взаимодействие с клиентом, снизить издержки по продажам, улучшить качество обслуживания клиентов и в целом обеспечить рост прибыли компании.

Литература.

1. Демин В. CRM нельзя купить, CRM - это стратегия вашего бизнеса [Электронный ресурс]. – Режим доступа: - http://www.kazna.ru/news.html?id=466

2. Митичкин С. Разработка в системе 1С: Предприятие 8.0. – М. : 1С-Паблишинг, 2003. – С. 25-42.

Ногина Н.В., Билык А.В.

Институт информатики и искусственного интеллекта ДонНТУ

Построение кратчайшего пути в помеченном графе при помощи локальной редукции графа

Задача поиска кратчайших путей в графе является широко известной и важной для разнообразных приложений. Известен ряд алгоритмов решения этой задачи [1]. В настоящем докладе рассматривается задача поиска кратчайшего пути в помеченном графе от начальной вершины к некоторой финальной. В отличие от известных алгоритмов, предложен алгоритм решения, основанный на известном методе перехода от отмеченного графа к регулярному выражению, описывающему все пути из начальной в финальные вершины [2]. Предлагаемый алгоритм является модификацией алгоритма из [3].

Помеченным графом назовем восьмерку $G=(Q, E, X, Y, \mu, \rho, q_0, F)$, где Q – конечное множество вершин, E – множество дуг, X – множество отметок вершин, Y – множество отметок дуг, $\mu: Q \to X$ – функция разметки вершин, $\rho: E \to Y$ – функция разметки дуг, q_0 – начальная вершина, F–множество финальных вершин. Предполагается, что множества Q, E, X, Y – конечны, а пометки $y \in Y$ – положительные действительные числа.

Путем в графе G будем называть конечную последовательность $l = q_1\ e_1\ q_2\ e_2 ... e_{k-1}\ q_k$, где q_i – вершина, e_i – дуга, началом которой является вершина q_i, а концом – q_{i+1}. Отметка пути l – это последовательность отметок $w(l)=x_1\ y_1\ x_2\ y_2 ... y_{k-1}\ x_k$, *где* $x_i = \mu(q_i)$, $y_i = \rho(e_i)$.

Пусть $Pre(q_i)$ – множество начальных вершин всех дуг, входящих в q_i; $Post(q_i)$ – множество конечных вершин всех дуг, исходящих из q_i. Весом пути между вершинами q_0 и q_k назовем величину

$$dis(q_0, q_k) = \sum_{i=0}^{k-1} y_i \ .$$

Алгоритм.

Дан помеченный граф G с начальной и множеством финальных вершин.

Требуется построить отметку кратчайшего по весу пути.

Шаг 1. Создаем представление графа G в виде списка дуг с их отметками и весом, при этом отметки соответствующих вершин переносятся на дугу. Для неориентированных графов для каждого ребра между парой вершин q_i и q_j вводится две дуги: (q_i, q_j) и (q_j, q_i). В список вершин вводится фиктивная конечная вершина *fin*, а в список дуг – дуга из каждой финальной вершины q_i в вершину *fin*. Эта дуга помечается отметкой $\mu(q_i)$.

Шаг 2. If в графе существует хоть одна петля или существуют вершины, не являющиеся начальными, из которых исходит хоть одна дуга, then go to Шаг 3
Else go to Шаг 6.

Шаг 3. Удаление кратных дуг и петель

1. Удаляем все кратные дуги кроме одной из них с минимальным весом *dis*.
2. Удаляем все петли, присутствующие в графе.

На шагах 4 – 5 происходит удаление одной вершины.

Шаг 4. Выбираем $q_i \in Pre(fin)$;

$q := q_i$.

Шаг5. <u>If</u> $q \neq q_0$ <u>then</u> удаляем вершину q и все входящие и исходящие из нее дуги. Если при этом есть некоторый путь $q_j\ e_i\ q\ e\ q_k$, где $q_j \in Pre(q)$ и $q_k \in Post(q)$, то в граф добавляется дуга $(q_j\ ,q_k)$ с отметкой, полученной при помощи операции сочленения, описанной в [3];

$dis(qj,qk)= dis(qj,q)+$
$dis(q,qk)$; <u>go to</u> Шаг 2;

<u>else</u> q_i равная q_0 не исключается;
выбираем $q_m \in Pre(q_0)$;
$q := q_m$ <u>go to</u> Шаг 5.

Шаг 6. Удаляем все вершины, не равные q_0 и *fin* и все входящие и исходящие из них дуги. Получим граф, состоящий только из двух вершин: начальной и *fin,* соединенных одной дугой с минимальным весом *dis* с отметкой кратчайшего пути в графе.

Доказано, что алгоритм корректен.

Данный алгоритм является модификацией алгоритма из [3], определяемой существом задачи. Модификация заключается в следующем.

1. При удалении кратных дуг оставляем дугу с минимальным весом, в то время как в [3] выполняется объединение двух языков.

2. Удаление петель происходит без использования операции зацикливания (итерации), поскольку в кратчайшем пути в графе петли не участвуют.

3. При удалении вершин дополнительно производится вычисление веса *dis* отметки пути, в то время как в [3] выполняется только конкатенация языков.

4. В результате работы алгоритма вычисляется пометку кратчайшего пути из начальной вершины в одну из финальных, в то время как алгоритм из [3] позволяет

находить множество отметок всех путей из начальной вершины во все финальные.

Литература.

1. Ахо А. Построение и анализ вычислительных алгоритмов / А. Ахо, Дж. Хопкрофт, Дж. Ульман. – М. : Мир, 1979. – 536 с.

2. Хопкрофт Дж. Введение в теорию автоматов, языков и вычислений. 2-е издание / Дж. Хопкрофт, Р. Мотвани, Дж. Ульман. – М. : Издательский дом "Вильямс", 2002. – 528 с.

3. Ногина Н.В. Анализ языков, порожденных помеченными графами. / Н.В. Ногина, И.С. Грунский // Тезисы докладов международной научно-технической конференции «Системный анализ и информационные технологии» (SAIT 2012). – Киев, 2012. – в печати.