УДК 509.6

ЗАДАЧА ФОРМИРОВАНИЯ ОТКАЗОУСТОЙЧИВОГО ПРОГРАММНОГО КОМПЛЕКСА УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМ ОБЪЕКТОМ

[1]Сташков Д.В., [2]Насыров И.Р., [2,3]Казаковцев Л.А.

[1]ЗАО «СИНЕТИК», Новосибирск;

[2]Сибирский государственный аэрокосмический университет им. ак. М.Ф. Решетнева, Красноярск;

[3]Сибирский федеральный университет, Красноярск, e-mail: stashkov@ngs.ru

Задача повышения надежности программных модулей систем управления технологическими процессами путем дублирования критически важных модулей при ограничении на бюджет сведена к задаче псевдобулевой оптимизации «рюкзачного» типа, для которой предложены известные методы решения. Параметрами оптимизационной модели являются программные модули, для каждого из которых определяется необходимость дублирования. Дополнительные версии модулей, для которых предусмотрено дублирование, должны быть разработаны независимо друг от друга различными командами исполнителей. Оценка вероятности обнаружения ошибки в программном модуле базируется на статистической модели Холстеда. Область применения предлагаемых моделей – синтез программных комплексов управления сложными системами промышленной автоматики, применяемыми как на производствах, так и в системах городской инфраструктуры, таких как автоматизированные системы управления очистных сооружений городской канализации.

Ключевые слова: промышленная автоматика, мультиверсионное программирование, псевдобулева оптимизация

PROBLEM OF FORMING A FAULT-TOLERANT INDUSTRIAL CONTROL SOFTWARE PACKAGE

[1]Stashkov D.V., [2]Nasyrov I.R., [2,3]Kazakovtsev L.A.

[1]SINETIC JSC, Novosibirsk;

[2]Siberian State Aerospace University Named after M.F. Reshetnev, Krasnoyarsk;

[3]Siberian Federal University, Krasnoyarsk, e-mail: stashkov@ngs.ru

Authors propose new statement of the problem of increase of fault tolerance of program control systems for technological processes via redundancy of the most important program units. This problem is stated as a constrained pseudo-Boolean optimization problem of knapsack type with budget constraint. Parameters of such optimization model are program units. For each unit, solving the optimization problem results in determining necessity of its duplication or using its single version. Additional versions of program units which need duplication must be developed independently by an independent team of computer programmers. Estimation of error detection probability for a program unit is based on the Halsted statistical model. The proposed models are useful for the synthesis of automated control systems used in industrial companies and city infrastructure systems such as automated control systems of purification plants of a city sewerage system.

Keywords: industrial automation, multi-version programming, pseudo-Boolean optimization

Современные системы управления промышленной автоматикой включают в себя сложные программно-аппаратные комплексы, состоящие из десятков и сотен компонентов: датчиков, исполнительных элементов, коммуникационных каналов, систем обработки информации и программных модулей. Дублирование функций аппаратных устройств достаточно хорошо изучено и описано в литературе [10, 3]. В критически важных узлах предусмотрено дублирование датчиков, исполнительных элементов, каналов передачи информации, серверов хранения и обработки информации, клиентских терминалов.

Производители устройств промышленной автоматики располагают обширной статистической информацией об отказах каждого вида устройств [15, 12]. Располагая подобной информацией, проектировщик имеет возможность оценить риски отказа той или иной подсистемы и спроектировать критически важные компоненты таким образом, чтобы применение наиболее надежных устройств в сочетании с дублированием и резервированием сводило риски к приемлемому уровню. В то же время сложные системы требуют разработки в своем составе программного комплекса управления технологическим процессом, который собственно и позволяет системе функционировать как единое целое, организуя взаимодействие ее компонентов в зависимости от условий окружающей среды и от хода технологического процесса. Отказы программного обеспечения, в том числе связанные с некорректной реализацией заложенных в систему управления математических моделей, могут приводить к отказу системы в целом, причем

дублирование одинаковых копий программного обеспечения в этом случае ни в коей мере не устраняет проблему. В этой связи средством повышения надежности может стать мультиверсионное программирование.

Причины отказов и статистические данные о них

Производители устройств промышленной автоматики снабжают потребителей, которыми зачастую являются проектные и внедренческие предприятия, исчерпывающим методическим обеспечением [15, 12] по разработке соответствующего программного обеспечения. Но разработка программного обеспечения всегда сопряжена с риском возникновения ошибок на различных этапах. Ошибки на этапе составления программного кода, выявляемые компилятором, здесь несущественны, поскольку выявляются до ввода программного модуля в эксплуатацию. То же можно сказать о грубых логических ошибках в коде, всегда выявляемых на этапе тестирования системы. Наибольшее влияние на надежность программного комплекса оказывают ошибки, незначительные в обычных режимах работы, но проявляющиеся при выполнении нескольких условий. Как правило, программные модули тестируются на работоспособность в предельных режимах работы. В то же время ошибки могут проявляться при достижении предельных значений нескольких параметров одновременно.

Еще более «коварными» являются ошибки в сложных математических моделях, на основе которых построен программный код. Современные узлы промышленной автоматики работают под управлением программных модулей различной сложности. Единственный программный модуль может реализовывать как очень простую, так и чрезвычайно сложную логику управления исполнительными элементами в зависимости от показаний множества датчиков и накопленной информации. Рассмотрим, например, программный модуль, реализующий управление подсистемой распределительных камер аэротенков в системе управления городской канализацией (ПРКА ГК, рисунок). Соответственно, сложность программной реализации модулей чрезвычайно сильно различается. В состав изображенной подсистемы исполнительных механизмов подсистемы входят 8 регулирующих затворов, контрольно-измерительными приборами являются датчики уровня стоков и датчики положения затворов. Основной функцией алгоритма управления ПРКА является автоматическое позиционирование затворов в зависимости от задания перепада уровней стоков в распределительных камерах. Задача подсистемы – поддержание заданного перепада уровней в распределительных камерах посредством регулирования положения. Математическая модель расчёта расхода стоков, протекающих через затвор:

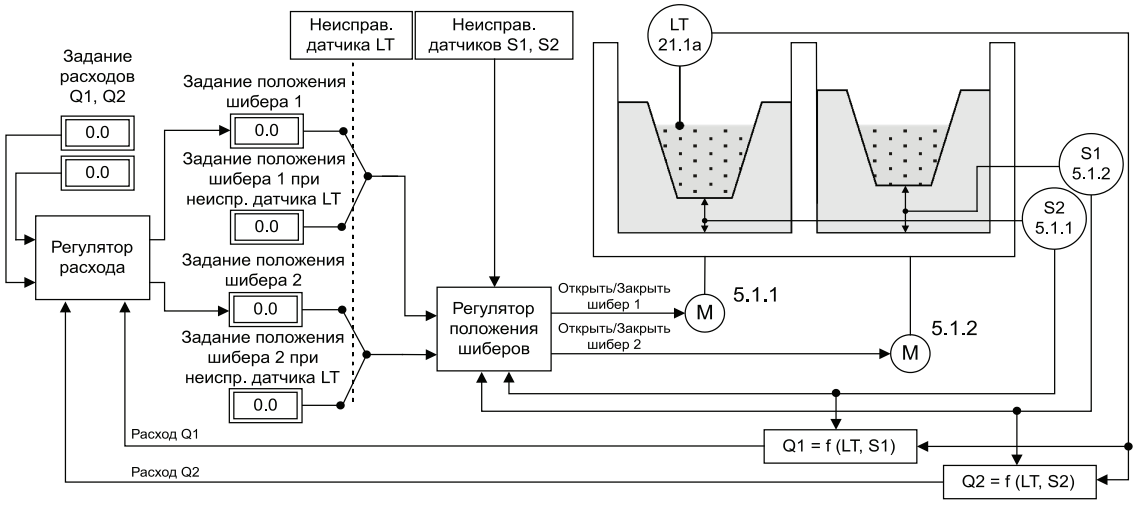
$$Q = 1{,}86 \cdot B \cdot (L - S)^{3/2},$$

где B – ширина нижнего края затвора; L – уровень воды в распределительной камере; S – положение шибера (в мм). Значения L и S также оцениваются с использованием соответствующих математических моделей. Кроме того, для данного алгоритма задан комплекс условий разрешения работы. Сформированные задания поступают на регулятор положения, задачей которого является позиционирование шиберов.

Также программный модуль содержит обширное множество алгоритмов работы в нештатных ситуациях – при неисправности датчиков и исполнительных элементов.

Готовый программный модуль должен быть настроен. Далеко не полный перечень настраиваемых параметров приведен в таблице. Некорректный программный код любого из алгоритмов, реализованных в программном модуле, или некорректные значения параметров приводят к сбою в программном модуле и некорректной работе всей системы. Поскольку программный модуль, изображенный на рисунке, реализует вспомогательные функции системы, его отказ не приведет к немедленной аварии всей системы. В то же время, в той же рассматриваемой системе имеются чуть менее сложные модули, управляющие ее критически важными элементами. Отказ модуля может быть обнаружен соответствующими средствами мониторинга. В то же время, некорректная работа модуля (некорректная реализация математических моделей) приводит к работе системы в неоптимальном режиме, что приводит к быстрому износу, повышению риска аварийных ситуаций в иных частях системы, работающих в данном случае с повышенной нагрузкой, а в случае систем, подобных представленной на рисунке, к серьезным экологическим последствиям в среднесрочной перспективе. Обнаружение подобных неявных ошибок чрезвычайно затруднено.

В случае отказа программного модуля вследствие аппаратного сбоя средством повышения надежности системы, безусловно, является дублирование аппаратных ресурсов – датчиков, серверов, телекоммуникационных каналов. Нарушение же логики работы программы, не приводящее к ее остановке, будет повторяться на любом количестве ее копий.



Структурная схема алгоритма управления ПРКА ГК (пример)

Параметры настройки алгоритма (пример)

Параметр	Описание	Значение
$Q1_{ном}$, $Q2_{ном}$	Номинальное значение задания расхода	150 мм
$Q1_{min}$, $Q2_{min}$	Минимальное значение задания расхода	50 мм
$Q1_{max}$, $Q2_{max}$,	Максимальное значение задания расхода	1000 мм
$T_{ош.рег}$	Задержка сообщения об ошибке регулирования	120 с
$S_{1\ безоп}$, …, $S_{8\ безоп}$	Уставка безопасного положения затворов	50 мм
$L_{1\ авар.\ выс}$, $L_{2\ авар.\ выс}$	Аварийно высокий уровень в распр. камере	5800 мм
$L_{1авар.низк}$, $L_{2\ авар.\ низк}$	Аварийно низкий уровень в распределительной камере	900 мм
$T_{зад.Lавар}$	Задержка формирования сообщения об аварийном уровне в распределительной камере	60 с

Методы мультиверсионного программирования и оценки вероятности возникновения ошибок

Технология высоконадежного мультиверсионного программирования традиционно применяется в военной, в основном – космической [9, 8] сфере. Наземные комплексы управления космическими аппаратами комплектуются программным обеспечением, отдельные модули которого дублируют функции друг друга. Версии модуля разрабатываются независимыми командами программистов, раздельно тестируются. Кроме того, часто используются различные средства разработки. Версии исполняются параллельно, результаты работы каждой из них передаются вспомогательному модулю, осуществляющему сравнение результатов. Если результаты (в пределах установленной погрешности) совпадают, исполнительному элементу передается результат одного из модулей. В случае несовпадения, например, управляющее воздействие выбирается «большинством голосов» (требуется нечетное число версий модулей). В любом случае применение мультиверсий позволяет выявить наличие проблемы в программном коде. Выявленная ошибка в программном коде может служить сигналом к переводу системы в особый аварийный режим работы, к остановке производства и т.д., до выявления причины ошибки и ее устранения. Таким образом, во многих случаях системы промышленной автоматики не требуют наличия нечетного количества мультиверсий. Наличие хотя бы двух независимо разработанных версий позволяет в большинстве случаев выявлять ошибку.

Пусть вероятность возникновения ошибки в i-й версии программного модуля равна p_i. Тогда вероятность одновременного возникновения ошибки во всех модулях равна

$$\prod_{i=1}^{N} p_i,$$

где N – число версий модуля, и вероятность обнаружения ошибки равна $1-\prod_{i=1}^{N} p_i$.

Очевидно, что данное значение стремится к 1 с ростом N. Поскольку программные модули проходят всестороннее тестирование, вероятность ошибки в каждой из версий модуля невелика. Например, если вероятность ошибки в каждой из двух версий модуля равна 0,001 %, вероятность одновременного возникновения ошибки сокращается до 0,0000001 %.

Для оценки предполагаемого числа ошибок используются разнообразные модели. Например, в [7] предлагается следующая оценка первоначального числа ошибок D_0:

$$D_0 = V/3000, \quad (1)$$

где V – объем программы в бит информации. Объем, в свою очередь, оценивается как

$$V = N \cdot \log_2 \eta,$$

где $\eta = \eta_1 + \eta_2$ – число уникальных операторов и операндов программы; $N = N_1 + N_2$ – число обращений к ним в ПО.

Число ошибок на m этапе тестирования модель Холстеда [13] оценивает как

$$D_m = \frac{E^{(2/3)}}{3200}, \quad (2)$$

где E – оценка сложности системы:

$$E = \frac{\eta_1 N_2 N \log_2 \eta}{2\eta_2}.$$

Данные модели сложности и предполагаемого числа ошибок могут быть использованы в качестве грубых оценок. Пусть для каждого из M модулей имеется несколько версий, V_j для j-го модуля. Пусть P_{ji} – вероятность безотказной работы i-й версии j-го модуля. Если отказ любого из модулей ведет к отказу всей системы, то вероятность P безотказной работы системы может быть выражена следующим образом:

$$P = \prod_{j=1}^{M} \left(1 - \prod_{i=1}^{V_j} \left(1 - P_{ij}\right)\right).$$

Оптимизационная модель надежности программного комплекса

В свою очередь, вероятность безотказной работы может быть оценена через число ошибок (1)–(2).

Сложность каждого из модулей можно считать приблизительно одинаковой вне зависимости от конкретной реализации. Таким образом, зная сложность единственной версии каждого модуля, может быть получена оценка вероятности безотказной работы системы в целом в зависимости от числа версий каждого из модулей и вероятности безотказной работы P_j единственной версии:

$$P = \prod_{j=1}^{M} \left(1 - V_j \left(1 - P_j\right)\right).$$

В то же время увеличение числа версий каждого из модулей сопряжено с увеличением стоимости системы. Для систем промышленной автоматики будем считать приемлемым число версий модуля, не превышающее 2. Введем булевы переменные x_j, равные 1, если предполагается дублирование j-го модуля и константы $\underline{C_j}$ – стоимость разработки j-го модуля. Тогда при заданном бюджете C на разработку всего программного комплекса имеем оптимизации:

$$\begin{gathered} P = \prod_{j=1}^{M} \left(1 - \left(1 + x_j\right)\left(1 - P_j\right)\right) \to \max, \\ \sum_{j=1}^{M} \left(1 + x_j\right) \cdot C_j \le C. \end{gathered} \quad (3)$$

Задачи псевдобулевой оптимизации подобного («рюкзачного») типа исследованы в работах А.Н. Антамошкина [11, 1, 2] и др. Созданы методы [5, 14], позволяющие получать субоптимальное (достаточно точное) решение подобных задач размерности до миллионов переменных, что позволяет решать задачи, подобные (3). Для задач с монотонной целевой функцией могут быть также успешно применены жадные эвристики [4] или специальные агломеративные жадные эвристики [6].

Заключение

Задача повышения надежности систем промышленной автоматики кроме дублирования функций аппаратных устройств требует в некоторых случаях дублирования также и программных модулей. Задача построения оптимальной конфигурации требуемых программных модулей может быть сведена к задаче псевдобулевой оптимизации.

Список литературы

1. Антамошкин А.Н., Казаковцев Л.А. Алгоритм случайного поиска для обобщенной задачи Вебера в дискретных координатах // Информатика и системы управления. – 2013. – Вып. 1. – С. 87–98.

2. Антамошкин А.Н., Казаковцев Л.А. Применение метода изменяющихся вероятностей для задач размещения на сети // Вестник СибГАУ. – 2014. – № 5(57).

3. Бабешко Е.В. Анализ возможностей современных промышленных контроллеров для разработки отказоустойчивых систем // Радіоелектронні і комп'ютерні системи. – 2006. – Вып. 7(19). – С. 36–39.

4. Дюбин Г.Н., Корбут А.А. Жадные алгоритмы для задачи о ранце: поведение в среднем // Сибирский журнал идустриальной математики. – 1999. – Т. II, № 2(4). – С. 68–93.

5. Казаковцев Л.А. Параллельный алгоритм случайного поиска с адаптацией для систем с распределенной памятью // Системы управления и информационные технологии. –2012. – № 3 (49). – С. 11–15.

6. Казаковцев Л.А., Ступина А.А., Орлов В.И. Модификация генетического алгоритма с жадной эвристикой для непрерывных задач размещения и классификации // Системы управления и информационные технологии. – 2014. – № 2(56). – С. 35–39.

7. Марков А.С. Модели оценки и планирования испытаний программных средств по требованиям безопасности информации // Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение». – 2011. Специальный выпуск «Технические средства и системы защиты информации».– С. 90–103.

8. Ступина А.А., Ежеманская С.Н. Технология надежностного программирования задач автоматизации управления в технических системах. – Красноярск: СФУ, 2011. – 164 с.

9. Царев Р.Ю., Капулин Д.В., Машурова Д.В., Тынченко Я.А., Ковтанюк Д.Н. Многоатрибутивное формирование гарантоспособных систем управления и обработки информации // Вестник СибГАУ. – 2012. – Вып. 5(45). – С. 106–110.

10. Ястребенецкий М.А. Определение надежности аппаратуры промышленной автоматики в условиях эксплуатации. – М.: Рипол Классик, 2014. – 134 с.

11. Antamoshkin A.N., Kazakovtsev L.A. Random Search Algorithm for the p-Median Problem, Informatica (Ljubljana). – 2013. – № 37(3). – P. 267–278.

12. GE Fanuc Automation. Series 90-70. Hot Standby CPU Redundancy. User's Guide // Ge Fanuc Automation North America. – 1993. – 93 c.

13. IEEE Std. 1061-1998 IEEE Computer Society: Standard for Software Quality Metrics Methodology. – 1998. – 20 p.

14. Kazakovtsev L.A. Random Constrained Pseudo-Boolean Optimization Algorithm for Multiprocessor Systems and Clusters // ICUMT 2012, International Congress on Ultra-Modern Telecommunications. IEEE Press. – SPb., 2012. – P. 650–656.

15. Siemens Simatic. S7-400H Programmable Controller. Fault-Tolerant Systems. – Siemens AG, 2003. – 328 c.

References

1. Antamoshkin A.N., Kazakovtsev L.A. *Algoritm sluchainogo poiska dlya obobschennoi zadachi Vebera v diskretnykh koordinatakh* [Random search algorithm for a generalized Weber problem in discrete coordinates]. *Informatika i sistemy upravleniya*. 2013, issue 1, p. 87–98.

2. Antamoshkin A.N., Kazakovtsev L.A.. *Primenenie metoda izmeniayuschikhsya veroyatnostei dlya zadach razmescheniya na seti* [Using the probability changing method for location problems on a network]. *Vestnik SibGAU*, 2014. 5(57).

3. Babeshko E.V. *Analiz vozmozhnostey sovremennykh promyshlennykh kontrollerov dlya razrabotki otkazoustoichivykh system* [Capability study of modern industrial controllers for developing reliable systems]. Radioelektronnye i kompyuternye sistemi. 2006, no. 7(19), pp. 36–39.

4. Dyubin G.N., Korbut A.A. *Zhadnye algoritmy dlya zadachi o rantse: povedenie v srednem* [Greedy algorithms for knapsack problem: average behavior]. *Sibirskii Zhurnal Industrialnoi Matematiki*. 1999. Vol. II, issue 2(4), p*p*. 68–93.

5. Kazakovtsev L.A. *Parallelnyi algoritm sluchainogo poiska s adaptatsiei dlya system s raspredelennoi pamyatyu* [Parallel random search algorithm with adaptation for shared memory systems]. *Sistemy upravleniya i informatsionnye tekhnologii*. 2012. issue 3 (49), pp. 11–15.

6. Kazakovtsev L.A., Stupina A.A., Orlov V.I. *Modifikatsiya genetichskogo algoritma s zhadnoi evristikoi dlya nepreryvnykh zadach razmescheniya i klassifikaksii* [Modification of genetic algorithm with greedy heuristic for location and classification problems]. *Sistemy upravleniya i informatsionnye tekhnologii*. 2014, issue 2(56), p**p**. 35–39.

7. Markov A.S.. *Modeli otsenki i planirovaniya ispytanii programmnykh sredstv po trebovaniiam bezopasnosti informatsii* [Models of estimating and planning of computer program tests according to information security demands]. Herald of the Bauman Moscow State Technical University. Mechanical Engineering. 2011. Special issue «Technical means and systems of information security», pp. 90–103.

8. Stupina A.A., Yezhemanskaya S.N. *Tekhnologiya nadezhnostnogo programmirovaniya zadach avtomatizatsii v tekhnicheskikh sistemakh* [**A**ault-tolerant programming of automation problems in technical systems]. Krasnoyarsk, Siberian Federal University, 2011**б** 164 p.

9. Tsarev R.Yu., Kapulin D.V., Mashurova D.V., Tynchenko Ya.A., Kovtanyuk D.N. *Mnogoatributivnoe formirovanie garantosposobnykh sistem upravleniya i obrabotki informatsii* [Multi-attributive synthesis of dependable control and information systems] // *Vestnik SibGAU*. 2012, no. 5(45), pp. 106–110.

10. Yastrebenetskii M.A. *Opredelenie nadezhnosti apparatury promyshlennoy avtomatiki v usloviyah ekspluatatsii* [Determining industrial automation systems reliability in service]. Moscow, Ripol Classic, 2014, 134 p.

11. Antamoshkin A.N., Kazakovtsev L.A. Random Search Algorithm for the p-Median Problem, Informatica (Ljubljana)**.** 2013. 37(3), p**p**. 267–278.

12. GE Fanuc Automation. Series 90-70. Hot Standby CPU Redundancy. Users Guide. Ge Fanuc Automation North America. 1993, 93 p.

13. IEEE Std. 1061-1998 IEEE Computer Society: Standard for Software Quality Metrics Methodology. 1998, 20 p.

14. Kazakovtsev L.A. Random Constrained Pseudo-Boolean Optimization Algorithm for Multiprocessor Systems and Clusters // ICUMT 2012, International Congress on Ultra-Modern Telecommunications. IEEE Press. S-Petersburg. 2012, pp. 650–656.

15. Siemens Simatic. S7-400H Programmable Controller. Fault-Tolerant Systems.– Siemens AG. 2003, 328 p.

Рецензенты:

Антамошкин А.Н., д.т.н., профессор, зав. кафедрой математического моделирования и информатики, ФГБОУ ВПО «Красноярский государственный аграрный университет», г. Красноярск;

Ступина А.А., д.т.н., профессор, зав. кафедрой «Экономика и информационные технологии менеджмента», ФГАОУ ВПО «Сибирский федеральный университет», г. Красноярск.