УДК 004:001.891.57

С.В. Пивнева[1], Н.В. Лада[2]

[1] *Тольяттинский государственный университет, Тольятти, Россия*
[2] *Черкасский государственный технологический университет, Черкассы, Украина*

ОСОБЕННОСТИ ПРИМЕНЕНИЯ МАТЕМАТИЧЕСКОГО АППАРАТА МИНИМИЗАЦИИ НЕДЕТЕРМИНИРОВАННЫХ КОНЕЧНЫХ АВТОМАТОВ В ОЦЕНКЕ ЭФФЕКТИВНОСТИ УПРАВЛЕНИЯ СОЦИОИНЖЕНИРИНГОМ

В работе рассмотрены особенности применения математического аппарата минимизации недетерминированных конечных автоматов в оценке эффективности управления социоинжинирингом, основанного на сочетании комбинаторных и эвристических методов оптимизации.

Ключевые слова: *социоинжениринг, недетерминированный конечный автомат (НКА), эвристические алгоритмы, минимизации НКА.*

Введение

Постанова проблемы. С началом формирования информационного общества, стало ясно, что для многих управленческих задач нет достаточно простого математического аппарата, позволяющего обрабатывать полученные социальные данные, определять наиболее значимые из них и производить оценку эффективности управления социумом. Поэтому важными стали вопросы исследования различны математических методов, в том числе, вопросы их экономного представления в памяти компьютера.

В представленной статье автором рассматривается возможность применения математического аппарата минимизации недетерминированных конечных автоматов в оценке эффективности управления социоинжинирингом.

Недетерминированным конечным автоматом (НКА) называется пятёрка вида: $K = (Q, \Sigma, \delta, S, F)$, где Q – некоторое конечное множество состояний (вершин), Σ – рассматриваемый алфавит, δ – функция переходов вида $\delta: Q\times(\Sigma\cup\{\varepsilon\})\rightarrow 2^Q$ (ε – пустое слово, 2^Q – булеан множества Q), $S\subseteq Q$ – множество стартовых состояний (входов), $F\subseteq Q$ – множество финальных состояний (выходов).

Подобные автоматы являются простейшим примером так называемых распознавателей, которые используются в различных областях теории формальных языков. НКА находят широкое применение в программах обработки больших массивов текста и распознавания речи, при построении лексических анализаторов, описании и верификации всевозможных систем, состоящих из конечного числа состояний с заданными переходами между ними (например, коммуникационных протоколов) и т.д.

Анализ последних исследований и публикаций. Проведя анализ последних исследований и публикаций, следует отметить [1, 2], что понятие «социоинжения» появилось достаточно давно, 20-40-е года прошлого столетия. Термин «социоинженирия» или «инжениринг» определяется как социальное регулирование и контроль различных организационных структур для решения социальных задач. Прикладные исследования в этой области проводятся в различных направлениях, в частности в политике, менеджменте и т.п. [2]. Однако в настоящее время не существует математического аппарата для построения логической структуры отбора наиболее значимых параметров управления, а также отсутствуют методы определения эффективности управления социоинженирингом. Именно поэтому решение данной задачи является практически необходимым для дальнейшей разработки теоретических основ методологии оценки качества социоинжениринга.

Цель статьи состоит в определении особенностей применения математического аппарата минимизации недетерминированных конечных автоматов в оценке эффективности управления социоинженирингом.

Основной материал

Варианты минимизации недетерминированных конечных автоматов и используемые эвристики. Наиболее полно теория минимизации НКА изложена в [3 – 5]. В данной статье рассмотрим основные принципы. При разных способах представления НКА более важными могут быть либо минимальное число вершин, либо минимальное число дуг. Соответственно приходится рассматривать различные варианты эвристических алгоритмов минимизации НКА, а именно – *вершинную* и *дуговую* минимизацию. Кроме того, рассматривается ещё один вариант минимизации – *звёздно-высотная*, т.е. задача построения конечного автомата, имеющего среди всех ему эквивалентных минимальную вложенность операции «звезда Клини» [6].

Используемый подход к минимизации НКА (предложенный в [5, 7] для более широкого класса задач дискретной оптимизации) основан на сочетании комбинаторных и эвристических методов оптимизации и применим с соответствующими изменениями во всех трёх случаях. Однако основные принципы рассмотрим для задачи вершинной минимизации НКА.

Важнейшей вспомогательной подзадачей при вершинной минимизации НКА является следующая. Задана прямоугольная матрица, заполненная элементами 0 или 1. Некоторую пару подмножеств строк и столбцов назовём блоком, если, во-первых, на всех их пересечениях стоят 1, и, во-вторых, это множество нельзя пополнить ни строкой, ни столбцом – без нарушения первого свойства. Т.н. *допустимым решением* является множество блоков, покрывающих все элементы 1 заданной матрицы. Требуется выбрать допустимое решение, содержащее минимальное число блоков – которое в данном случае будет *оптимальным* решением.

Рассмотрим, например, матрицу, изображенную на рис. 1. В ней имеются следующие 5 блоков:

$$\alpha=\{A,B,C,D\}\times\{U\},\quad \beta=\{A,C,D\}\times\{Z,U\},$$
$$\gamma=\{B,C,D\}\times\{X,U\},\quad \delta=\{C,D\}\times\{X,Z,U\},$$
$$\omega=\{D\}\times\{X,Y,Z,U\}.$$

Для покрытия всех значений 1 данной матрицы достаточно использовать 3 из этих 5 блоков: β, γ и ω.

	X	Y	Z	U
A	0	0	1	1
B	1	0	0	1
C	1	0	1	1
D	1	1	1	1

Рис. 1. Матрица исходных данных

В первую очередь отметим, что эвристические алгоритмы автором статьи понимаются как т.н. anytime-алгоритмы, т.е. алгоритмы реального времени, которые в каждый определённый момент работы имеют лучшее на данный момент решение. Для создания алгоритмов применяем незавершённый метод ветвей и границ (МВГ) [3].

Эвристикой, непосредственно относящейся к задаче вершинной минимизации НКА, является выбор разделяющего элемента для МВГ. Это пример жадной эвристики (несколько упрощая ситуацию, можно сказать, что мы стремимся выбрать блок с относительно большим числом символов 1, не вошедших в уже выбранные блоки). Сами разделяющие элементы генерируются динамически, в процессе работы «основного» МВГ, а для их генерации применяется «вспомогательный» МВГ. Такой подход позволяет не строить все множество разделяющих элементов заранее, что делает возможным его применение для минимизации НКА *больших размерностей*.

Пусть рассматривается клетка матрицы с координатами (i, j), пусть также R(i) – количество знаков 1 (или #) в строке i, а C(j) – количество знаков 1 в столбце j. Тогда:

1. Для каждой пары (i,j), содержащей #, посчитаем значение

$$F(i,j) = G(H(R(i)),H(C(j))), \tag{1}$$

где G(x,y) и H(x) – некоторые функции. Эти функции выбираются с помощью генетических алгоритмов (ГА), причем вид функции H(x) заранее неизвестен. А про функцию G(x,y) известно лишь то, что она является возрастающей. Таким образом, подбор параметров, необходимых для описания этих функций, есть объект самообучения.

2. При самообучении мы обычно используем следующий конкретный вид функции H(x):

$$H(x)= e^{-(P1X)} + P_2N(x-1)^{-1} + P_3\, x + 1, \tag{2}$$

где N – соответствующая размерность таблицы соответствий состояний. Отметим, что здесь возникает аналогия с результатами, полученными нами при самообучении игровых программ: при начале самообучения с функциями произвольного вида, т.е. функциями, не имеющими вид (2), результат работы ГА даёт функции вида, близкого к указанному. Однако вид (2) всё же был априори подобран эвристически.

3. Среди всех блоков выбираем такой (**R**,**C**), для которого т.н. итоговая рейтинговая сумма $\sum F(R_i,C_j)$, где $R_i\in\mathbf{R}$ и $C_j\in\mathbf{C}$, будет наибольшей. Ещё раз отметим, что функция F выбирается согласно (1), а функции G и H при этом получаются в результате самообучения методами ГА. Помещаем выбранный блок в итоговое множество.

4. Специальным образом помечаем элементы, входящие в выбранный на шаге 3 блок, чтобы они не учитывались при вычислении $F(R_i,C_j)$, но могли входить в любой вновь выбираемый блок. Итак, в будущем, т.е. при повторном выполнении шагов 1–3, эти ячейки уже не считаем помеченными 1.

5. Если остались еще ячейки, не вошедшие ни в один блок из итогового множества, то возвращаемся на этап 1. Заметим, что блоки, которым принадлежит хотя бы одна ячейка, не имеющая соседей по вертикали либо по горизонтали, обязательно будут включены в искомое множество, т.к., согласно виду функции F, их рейтинговая сумма равна бесконечности.

Полученное множество блоков считается искомым и используется либо для проведения очередного шага т.н. турнирного самообучения, либо для проведения очередного шага МВГ [8]. Каждый раз при выборе разделяющего элемента мы принимаем решение на основе нескольких вариантов (выдаваемых на основе работы нескольких вспомогательных эвристик) и каждый из них количественно оценивается несколькими т.н. предикторами. А итоговое решение о единственном разделяющем элементе (блоке) может быть принято на основе работы любого алгоритма многокритериальной оптимизации; мы же принимаем решение на основе т.н. «игровой» эвристики – с помощью т.н. функций риска [9].

Ещё одну применяемую эвристику можно считать специальной модификацией эвристики локального поиска, более того – её связи с незавершённым МВГ. Опишем суть этой эвристики очень кратко. После нахождения некоторого текущего псевдооптимального решения задачи мы можем получить ещё одно текущее решение путём применения некоторой эвристики локального поиска. Получив это новое решение, мы фактически имеем для него последовательность разрешающих элементов. Эту последовательность можно прервать – причём на любом уровне; прервав её, мы получаем очередную подзадачу, дальнейшее решение которой с приемлемо большой вероятностью может привести к «хорошему» варианту, т.е. к улучшению текущего псевдооптимального решения.

Теперь кратко опишем применение для незавершённого МВГ наших версий алгоритмов кластеризации ситуаций (более подробно см. в [3]). Кластеризация в первую очередь проводится на множестве подзадач – для того, чтобы после реализации одного шага МВГ для решения некоторой подзадачи применять выбор того же самого разделяющего элемента (решение подзадач «по аналогии»). Для применения обычных алгоритмов кластеризации нам необходимо на множестве подзадач выбрать метрику – и это делается следующим образом.

Пусть X и Y – некоторые множества, $n=|X \cap Y|$ – число элементов их пересечения, $N=|X \cup Y|$ – число элементов их объединения. Тогда будем писать $\Omega(X,Y)=1-n/N$. Рассмотрим эвристику для определения метрики на множестве блоков подзадач. Пусть P_1 – множество клеток матрицы первой подзадачи, имеющих значение 1; P_2 – то же для второй подзадачи. Тогда в качестве метрики используем значение $\Omega(P_1,P_2)$.

Алгоритм минимизации НКА.

Сформулируем теперь окончательно алгоритм, описывающий предлагаемый подход к задаче вершинной минимизации НКА. По терминологии, принятой в современной литературе по алгоритмизации, его можно назвать схемой для разработки алгоритмов.

(инициализация списка подзадач одним элементом - исходной задачей)

(«большой» шаг) loop {

1-й «малый» шаг – выбор очередной задачи из списка подзадач;

для неё – выбор разделяющего элемента; а для него – реализация одного шага метода ветвей и границ. Отметим, что сам алгоритм реализации выбора разделяющего элемента для одного шага МВГ является отдельной сложной задачей и уже был описан выше. Также отметим, что во время этого шага одновременно формируется описанная выше т.н. последовательность правых подзадач (ППЗ; см. [3, 5]).

2-й «малый» шаг связан с предметом кластеризации ситуаций;

подробное описание реализуемого алгоритма кластеризации ситуаций приведено в наших предыдущих публикациях, а упрощённое описание может быть дано следующим образом.

Ранее при формировании левой задачи мы отмечаем, какой именно правой задаче она «родственна» (т.е. отмечаем пару левая-правая); при этом если в будущем какое-то решение (о разделяющем элементе) выбирается для одной из них – то, по возможности, то же самое мы делаем и в другой. Весь этот вспомогательный алгоритм работает очень быстро (поскольку разделяющий элемент уже выбран): если этот элемент имеется в «парной» задаче - мы его и выбираем; если же его в «парной задаче» нет – то при этом мы тоже ничего не теряем. Все подобные действия также связаны со своим алгоритмом формирования ППЗ.

3-й «малый» шаг –

генерация очередного потенциального разделяющего элемента (блока). В данном случае (в отличие от 1-го «малого» шага) этот разделяющий элемент пока не является конкретным разделяющим элементом ни для какой подзадачи; сам алгоритм соответствующей генерации тоже, конечно, сильно зависит от конкретной задачи. Этот шаг (в отличие от следующего, 4-го «малого» шага) выполняется с помощью «жадных» алгоритмов и случайного выбора – и обязательно даёт результат; он аналогичен вышеупомянутому турнирному самообучению.

4-й «малый» шаг –

один шаг метода ветвей и границ для вспомогательной ЗДО – которая использует метод ветвей и границ не для решения всей задачи, а для построения близкого к максимальному (по какой-либо естественной метрике) блока (т.е. потенциального разделяющего элемента). При этом, в отличие от «большой» ЗДО, мы здесь сохраняем все решения – даже заведомо далёкие от оптимальных, поскольку нам для дальнейшей работы (для «большой» ЗДО) нужны, вообще говоря, все потенциальные разделяющие элементы (точнее – будут нужны на дальнейших стадиях работы алгоритма; как уже отмечалось, наши алгоритмы хорошо работают для очень больших размерностей именно в связи с тем, что генерация всех потенциальных разделяющих элементов заранее не производится. Поэтому в данной ситуации структура задачи как объекта немного иная; однако, здесь мы также формируем ППЗ, аналогичную 1-му «малому» шагу.)

}

Отметим, что данный алгоритм (точнее – схема алгоритма) очень удобен и для параллельной реализации. Также параллельная реализация применима при выборе очередной подзадачи из списка по разным критериям [10, 11].

Заключение

Возможности практического применения различных вариантов эвристических алгоритмов минимизации НКА больших размерностей становятся очевидными. Так управление в социоинженирии связанно с базами данных больших размерностей и это практически позволяет использовать именно недетерминированные автоматы.

При достижении наиболее подходящего способа управления, также удобно использовать улучшение текущего псевдооптимального решения задачи путём применения подходящей эвристики локального поиска.

Система турнирного самообучения, применяемая для проведения очередного шага метода ветвей и границ, позволяет настраивать уровень значимости факторов, влияющих на управление социоинженирией в определенной области.

Работа частично поддержана ФЦП "Научные и научно-педагогические кадры инновационной России" на 2009-2013 годы (соглашение № 14.B37.21.1934)

Список литературы

1. *Кузнецов М. Социальная инженерия и социальные хакеры [Электронный ресурс] / М. Кузнецов, И. Симдянов. – Режим доступа: http://hrazvedka.ru/book/socialnaya-inzheneriya.html.*
2. *Формирование имиджа политического лидера [Электронный ресурс]. – Режим доступа: .http://politobzor.net/show-6306-formirovanie-imidzha-politicheskogo-lidera.html.*
3. *Melnikov B. Edge-minimization of non-deterministic finite automata / B. Melnikov. – The Korean Journal of Computational and Applied Mathematics. – 2001. – № 3. – P. 469–479.*
4. *Hromkovic J. Algorithmics for Hard Problems. Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics / J. Hromkovic. – Springer, 2004. – 548 c.*
5. *Мельников Б.Ф. Мультиэвристический подход к задачам дискретной оптимизации / Б.Ф. Мельников // Кибернетика и системный анализ (НАН Украины). – 2006. – № 3. – С. 43-49.*
6. *Баумгертнер С.В. Мультиэвристический подход к проблеме звёздно-высотной минимизации недетерминированных конечных автоматов / С.В. Баумгертнер, Б.Ф. Мельников, С.В. Пивнева // Математические и компьютерные методы в технических, гуманитарных и общественных науках: Монография. – Пенза: Изд-во Приволжский Дом знаний, 2011. – . С. 101-112.*
7. *Мельников Б.Ф. Эвристические алгоритмы принятия решений в гуманитарных областях / С.В. Пивнева, Б.Ф. Мельников // «Известия Самарского научного центра Российской академии наук», Вып. 8. – Самара: Изд-во Самарского научного центра РАН. – 2009.*
8. *Пивнева С.В. Модель турнирного самообучения / С.В. Пивнева // Вектор науки. – ТГУ, 2010. – № 3(13). – С. 25-28.*
9. *Пивнева С.В. Принятие решений в прикладных задачах с применением динамически подобранных функций риска / С.В. Пивнева, Б.Ф. Мельников // Вестник транспорта Поволжья. – № 3 (23). – Самара : Самарский гос. ун-т путей сообщения, 2010. – С. 28-34.*
10. *Крайнюков Н.И. Вычисление конечно-автоматной сложности булевых функций как задача дискретной оптимизации / Н.И. Крайнюков, С.В. Пивнева // Вектор науки. – ТГУ, 2012. – № 4. – С. 30-54.*
11. *Рудницкий В.Н. Параллельная реализация процесса минимизации систем частично или полностью определенных булевых функций с большим числом переменных / В.Н. Рудницкий, С.В. Пивнева, С.В. Бурмистров // Вектор науки. – ТГУ, 2013. – № 4. – С. 55-59.*

Поступила в редколлегию 10.12.2013

Рецензент: д-р техн. наук, проф. В.Н. Рудницкий, Черкасский государственный технологический университет, Черкассы.

ОСОБЛИВОСТІ ЗАСТОСУВАННЯ МАТЕМАТИЧНОГО АПАРАТУ МІНІМІЗАЦІЇ НЕДЕТЕРМІНОВАНИХ КІНЦЕВИХ АВТОМАТІВ ДЛЯ ОЦІНКИ ЕФЕКТИВНОСТІ УПРАВЛІННЯ СОЦІОІНЖЕНІРИНГОМ

С.В. Півнєва, Н.В. Лада

У роботі розглянуто особливості застосування математичного апарату мінімізації недетермінованих кінцевих автоматів для оцінки ефективності управління соціоінженірингом, який базується на поєднанні комбінаторних і евристичних методів оптимізації.

Ключові слова: *соціоінженіринг, недетермінований кінцевий автомат (НКА), евристичні алгоритми, мінімізація НКА.*

THE PECULIARITIES OF APPLICATION OF MATHEMATICAL APPARATUS FOR MINIMIZATION OF NON-DETERMINISTIC FINITE AUTOMATONS FOR ASSESSING MANAGEMENT EFFECTIVENESS IN SOCIOENGINEERING

S.V. Pivneva, N.V. Lada

The paper discusses the features of mathematical minimization of non-deterministic finite automatons (NFA), based on the combination of heuristic and combinatorial optimization.

Keywords: *socioengineering, non-deterministic finite automaton (NFA), heuristic algorithm, to minimize the NCA.*