может быть назначен на работу под номером i . Запомним номер i , и теперь будем искать назначение для $n-1$ механизма на работу с номером $j \neq i$, не допуская назначения механизму $n-1$ работы i , так как её уже выполняет механизм под номером n . Сложность этого участка алгоритма в худшем случае $O(n^n)$, однако реально, благодаря предварительным этапам алгоритма, она сводится к произведению размеров каждого из множеств совпадений:

$$O\left(\prod_1^n m_j\right)$$

где m_j – размер множества совпадений S_j . Зачастую число m_j близко к 1, и всегда натуральное.

Ниже представлен фрагмент программного кода (рисунок 5), реализующий вышеописанный алгоритм назначения механизмов на работы при помощи рекурсии.

Cur_union – номер механизма, которому ищем работу.

Sys_representers – массив, содержащий множества представителей (можно считать обычным двумерным массивом).

Result – массив, содержащий все возможные назначения механизмов на работы.

Index – номер строки в таблице результатов.

