

## ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

УДК 681.518.54

DOI: 10.22213/2410-9304-2017-3-88-93

С. Н. Брускин, кандидат экономических наук

А. А. Дружаев, кандидат технических наук

А. И. Марон, кандидат технических наук

М. А. Марон, аспирант

Национальный исследовательский университет «Высшая школа экономики», Москва

### ЭФФЕКТИВНЫЕ МЕТОДЫ ПОСТРОЕНИЯ АЛГОРИТМОВ ПОИСКА НЕИСПРАВНОСТЕЙ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

*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. Они связаны с необходимостью составления таблицы неисправностей для конкретной информационной системы и нахождения оптимального алгоритма среди огромного количества возможных алгоритмов поиска неисправностей. Первая из этих проблем в значительной степени нашла свое решение в результате появления мощных программных средств моделирования технических систем. Вторая по-прежнему остается нерешенной. 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 ведется по критерию минимума среднего времени поиска неисправностей. Предложено рекуррентное соотношение, которое не требует пересчета вероятностей при последовательном нахождении минимального среднего времени поиска неисправностей методом динамического программирования. Материалы статьи могут быть полезными специалистам по обслуживанию, контролю и диагностике информационных систем.*

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

#### **Вычислительные проблемы построения оптимальных алгоритмов поиска неисправностей**

Информационные системы являются важнейшим элементом современной жизни. В большинстве отраслей промышленности и на транспорте от надежности их работы зависит качество основных технологических процессов [1–3]. Важным показателем надежности 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 [4]. Восстанавливаемость информационных систем можно значительно повысить за счет совершенствования технологии поиска неисправностей, а именно перехода на поиск неисправностей по составленным заранее алгоритмам [5, 6]. Для одной и той же информационной системы можно построить огромное количество алгоритмов поиска неисправностей, от-

личающихся порядком выполнения проверок. Каждому из них будет однозначно соответствовать определенное среднее время поиска неисправностей данной системы. Если потери от отказа системы линейно зависят от времени ее 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  $E_1 = \{e_i\}$  и их вероятности  $p_i$  ( $i = 1, 2, \dots, n$ );

2) множество допустимых проверок  $\Pi = \{\pi_j\}$  и длительности их выполнения  $t_j$  ( $j = 1, 2, \dots, m$ );

3) глубина диагноза – перечень подмножеств неисправностей, объединенных способом устранения;

4) соответствие между неисправностями и результатами проверок.

Информации о длительностях проверок, указанных в п. 2, достаточно, когда время каждой проверки не зависит от того, какие проверки выполнялись до нее. Общей формой указания соответствия между неисправностями и результатами проверок является таблица неисправностей. Эта таблица размерности  $n \times m$ , строки которой соответствуют возможным неисправностям, а столбцы – допустимым проверкам. На пересечении строки  $i$  со столбцом  $j$  указывается результат проверки  $\pi_j$  при неисправности  $e_i$ . 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 [7] 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 [8, с. 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, объединенных 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  $E = E_1$ . 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 [9, с. 90–91]:

$$T(E) = \min \left[ t_j + \sum P(E \setminus r_{jk}) \cdot T(E \setminus r_{jk}) \right], \quad (1)$$

где  $T(E)$  – минимальное среднее время поиска неисправностей, начиная из состояния  $E$ ;  $t_j$  – время проверки  $\pi_j$ ;  $r_{jk}$  – один из возможных результатов проверки  $\pi_j$  в состоянии  $E$ ;  $E \setminus r_{jk}$  – состояние, которое возникнет, если при выполнении проверки  $\pi_j$  получен результат  $r_{jk}$ ;  $P(E \setminus r_{jk})$  – вероятность того, что при выполнении в состоянии  $E$  проверка  $\pi_j$  будет иметь результат  $r_{jk}$ ;  $T(E \setminus r_{jk})$  – минимальное среднее время поиска неисправностей, начиная из состояния  $E \setminus r_{jk}$ .

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

Вычисление ведется начиная от состояний, в которых возможными являются две неисправности, и заканчивается расчетом минимального среднего времени неисправности для начального состояния  $E_1$ . Соответственно, когда расчет ведется для состояния  $E$ , значения  $T(E \setminus r_{jk})$  уже рассчитаны ранее, поскольку

$$\|E \setminus r_{jk}\| < \|E\|.$$

Основные вычислительные проблемы связаны со следующим:

1. Расчет по формуле (1) необходимо выполнить для всех возможных фазовых состояний  $E$ , кроме конечных. Их число, в худшем случае, составит  $2^n - n$ .

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.

### **Уменьшение вычислительных трудностей при построении оптимальных алгоритмов поиска неисправностей**

#### ***Метод определения состояний с помощью характеристических векторов***

Рассмотрим проверку  $\pi_j$ , которой в таблице неисправностей соответствует набор различных результатов  $\{r_{jk}\}$ . Поставим каждому результату  $r_{jk}$  в соответствие двоичный вектор  $r_{jk}$  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  $\pi_2$  с номером  $j = 2$  может иметь  $L = 3$  результата. Первый из них  $r_{21}$  будет иметь место только при неисправности с номером 6, которой соответствует строка 6 таблицы. Вторым  $r_{22}$  при неисправностях с номерами 1 и 2. Третьим  $r_{23}$  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  $\mathbf{E} = (010111)$ . Назовем его характеристическим вектором состояния  $E$ .

Для того чтобы найти  $E \vee r_{jk}$  – состояние, которое возникнет, если в состоянии  $E$  при выполнении проверки  $\pi_j$  получен результат  $r_{jk}$ , достаточно вычислить вектор  $\mathbf{E} \& \mathbf{r}_{jk}$ , осуществив поэлементное логическое умножение характеристических векторов  $\mathbf{E}$  и  $\mathbf{r}_{jk}$ . Так, после выполнения проверки  $\pi_2$  в состоянии  $\mathbf{E} = (010111)$  могут возникнуть состояния, характеризуемые векторами:

$$\begin{aligned} \mathbf{E} \& \mathbf{r}_{21} &= (010111) \& (000001) = (000001); \\ \mathbf{E} \& \mathbf{r}_{22} &= (010111) \& (110000) = (010000); \\ \mathbf{E} \& \mathbf{r}_{23} &= (010111) \& (001110) = (000110). \end{aligned}$$

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 расчеты. Еще более значительно можно упростить вычисления, отказавшись от расчета вероятностей результатов проверок. Об этом пойдет речь далее.

***Уравнение Беллмана для построения алгоритма поиска неисправностей, не требующее расчетов вероятностей результатов проверок***

Пусть имеется состояние, характеризуемое вектором  $\mathbf{E}$ , возникшее в результате выполнения определенных проверок. Ему соответствует  $T(\mathbf{E})$  – минимальное среднее

время поиска неисправностей, начиная из этого состояния. Введем величину

$$T_0(\mathbf{E}) = T(\mathbf{E}) \cdot P(\mathbf{E}), \quad (2)$$

где  $P(\mathbf{E})$  – сумма исходных вероятностей  $p_i$  неисправностей, возможных в состоянии  $\mathbf{E}$ .

Назовем ее весом фазового состояния  $\mathbf{E}$ . Очевидно, что для начального состояния  $\mathbf{E}_1$  эта величина совпадает с минимальным средним временем поиска неисправностей, поскольку  $P(\mathbf{E}_1) = 1$ .

Можно доказать, что для неконечного состояния  $E$ , характеризуемого вектором  $\mathbf{E}$ , вес состояния можно найти с помощью рекуррентного соотношения

$$T_0(\mathbf{E}) = \min[t_j \cdot P(\mathbf{E}) + \sum T_0(\mathbf{E} \& \mathbf{r}_{jk})], \quad (3)$$

где  $T_0(\mathbf{E})$  – вес состояния;  $t_j$  – время проверки  $\pi_j$ ;  $P(\mathbf{E})$  – сумма исходных вероятностей  $p_i$  неисправностей, возможных в состоянии  $\mathbf{E}$ ;  $T_0(\mathbf{E} \& \mathbf{r}_{jk})$  – вес состояния  $\mathbf{E} \& \mathbf{r}_{jk}$ .

Минимум в (3) берется по всем проверкам, результат которых при состоянии  $\mathbf{E}$  не является предопределенным.

Начав расчеты с состояний, для которых  $\|\mathbf{E}\| = 2$ , и закончив их в состоянии  $\mathbf{E}_1$ , найдем минимальное среднее время поиска неисправностей заданной информационной системы. При этом для каждого состояния будет найдена проверка  $\pi(\mathbf{E})$ , при которой достигается минимум в соотношении (3). Именно ее надо выполнять, если в процессе поиска неисправностей состояние  $\mathbf{E}$  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.

Для всех возможных состояний  $E$ , начиная с состояний, для которых  $\|\mathbf{E}\| = 2$  и заканчивая начальным состоянием  $\mathbf{E}_1$ , оп-

ределить вес состояния и проверку, которую надо выполнять, если это состояние возникнет.

Проверку  $\pi(E_1)$ , найденную для состояния  $E_1$ , принять в качестве первой проверки алгоритма поиска неисправностей. Проверки, найденные для состояний, которые могут возникнуть при ее 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 еще на этапе проектирования информационных систем учитываются требования по упрощению поиска неисправностей. Например, система создается по блочному принципу таким образом, чтобы ее 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 [10].

### 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.

В данной работе предложены решения, которые позволяют в значительной степени преодолеть эти трудности при построении оптимальных алгоритмов поиска неисправностей методом динамического программирования.

Кардинальное уменьшение вычислительных проблем возможно только в случае, если еще на этапе проектирования информационных систем учитываются требования по упрощению поиска неисправностей.

### Библиографические ссылки

1. Диагностика автоматизированного производства / В. Гурин и др. – М. : Машиностроение, 2011. – 600 с.
2. Ефанов Д. В., Плеханов П. А. Обеспечение безопасности движения за счет технического диагностирования и мониторинга устройств железнодорожной автоматики и телемеханики // Транспорт Урала. – 2011. – № 3. – С. 44–48.
3. Лисенков В. М. Статистическая теория безопасности движения поездов. – М. : ВИНТИ РАН, 1999. – 332 с.
4. Шимарев В. Диагностика и надежность автоматизированных систем. – М.: Academia, 2013. – 352 с.
5. Гриненко А. В., Нестеров В. В., Лабецкий В. Л. Автоматизированная обучающая система для дистанций сигнализации и связи // Автоматика, связь, информатика. – 2001. – № 11. – С. 22–25.

6. Сапожников Вл. В., Сапожников В. В. Основы технической диагностики. – М. : Маршрут, 2004. – 316 с.

7. Якимович Б. А., Корицунов А. И., Кузнецов А. П. Теоретические основы конструктивно-технологической сложности изделий и структур-стратегий производственных систем машиностроения. – Ижевск : Изд-во ИЖГТУ, 2007. – 280 с.

8. Пархоменко П. П., Согомонян Е. С. Основы технической диагностики: (Оптимизация алгоритмов диагностирования, аппаратурные средства). – М. : Энергия, 1981. – 320 с.

9. Там же.

10. 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.

\*\*\*

S. N. Bruskin, PhD in Economics, National Research University “Higher School of Economics”, Moscow

A. A. Druzhayev, PhD in Engineering, National Research University “Higher School of Economics”, Moscow

A. I. Maron, PhD in Engineering, National Research University “Higher School of Economics”, Moscow

M. A. Maron, Post-graduate, National Research University “Higher School of Economics”, Moscow

### **Effective Methods for Creation of Malfunction Search Algorithms in Information Systems**

*The relevance of the studied problem is 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. They are connected with need of drawing up the table of malfunctions, for a concrete information system, and finding the optimum algorithm among a huge number of possible malfunctions search algorithms. The first of these problems substantially found the solutions because of emergence of the powerful software for technical systems modeling. The second still remains unsolved. The purpose of the paper is in increasing the efficiency of methods for creation of algorithms of malfunction search in information systems. In this paper solutions are proposed which simplify the process of creation for required algorithms, when the method of dynamic programming is applied. Optimization is performed by criterion of a minimum of the average time for search of malfunctions. The recurrence relation is offered, which does not demand recalculation of probabilities at consecutive finding of the minimum average time of malfunction search by the method of dynamic programming. Materials of the paper can be useful to specialists in service, control and diagnostics of information systems.*

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

Получено: 10.08.2017