**УДК 681.518.54;**

***Брускин Сергей Наумович,***

***кандидат экономических наук, специальность 08.00.13***

***Национальный исследовательский университет   
«Высшая школа экономики», Москва, ул. Мясницкая, 20,***

***кафедра «Бизнес – аналитика», E-mail: sbruskin@hse.ru;***

***Дружаев Алексей Александрович,***

***кандидат технических наук, специальность 05.13.11,***

***Национальный исследовательский университет   
«Высшая школа экономики», Москва, ул. Мясницкая, 20,***

***кафедра «Бизнес – аналитика», E-mail:******adruzhaev@hse.ru;***

***Марон Аркадий Исаакович,***

***кандидат технических наук, старший научный сотрудник, специальность 05.13.07,***

***Национальный исследовательский университет   
«Высшая школа экономики», Москва, ул. Мясницкая, 20,***

***кафедра «Бизнес – аналитика», E-mail: amaron@hse.ru;***

***Марон Максим Аркадьевич,***

***магистр бизнес – информатики,***

***Национальный исследовательский университет   
«Высшая школа экономики», Москва, ул. Мясницкая, д. 20***

***аспирант кафедры «Бизнес-аналитика»,   
аспирантская школа по компьютерным наукам,   
специальность 05.13.18, E-mail:******maxxx-fizik@mail.ru***

**Эффективные методы построения алгоритмов поиска неисправностей в информационных системах.**

**Аннотация.** Актуальность исследуемой проблемы обусловлена тем, что оптимальные алгоритмы поиска неисправностей позволяют значительно сократить потери при отказах информационных систем. Вместе с тем, на пути построения таких алгоритмов нередко возникают непреодолимые вычислительные трудности. Цель статьи заключается в том, чтобы повысить эффективность методов построения алгоритмов поиска неисправностей в информационных системах. В статье предложены решения, которые упрощают процесс построения искомых алгоритмов при применении метода динамического программирования. Оптимизация ведётся по критерию минимума среднего времени поиска неисправностей. Материалы статьи могут быть полезными специалистам по обслуживанию, контролю и диагностике информационных систем.

**Ключевые слова:** эффективность, методы оптимизации, алгоритмы поиска неисправностей, информационные системы.

1. Вычислительные проблемы построения оптимальных алгоритмов поиска неисправностей

Информационныесистемы являются важнейшими элементом современной жизни. В большинстве отраслей промышленности и на транспорте от надёжности их работы зависит качество основных технологических процессов [2], [3], [4]. Важным показателем надёжности информационных систем является время восстановления работоспособности. Оно складывается из времени поиска неисправности, приведшей к отказу, и времени устранения неисправности. При этом, как правило, время поиска неисправности является доминирующим [7]. Восстанавливаемость информационных систем можно значительно повысить за счёт совершенствования технологии поиска неисправностей, а именно перехода на поиск неисправностей по составленным заранее алгоритмам [1], [6]. Для одной и той же информационной системы можно построить огромное количество алгоритмов поиска неисправностей, отличающихся порядком выполнения проверок. Каждому из них будет однозначно соответствовать определённое среднее время поиска неисправностей данной системы. Если потери от отказа системы линейно зависят от времени её восстановления, то оптимальным будет алгоритм, при котором среднее время поиска неисправностей минимально.

Если неисправности в информационной системе не накапливаются, то при отказе можно считать, что его причиной является одна единственная неисправность. В этом случае, для построения оптимального алгоритма поиска неисправностей должны быть заданы:

1. множество возможных неисправностей E1 = {ei} и их вероятности pi(i = 1,2,…n);
2. множество допустимых проверок j} и длительности их выполнения tj (j = 1,2,…m);
3. глубина диагноза – перечень подмножеств неисправностей, объединённых способом устранения;
4. соответствие между неисправностями и результатами проверок.

Информации о длительностях проверок, указанных в пункте 2, достаточно, когда время каждой проверки не зависит от того, какие проверки выполнялись до неё. Общей формой указания соответствия между неисправностями и результатами проверок является таблица неисправностей. Эта таблица размерности n X m, строки которой соответствуют возможным неисправностям, а столбцы – допустимым проверкам. На пересечении строки i со столбцом j указывается результат проверки j при неисправности ei . Каждая проверка может иметь 2 или более различных результата.

Сложность составления таблиц неисправностей существенно сдерживало внедрение методов технической диагностики. Эта проблема постепенно решается с развитием методов моделирования сложных технических систем [8] и созданием программного обеспечения, поддерживающего такие методы.

Для построения алгоритмов поиска неисправностей, оптимальных по критерию минимума среднего времени поиска, применяют метод динамического программирования или метод ветвей и границ [5, c. 65-112]. Это регулярные методы, гарантирующие построение оптимальных алгоритмов поиска неисправностей, как правило, за время меньшее, чем потребовалось бы для полного перебора возможных вариантов. Однако и их применение к реальным системам зачастую связано с огромными вычислительными проблемами.

Остановимся на реализации метода динамического программирования для построения оптимального алгоритма поиска неисправностей. Для простоты изложения предположим, что неисправностей, объединённых способом устранения, нет, и поиск должен осуществляться с точностью до неисправности, вызвавшей отказ информационной системы. Для применения метода динамического программирования предлагается рассматривать систему, для которой строится алгоритм поиска неисправностей, как стохастический объект управления, перемещающийся в фазовом пространстве возможных неисправностей, под действием допустимых проверок. Координатой каждого состояния в этом пространстве является текущий перечень возможных неисправностей E. В начальный момент E E1. Конечными являются состояния, для которых ||E|| = 1. Задача состоит в том, чтобы перевести объект управления из начального состояния в конечное за минимальное в среднем время. С точностью до обозначений, в литературе по технической диагностике предлагается следующий вид уравнения Беллмана, для вычисления среднего времени поиска неисправностей из промежуточного состояния [5, c. 90-91]:

] , ( 1)

где

T(E) – минимальное среднее время поиска неисправностей, начиная из состояния E;

tj – время проверки πj;

rjk – один из возможных результатов проверки πj  в состоянии E;

E\rjk – состояние, которое возникнет, если при выполнении проверки πj получен результат rjk;

P(E\rjk) – вероятность того, что при выполнении в состоянии E проверка πj будет иметь результат rjk;

T(E\rjk) – минимальное среднее время поиска неисправностей, начиная из состояния E\rjk.

В уравнении (1) минимум берется по всем проверкам, результат которых при состоянии E не является предопределённым, а суммирование осуществляется по всем результатам проверки πj, которые возможны в состоянии E.

Вычисление ведётся, начиная от состояний, в которых возможными являются две неисправности, и заканчивается расчётом минимального среднего времени неисправности для начального состояния E1. Соответственно, когда расчёт ведётся для состояния E, значения T(E\rjk) уже рассчитаны ранее, поскольку

|| E\rjk|| < ||E||.

Основные вычислительные проблемы связаны со следующим.

1. Расчёт по формуле (1) необходимо выполнить для всех возможных фазовых состояний E, кроме конечных. Их число, в худшем случае, составит 2n- n.
2. В каждом состоянии E для каждого возможного результата каждой допустимой проверки необходимо определить состояние, которое возникнет, если именно этот результат будет зафиксирован.
3. Для каждого состояния E надо рассчитать вероятности возникновения всех возможных результатов всех допустимых проверок.

Далее предложены решения, которые позволяют значительно уменьшить вычислительные проблемы при построении алгоритмов поиска неисправностей методом динамического программирования.

2. Уменьшение вычислительных трудностей при построении оптимальных алгоритмов поиска неисправностей

2.1. Метод определения состояний с помощью характеристических векторов.

Рассмотрим проверку πj, которой в таблице неисправностей соответствует набор различных результатов {rjк}. Поставим каждому результату rjк в соответствие двоичный вектор **r**jк размерности n, в котором элемент i равен “1”, если при неисправности i этот результат будет иметь место, и равен “0”, в противном случае.

Например, в таблице неисправностей 6 строк. Проверка π2 с номером j=2 может иметь L=3 результата. Первый из них r21 будет иметь место только при неисправности с номером 6, которой соответствует строка 6 таблицы. Второй r22 при неисправностях с номерами 1 и 2. Третий r23 при неисправностях с номерами: 3,4,5. Этим результатам будут соответствовать вектора: **r**21 =(000001); **r**22=(110000); **r**23=(001110). Будем называть их в дальнейшем характеристическими векторами результатов проверки.

Фазовому состоянию E поставим в соответствие двоичный вектор **E** размерности n, в котором элемент i равен “1”, если неисправность i возможна в состоянии E, и равен “0”, в противном случае. Например,   
если в состоянии E возможными являются неисправности с номерами 2,4,5,6, то этому состоянию E = {2,4,5,6} соответствует вектор **E** = (010111). Назовём его характеристическим вектором состояния E.

Для того, чтобы найти E\rjk – состояние, которое возникнет, если в состоянии E при выполнении проверки πj получен результат rjk, достаточно вычислить вектор **E**&**rjk**, осуществив поэлементное логическое умножение характеристических векторов **E** и **r**jк. Так после выполнения проверки π2 в состоянии **E** = (010111) могут возникнуть состояния, характеризуемые векторами:   
**E**&**r21 =** (010111)&(000001) = (000001);

**E**&**r22** = (010111)&(110000) = (010000);

**E**&**r23** = (010111)&(001110) = (000110).

Первые два из них – конечные, а третье по-прежнему является промежуточным.

Переход к фазовому пространству, в котором координатами состояний являются характеристические вектора возможных состояний, а каждая проверка (управление), характеризуется набором характеристических векторов возможных результатов, существенно упрощает расчёты. Ещё более значительно можно упростить вычисления, отказавшись от расчёта вероятностей результатов проверок. Об этом пойдёт речь далее.

2.2. Уравнение Беллмана для построения алгоритма поиска неисправностей, не требующее расчётов вероятностей результатов проверок

Пусть имеется состояние, характеризуемое вектором **E**, возникшее в результате выполнения определённых проверок. Ему соответствует T(**E**) - минимальное среднее время поиска неисправностей, начиная из этого состояния. Введём величину

T0(**E**) = T(**E**)\* P(**E**), (2)

где

P(**E**) – сумма исходных вероятностей pi неисправностей, возможных в состоянии **E**.

Назовём её весом фазового состояния **E**. Очевидно, что для начального состояния **E1** эта величина совпадает с минимальным средним временем поиска неисправностей, поскольку P(**E1**) = 1.

Можно доказать, что для неконечного состояния E, характеризуемого вектором **E** вес состояния можно найти с помощью рекуррентного соотношения

] (3)

где

T0(**E) –** вес состояния;

tj – время проверки πj;

P(**E**) – сумма исходных вероятностей pi неисправностей, возможных в состоянии **E**;

T0(**E**&**rjk**) – вес состояния **E**&**rjk.**

Минимум в (3) берётся по всем проверкам, результат которых при состоянии **E** не является предопределённым.

Начав расчёты с состояний, для которых ||**E**|| **=** 2, и закончив их в состоянии **E**1, найдём минимальное среднее время поиска неисправностей заданной информационной системы. При этом для каждого состояния будет найдена проверка π(**E**), при которой достигается минимум в соотношении (3). Именно её надо выполнять, если в процессе поиска неисправностей состояние **E** возникнет.

Вычисления с помощью соотношения (3) занимают на порядок меньше времени, чем вычисления при использовании соотношения (1).

Таким образом, процедура построения оптимального алгоритма поиска неисправностей такова.

1. Для всех возможных состояний E, начиная с состояний, для которых ||**E**|| **=** 2 и заканчивая начальным состоянием **E**1 определить вес состояния и проверку, которую надо выполнять, если это состояние возникнет.
2. Проверку π(**E1**), найденную для состояния **E**1 принять в качестве первой проверки алгоритма поиска неисправностей. Проверки, найденные для состояний, которые могут возникнуть при её различных результатах, принять в качестве следующих за ней, при соответствующих результатах. Продолжать аналогичный процесс, пока алгоритм не будет полностью построен.

Визуально такой условный алгоритм поиска представляется в виде дерева. Его внутренние вершины соответствуют проверкам, а листья – возможным неисправностям. Дуги соответствуют результатам проверок. В случае, если проверка не является двоичной, из вершины может выходить более двух дуг.

Предложенный переход к использованию характеристических векторов и весов состояний существенно снижает сложность вычислений.

Вместе с тем, кардинальное уменьшение вычислительных проблем возможно только в случае, если ещё на этапе проектирования информационных систем учитываются требования по упрощению поиска неисправностей. Например, система создаётся по блочному принципу, таким образом, чтобы её можно было представить функциональной моделью. Заметим, что представление объектов диагноза функциональными моделями эффективно не только для решения задач диагностики технических систем, но и для решения задач контроля правильности выполнения проектов [9].

Заключение

Изложенное позволяет сделать следующие выводы.

1. Переход к поиску неисправностей информационных систем по составленным заранее оптимальным условным алгоритмам поиска позволяет существенно снизить потери от отказов.
2. Построение таких алгоритмов связано со значительными вычислительными трудностями, несмотря на применение методов динамического программирования и метода ветвей и границ.
3. В данной работе предложены решения, которые позволяют в значительной степени преодолеть эти трудности при построении оптимальных алгоритмов поиска неисправностей методом динамического программирования.
4. Кардинальное уменьшение вычислительных проблем возможно только в случае, если ещё на этапе проектирования информационных систем учитываются требования по упрощению поиска неисправностей.

Литература

1. Гриненко А. В., Нестеров В. В., Лабецкий В. Л. Автоматизированная обучающая система для дистанций сигнализации и связи // Автоматика, связь, информатика. 2001. № 11. C. 22–25.
2. Гурин В. и др. Диагностика автоматизированного производства.- М.: Машиностроение. 2011. 600 с.
3. Ефанов Д. В., Плеханов П. А. Обеспечение безопасности движения за счет технического диагностирования и мониторинга устройств железнодорожной автоматики и телемеханики // Транспорт Урала, 2011. № 3. C. 44–48.
4. Лисенков В. М. Статистическая теория безопасности движения поездов. М: ВИНИТИ РАН. 1999. 332 с.
5. Пархоменко П. П., Согомонян Е. С. Основы технической диагностики: (Оптимизация алгоритмов диагностирования, аппаратурные средства). – М.: Энергия. 1981. 320 с.
6. Сапожников Вл. В., Сапожников В. В. Основы технической диагностики - М.: Маршрут. 2004. 316 с.
7. Шишмарев В. Диагностика и надежность автоматизированных систем. - М.: Academia, 2013. 352 с.
8. Якимович Б. А., Коршунов А. И., Кузнецов А. П. Теоретические основы конструктивно-технологической сложности изделий и структур-стратегий производственных систем машиностроения. – Ижевск, Издательство ИжГТУ. 2007. 280с.
9. Maron M. A. The choice of control points of projects taking into account possible change of structure of works // Business Informatics. 2016. 2 (36). P. 57 -61.

**References**

1. Grinenko A. V., Nesterov V. V., Labeckij V. L. Avtomatizirovannaya obuchayushchaya sistema dlya distancij signalizacii i svyazi // Avtomatika, svyaz', informatika. 2001. № 11. S. 22–25.
2. Gurin V. i dr. Diagnostika avtomatizirovannogo proizvodstva.- M.: Mashinostroenie. 2011. 600 s.
3. Efanov D. V., Plekhanov P. A. Obespechenie bezopasnosti dvizheniya za schet tekhnicheskogo diagnostirovaniya i monitoringa ustrojstv zheleznodorozhnoj avtomatiki i telemekhaniki // Transport Urala, 2011,   
   № 3. S. 44–48.
4. Lisenkov V. M. Statisticheskaya teoriya bezopasnosti dvizheniya poezdov. M: VINITI RAN. 1999. 332 s.
5. Parhomenko P. P., Sogomonyan E. S. Osnovy tekhnicheskoj diagnostiki: (Optimizaciya algoritmov diagnostirovaniya, apparaturnye sredstva). – M.: EHnergiya. 1981, 320 s.
6. Sapozhnikov Vl. V., Sapozhnikov V. V. Osnovy tekhnicheskoj diagnostiki - M.: Marshrut. 2004. 316 s.
7. SHishmarev V. Diagnostika i nadezhnost' avtomatizirovannyh sistem. - M.: Academia. 2013. 352 s.
8. YAkimovich B. A., Korshunov A. I., Kuznecov A. P. Teoreticheskie osnovy konstruktivno-tekhnologicheskoj slozhnosti izdelij i struktur-strategij proizvodstvennyh sistem mashinostroeniya. – Izhevsk. Izdatel'stvo IzhGTU. 2007. 280 s.
9. Maron M. A. The choice of control points of projects taking into account possible change of structure of works // Business Informatics. 2016. 2 (36). P. 57 -61.

***Sergey N. Bruskin,   
PhD of Economic Sciences, specialty 08.00.13***

***National Research University  
Higher School of Economics, Moscow, Myasnitskaya St., 20,***

***Department of Business Analytics, E-mail: sbruskin@hse.ru;***

***Alexey A. Druzhayev,***

***PhD of Engineering Sciences, specialty 05.13.11,***

***National Research University   
Higher School of economics, Moscow, Myasnitskaya St., 20,***

***Department of Business Analytics, E-mail:******adruzhaev@hse.ru;***

***Arkadiy I. Maron,***

***PhD of Engineering Sciences,*** ***Senior Research Officer, specialty 05.13.07,***

***National Research University   
Higher School of Economics, Moscow, Myasnitskaya St., 20,***

***Department of Business Analytics, E-mail: amaron@hse.ru;***

***Maxim A. Maron, Master of Science,***

***National Research University   
Higher School of Economics, Moscow, Myasnitskaya St., 20,***

***Postgraduate student, Department of Business Analytics,   
Doctoral School of Computer Science, specialty 05.13.18,   
E-mail:******maxxx-fizik@mail.ru***

**Effective methods for creation of malfunctions search algorithms in information systems.**

**Abstract:** relevance of the studied problem caused by the fact that optimum malfunctions search algorithms allow to reduce considerably losses at refusals of information systems. At the same time, on the way of creation of such algorithms quite often there are insuperable computing difficulties. The purpose of article is in increasing efficiency of methods for creation malfunctions search algorithms in information systems. In article, solutions proposed which simplify process of creation for required algorithms, when applied a method of dynamic programming. Optimization conducted by criterion of a minimum of average time for search of malfunctions. Materials of article can be useful to specialists in service, control and diagnostics of information systems.

**Keywords:** efficiency, optimization methods, malfunctions search algorithms, information systems.