*Скворцов М.С.*

# Решение задачи оптимизации надежности с помощью метода логико-вероятностных вкладов

*Рассматривается задача оптимизации надежности последовательно-параллельной системы состоящей из нескольких комбинаторных подсистем функционирующих по принципу k из n. Каждая подсистема состоит из разных (неодинаковых) компонентов. Задача заключается в выборе компонентов и кратности их резервирования, таким образом, чтобы минимизировать стоимость системы, при выполнении заданного требования к надежности системы в целом. В такой постановке эта задача была рассмотрена в [1,2]. В данной статье вероятность безотказной работы комбинаторных подсистем рассчитывается по точным аналитическим формулам, полученным с помощью метода из работы [14], позволяющего достаточно просто получить аналитическое выражение для вероятности безотказной работы систем функционирующих по принципу k из n. Предложенный оптимизационный подход использует значения первых частных производных функции вероятности безотказной работы, и в этом аналогичен методу градиентного спуска. Предложенный метод имеет важные отличия от метода градиентного спуска, которые, позволяют резко снизить вероятность попадания в локальные экстремумы, за счет:*
*1) возможности в процессе поиска оптимального решения нарушать ресурсное ограничение и 2) на каждом шаге алгоритма решается задача одномерной оптимизации вдоль каждого возможного направления движения. Работоспособность предложенного подхода проверена сравнением полученных результатов с результатами из работы [1].*

## 1 Постановка задачи

В данной статье, на примере решения оптимизационной задачи выбора компонентов и кратности их резервирования (задачи оптимизации надежности) для последовательно-параллельной системы, представлен разработанный оптимизационный *метод логико-вероятностных вкладов*. Разработанный алгоритм имеет сходство с методом градиентного спуска, но имеет важные отличия, позволяющие резко снизить вероятность попадания в локальные экстремумы. Данная задача оптимального резервирования является комбинаторной оптимизационной задачей, в которой целевая надежность достигается при помощи дискретного выбора компонентов из доступных вариантов.

Проектирование технической системы связано с множеством дискретных операций выбора среди различных типов доступных компонентов на основании их стоимости, надежности, производительности, веса и т.п. В случае, когда необходимо минимизировать стоимость при заданном требовании к надежности, решением задачи является найденный оптимальный элементный состав подсистем, которые в рассматриваем случае образуют последовательно-параллельную систему. Если существует множество функционально одинаковых элементов, из которых возможен выбор, то становится сложно определить оптимальное решение, особенно в случае если для увеличения надежности можно использовать резервирование. В данной статье под резервированием понимается совместная параллельная работа функционально схожих (но необязательно идентичных) элементов, такая, что если один элемент отказал, то резервный способен выполнить заданную функцию без возникновения отказа системы.

Одна из трудностей при решении данной задачи заключается в сложности расчета показателя надежности системы. В статье [1] отмечается, что сложность в вычислении численного показателя надежности присутствует для задач оптимизации надежности для систем с сетевой структурой и для систем включающих в себя несколько подсистем с резервированием $k$-из-$n$, в особенности если $n$ элементов включенные параллельно друг другу могут быть различными (неодинаковыми). Вычисление надежности таких подсистем становится сложной комбинаторной задачей при больших значениях $k$. В статье [3] показано, что оптимизация таких систем есть NP-сложная задача.

Далее, будет показано, что применение алгоритма разработанного Можаевым А.С. [14] позволяет снять некоторые сложности связанные с точным вычислением надежности комбинаторных систем.

Резервирование $k$-из-$n$ определим как $n$ параллельно соединенных компонентов, при этом как минимум любые $k$-из-$n$ должны быть в работе (исправны) для того, что бы подсистема в целом была работоспособна. Максимальное количество параллельно соединенных компонентов для каждой подсистемы $n_i$ может принимать любое целое значение большее или равное $k_i$, где $i$ номер подсистемы. В данной статье, также как и в работе [1], принято, что $k_i$ задано, в отличие от $n_i$, которая остается переменной (ограниченной сверху) и должна быть определена в процессе оптимизации. На рис.1 показан пример типовой последовательно-параллельной системы.

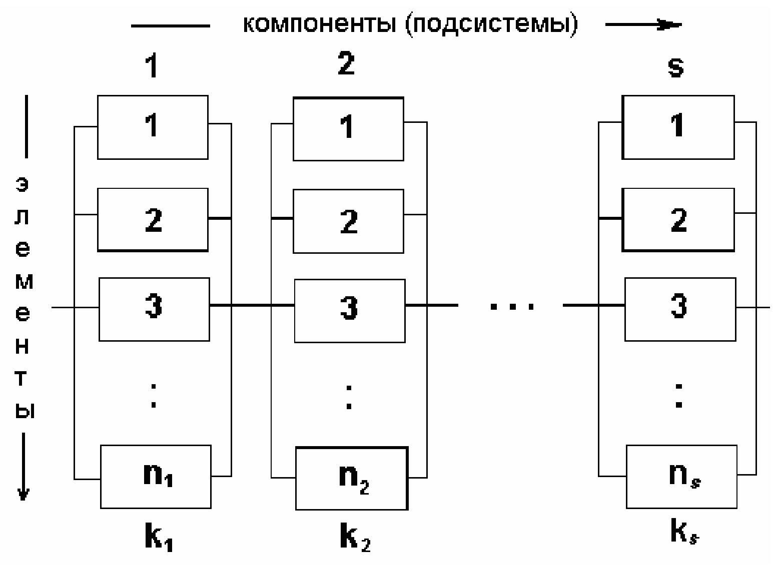



Рис. 1 Типовая последовательно-параллельная система

Задача оптимизации надежности для системы, приведенной на рис.1 может быть сформулирована следующим образом:

$$\min \sum_{i=1}^{S} C_i(\mathrm{X}_i)$$

$$\prod_{i=1}^{S} R_i(x_i \mid k_i) \geq R \tag{1}$$

$$\sum_{j=1}^{m_i} x_{ij} \geq k_i,\ \forall i$$

где $C_i$ – стоимость $i$-ой подсистемы;

$R_i$ – надежность $i$-ой подсистемы;

$R$ – ограничение на надежность;

$\mathrm{X}_i = (x_{i1}, x_{i2}, \ldots, x_{im_i})$, $\quad n_i = \sum_{j=1}^{m_i} x_{ij}$;

$x_{ij}$ – количество $j$-ых компонентов, использованных в $i$-ой подсистеме, $x_{ij} \in (0,1,2\ldots)$.

Если верхний предел для $n$ не ограничен, то количество возможных структурных конфигураций также не ограничено. Однако, на практике, имеет смысл установить верхний предел на $n$ ($n_{\max}$). Если ограничить $n$ сверху, то становится возможным посчитать общее количество вариантов конфигурации (элементного состава) системы, рассматривая выбор элементного состава компонентов для каждой подсистемы как задачу о размещениях [3]. Если количество функционально одинаковых компонентов для каждой подсистемы $i$ обозначить $m_i$, то количество уникальных вариантов построения системы может быть вычислено по следующей формуле [3]:

$$N = \prod_{i=1}^{S} \left[ \binom{m_i + n_{\max}}{m_i} - \binom{m_i + k_i - 1}{m_i} \right] \tag{2}$$

Для задачи, рассмотренной в статье [1] $s = 6$, $m_i = 10$ $(\forall i)$ и $n_{\max} = 8$ возможно более чем $6.9 \times 10^{27}$ различных вариантов построения системы. Поэтому использование прямого перебора вариантов сильно затруднено.

В работе [1], посвященной проблеме (1), для решения был использован оптимизационный подход на основе генетического алгоритма. Генетический алгоритм представляет собой вероятностный эвристический метод оптимизации, основанный на принципах естественного отбора имеющего место в процессе биологической эволюции. Методологическая основа генетического алгоритма состоит из: представления решения (элементного состава системы) в виде вектора (набора хромосом); случайного выбора начальной популяции решений; вычисление приспособленности (значения целевой функции) для каждого решения; выбор родителей; скрещивание родителей (рекомбинация хромосомных наборов) для создания новых решений (детей); мутация существующих решений, и отбраковка худших решений. Этот процесс, носящий итеративный характер, называется генерацией поколений, и он продолжается до тех пор, пока процесс поиска не сойдется или не наступит иное условие останова. Несмотря на все достоинства, недостатком генетического алгоритма является его эвристический характер.

В той же работе [1] предложено использовать нейронные сети для оценки вероятности безотказной работы отдельных $k$ из $n$ подсистем, основываясь на $k$, $n$, и вероятностях отдельных элементов, образующих подсистемы. Достоинствами нейронных сетей является их достаточная эффективность моделирования для случаев нелинейного и статистически значимого взаимодействия. Недостатками является необходимость обучения нейронной сети, а также приближенная оценка вероятности безотказной работы подсистемы. Для частичного устранения второго недостатка, авторы статьи [1] предлагают использование псевдоограничения, включающего в себя небольшую положительную добавку, для того чтобы полученное решение удовлетворяло изначальному ограничению:

$$R_{psuedo} = R + \gamma, \tag{3}$$

где $\gamma \sim f(s)$.

Зависимость $\gamma$ в формуле (3) от количества подсистем $s$, обусловлена тем, что нейронная сеть используется для расчета вероятности безотказной работы отдельных подсистем, поэтому происходит геометрическое нарастание ошибки при расчете надежности последовательных структур. Таким образом, неочевидный выбор $\gamma$ для каждой конкретной задачи является фактором, затрудняющим использование предложенного в [1] подхода.

В отличие от генетического подхода *оптимизационный метод логико-вероятностных вкладов* является точным (в плане расчета вероятности безотказной работы комбинаторных подсистем) методом, позволяющим достаточно быстро найти оптимальное решение за счет использования частных производных функции вероятности безотказной работы (вероятностного многочлена). Для расчета надежности комбинаторных подсистем функционирующих по принципу $k$ из $n$ на этапе построения логической функции работоспособности используется алгоритм формирования комбинаций [14], на этапе перехода от логической функции к вероятностному многочлену метод квазиортогонализации по одной переменной [7,8,14]. Использование этих методов позволяет провести точный расчет надежности комбинаторных подсистем с небольшими вычислительными затратами. Пример расчета надежности комбинаторных подсистем будет рассмотрен в статье далее.

## 2 Теоретические основы оптимизационного метода логико-вероятностных вкладов

В основе метода лежит общий логико-вероятностный метод моделирования и расчета надежности систем [4-6]. Логическая модель последовательно-параллельной системы строится при помощи аппарата схем функциональной целостности (СФЦ) [4-6]. Исчерпывающее описание аппарата СФЦ и общего логико-вероятностного метода приведено в [7]. Здесь лишь отметим, что при помощи СФЦ в графическом виде показываются функциональные связи между элементами образующими систему и характер обеспечения работоспособности одних элементов другими элементами. СФЦ используется для записи в виде *направленного графа* логических уравнений, отражающих функциональные связи элементов друг с другом. Затем, с ее помощью, находят кратчайшие пути успешного функционирования (или минимальные сечения отказов) при помощи любого из известных методов, таких как: метода прямого поиска в глубину, метода прямого поиска в ширину, метод универсальной аналитической подстановки [7]. Далее происходит переход от логической функции работоспособности к вероятностному многочлену при помощи любого из алгоритмов, описанных в [7, 11, 14], например, алгоритма разрезания, алгоритма ортогонализации, алгоритма наращивания путей, рекуррентного алгоритма, комбинированного алгоритма. Графический аппарат СФЦ, автоматическое построение логической функции, а затем автоматический переход от нее к вероятностному расчетному многочлену, вычисление значимостей и вкладов элементов – все эти возможности реализованы в программном комплексе «АРБИТР» [8], и использовались в процессе решения оптимизационной задачи.

Согласно теореме разложения Поспелова [9], обобщившего теорему разложения Шеннона на произвольное количество логических переменных, любая произвольная функция алгебры логики может быть представлена в виде совершенной дизъюнктивной нормальной формы, особенностью которой является отсутствие повторов логических переменных в отдельных конъюнкциях ее образующих, взаимная попарная ортогональность конъюнкций, одинаковый ранг всех конъюнкций равный количеству логических переменных в функции алгебры логики [10]. Совершенная дизъюнктивная нормальная форма произвольной логической функции может быть записана в виде:

$$K_1 \vee K_2 \vee \ldots \vee K_s, \tag{4}$$

где $K_j = x_1 \wedge x_2 \wedge \ldots \wedge x_n$,

$K_j$ – элементарная конъюнкция ранга $n$, может содержать инверсии логических переменных.

Совершенная дизъюнктивная нормальная форма записи логической функции является так называемой формой прямого замещения, которая допускает непосредственный переход к вероятностной форме путем замены логических переменных вероятностями, а логических операций соответствующими арифметическими операциями. Логическая операция И заменяется на умножение, логическая ИЛИ – на операцию сложения. Таким образом, преобразовав логическую функцию в вероятностный многочлен, мы также не будем иметь повторов вероятностей элементов в пределах отдельных одночленов, т.е. вероятностный многочлен будет линейным однородным многочленом степени $n$ (линейной формой степени $n$). Отсюда следует что, частная производная вероятностной функции по любой переменной, не будет от этой переменной зависеть.

$$\frac{\partial R_S}{\partial p_i} = f\left(p_1, \dots p_{i-1}, p_{i+1}, \dots p_m\right)$$
$$\frac{\partial R_S}{\partial p_i}\left(p_1, \dots p_m\right) = const \tag{5}$$

где $R_s = f\left(p_1, \dots p_m\right)$ – вероятность безотказной работы, зависящая от $p_i$

В монографии И. А. Рябинина [11, 13] было доказано следующее равенство, а частная производная была названа значимостью элемента:

$$\xi_i = \frac{\partial R_s}{\partial p_i} = P\left\{\Delta_{x_i} f\left(X_m\right) = 1\right\}, \tag{6}$$

$$\Delta_{x_i} f\left(x_1, \dots, x_m\right) = f\left(x_1, \dots, x_i, \dots, x_m\right) \oplus f\left(x_1, \dots, x_i', \dots, x_m\right), \tag{7}$$

где $\Delta_{x_i} f$ булева разность,

$\xi_i$ значимость i-го элемента,

$\oplus$ – операция логического сложения по модулю 2.

Физический смысл значимости элемента, а значит и частной производной, есть количественная разность между вероятностью безотказной работы системы, в случае если вероятность безотказной работы элемента $x_i = 1$ и вероятностью безотказной работы системы, в случае если вероятность безотказной работы элемента $x_i = 0$.

Для нахождения значимости элемента (частной производной) можно использовать правило из работы [7]. Последовательно слева направо просматриваются все одночлены многочлена вероятностной функции и проверяется наличие в них параметров $p_i$ или $q_i$. Если в одночлене нет ни параметра $p_i$, ни параметра $q_i$, данный одночлен должен быть приравнен к нулю, если в одночлене есть прямой параметр $p_i$, то он заменяется на 1, если в одночлене есть обратный параметр $q_i$, то он заменяется на 0.

Следовательно, вычислительная сложность нахождения точной частной производной многочлена вероятностной функции по формуле (6) аналогична вычислению значения $R_S$ при произвольном векторе $\{p_i\}$, $i = 1, \dots, n$

Из (5) следует что, зная вероятность безотказной работы системы $R_S$ при некотором значении аргументов $\{p_1,..., p_i,\ldots, p_n\}$ можно легко вычислить значение $R_S^*$ при изменении одного аргумента и неизменных остальных $\{p_1,..., p_i^*,\ldots, p_n\}$ по следующей формуле:

$$R_S^* = R_S + \frac{\partial R_S}{\partial p_i}\Delta p_i = Rs + \xi\,\Delta p_i$$
$$R_S = R(p_1,\ldots, p_i,\ldots p_n),\quad R_S^* = R(p_1,\ldots, p_i^*,\ldots p_n) \tag{8}$$

Следовательно, вычисляя в начале каждого шага алгоритма значимости всех элементов системы (частные производные многочлена вероятностной функции по всем аргументам) можно легко вычислять значение надежности всей системы $R_S$ при изменении одного любого аргумента относительно текущего набора аргументов $\{p_i\}$.

В работе [7] вклад $B_i$ элемента $i$ в общую надежность системы определен как доля значимости элемента системы при изменении собственного показателя надежности элемента от текущего состояния до единицы:

$$\beta_i^+ = (1 - p_i)\xi_i = (1 - p_i)\frac{\partial P_c}{\partial p_i} \tag{9}$$

Таким образом, положительный вклад элемента в общую надежность системы численно равен приращению надежности всей системы, при увеличении надежности элемента от текущего значения – до максимального теоретически возможного – единицы [14]. Положительные вклады используются для выбора направления движения в алгоритме логико-вероятностных вкладов при возврате в область допустимых решений после нарушения ресурсных ограничений.

Связь приращения надежности элемента, его значимости и положительного вклада с приращением надежности всей системы показана в формуле (10). Вычислив значимости или положительные вклады элементов, мы сможем легко определить приращение надежности всей системы при увеличении надежности отдельного элемента (при фиксированных значениях надежности остальных элементов), не пересчитывая полностью вероятностный многочлен:

$$\Delta R_S = \xi_i \Delta p_i = \frac{\beta_i^+ \Delta p_i}{1 - p_i^0}, \tag{10}$$

где $\beta_i^+$ – вклад элемента $i$ в надежность (вероятность безотказной работы) всей системы;

$\Delta p_i$ – приращение вероятности безотказной работы элемента $i$ в результате его замены на более надежный элемент;

$p_i^0$ – начальная (до замены на более надежный элемент) вероятность безотказной работы элемента $i$.

## 3 Оптимизационный метод логико-вероятностных вкладов

На каждом шаге алгоритма происходит замена одного из компонентов (подсистемы) системы. Выбор компонента (подсистемы) для замены и элемента, на который он будет заменен, происходит одновременно.

Количественный показатель надежности (например, вероятность безотказной работы) и стоимость различных вариантов элементов упорядочим по возрастанию вероятности безотказной работы и пронумеруем, таким образом, что элемент с меньшим номером будет иметь меньшую надежность. Если в для качестве компонентов системы могут быть использованы резервированные подсистемы или комбинаторные подсистемы, работающие по принципу $k$ из $n$, то надежность и стоимость для всех возможных комбинаций элементного состава должна быть предварительно рассчитана. После чего все эти варианты должны быть отсортированы по возрастанию надежности или стоимости в зависимости от того, какой системный показатель оптимизируется (например, при оптимизации стоимости системы сортировка производится по стоимости).

Перед началом работы алгоритма оптимизации необходимо подготовить следующие исходные данные:

1. построить логическую функцию работоспособности для системы;
2. перейти от логической функции работоспособности к вероятностному многочлену для расчета системного показателя надежности;
3. если допускается использовать резервирование элементов подсистем или система содержит комбинаторные подсистемы, функционирующие по принципу $k$ из $n$, то для всех возможных вариантов элементного состава компонентов (подсистем) системы должны быть предварительно рассчитаны их надежность и стоимость;
4. варианты элементного состава для каждого компонента (подсистемы) системы должны быть отсортированы по возрастанию надежности или стоимости.

Опишем алгоритм оптимизации по шагам, в предположении, что оптимизируется стоимость системы, а ограничение задано на надежность системы.

Шаг 1. Вычисляются границы возможного изменения надежности системы, путем расчета надежности системы состоящей из самых ненадежных элементов $R_{\min} = R(X_{\min}) = R(X_1^1, \ldots, X_n^1)$ и системы состоящей из самых надежных элементов $R_{\max} = R(X_{\max}) = R(X_1^{z_1}, \ldots, X_n^{z_n})$,

где нижний индекс $(\overline{1,n})$ обозначает номер компонента (подсистемы),

верхний индекс обозначает номер варианта элемента использованного в качестве компонента (подсистемы),

$z_i$ – количество вариантов для компонента (подсистемы) $i$.

Если $R_{\min} < R < R_{\max}$, то решение существует, и в качестве текущего решения выбираем вариант $X_{\min} = (x_1^1,...,x_n^1)$.

Шаг 2. Вычисляем частные производные (значимости) по всем $n$ переменным входящим в вероятностную функцию.

Шаг 3. Для каждой подсистемы (элемента), для каждого более надежного, чем текущий, варианта построения подсистемы (номер элемента должен быть больше, чем текущий номер используемого элемента) вычисляется показатель $\alpha_j^i = \dfrac{\Delta p_j^i}{\Delta c_j^i} = \dfrac{p_j^i - p_j^*}{c_j^i - c_j^*}$, где $p_j^*, c_j^*$ – текущие надежность и стоимость элемента.

Для каждой подсистемы находим максимальное значение

$$\alpha_j^{\max} = \max_i \frac{\Delta p_j^i}{\Delta c_j^i} \tag{11}$$

Шаг 4. Находим компонент системы (подсистемы) и элемент (вариант элементного состава), для которого максимально следующее выражение:

$$\max_{i,j}\left(\frac{\Delta P_S}{\Delta C_S}\right) = \max_{i,j}\left(\frac{\Delta p_j^i}{\Delta c_j^i}\xi_j\right) = \max_j \alpha_j^{\max}\xi_j \tag{12}$$

где $j$ – номер компонента (подсистемы),
$i$ – номер элемента (или элементного состава) использованного в качестве компонента (подсистемы) $j$.

Таким образом, мы определяем компонент (подсистему) для которого будет произведена замена на данном шаге, и элемент на который будет произведена замена. Другими словами, на данном шаге происходит выбор направления движения и величина шага.

Проводим замену элемента и уточняем решение.

Шаг 5. Уменьшаем стоимость доступных средств на величину стоимости замены элемента найденного на предыдущем шаге (шаге 4 или 6). Если количество доступных средств $C_F \geq 0$, то переходим на шаг 3, иначе ($C_F < 0$) переходим на шаг 6.

Шаг 6. Находим компонент системы (подсистемы) и элемент (вариант элементного состава), для которого максимально следующее выражение:

$$\max_{i,j}\left((1 - p_j^i)\frac{\Delta P_S}{\Delta C_S}\right) = \max_{i,j}\left(\frac{\Delta p_j^i}{\Delta c_j^i}(1 - p_j^i)\xi_j\right) = \max_{i,j} \alpha_j^{\max}\beta_j^i \tag{13}$$

Таким образом, мы определяем компонент (подсистему) для которого будет произведена замена на данном шаге, и элемент на который будет произведена замена. Проводим замену элемента и уточняем решение. Если решение, полученное на данном шаге лучше ранее запомненного решения $x^*$, то сохраняем текущее решение в качестве $x^*$, если хуже – не сохраняем и переходим на шаг 5. Если решение, полученное на данном шаге, совпало с ранее запомненным решением $x^*$, то уменьшаем стоимость доступных средств на величину стоимости замены на найденный элемент и переходим на шаг 7.

Шаг 7. Если существуют более надежные элементы, стоимость замены на которые не превышает количество доступных средств $C \geq 0$, то находим элемент $i$ для которого $\max\limits_{i,j} \Delta P_S = \max\limits_{i,j}(\Delta p_j^i \xi_j)$. Проводим замену элемента и уточняем решение, уменьшаем стоимость доступных средств на величину стоимости замены на найденный элемент, остаемся на шаге 7. Если не существуют более надежные элементы, стоимость замены на которые не превышает количество доступных средств $C$, или $C = 0$, то текущее решение является решением оптимизационной задачи.

## 4 Преимущества и ограничения метода

Преимущества метода:

1. Метод позволяет найти точное решение оптимизационной задачи, в отличие от эвристических и приближенных методов.
2. Метод позволяет точно и с небольшими вычислительными затратами находить значение вероятностной функции при изменении надежности одного компонента системы при неизменных показателях надежности остальных, используя информацию о первых частных производных вероятностной функции.
3. Метод позволяет уменьшить вероятность нахождения локального экстремума за счет возможности нарушения ограничения в процессе решения задачи.

Ограничения метода:

Метод применим для оптимизации монотонного класса систем, для которых при увеличении надежности любого компонента системы надежность всей системы не убывает.

К ограничениям метода можно отнести необходимость предварительного расчета всех вариантов элементов и их сортировки для каждой подсистемы, в случае, когда для увеличения надежности разрешено резервирование или система содержит комбинаторные подсистемы, функционирующие по принципу $k$ из $n$.

## 5 Иллюстративные примеры

Рассмотрим задачу оптимизации для последовательно-параллельной системы в постановке (1). В такой же постановке задача рассматривается в статье Д. Коит и др.[1] Исходные данные задачи также взяты из работы [1].

Система состоит из шести последовательно соединенных подсистем, каждая подсистема функционирует по принципу $k$ из $n$. В примере, рассмотренном в статье [1], $s=6$, $m_i=10$ $(\forall i)$ и $n_{\max}=8$, и возможно более $6.9\times10^{27}$ различных вариантов построения системы.

Таблица 1. Надежность элементов для подсистем (компонентов) примера

| Подсистема | k | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 0.98 | 0.93 | 0.73 | 0.72 | 0.71 | 0.70 | 0.66 | 0.62 | 0.60 | 0.35 |
| 2 | 2 | 0.93 | 0.92 | 0.89 | 0.86 | 0.84 | 0.81 | 0.61 | 0.43 | 0.39 | 0.34 |
| 3 | 1 | 0.94 | 0.88 | 0.85 | 0.76 | 0.73 | 0.62 | 0.60 | 0.59 | 0.34 | 0.31 |
| 4 | 1 | 0.93 | 0.67 | 0.63 | 0.62 | 0.62 | 0.48 | 0.41 | 0.41 | 0.39 | 0.32 |
| 5 | 2 | 0.95 | 0.95 | 0.90 | 0.86 | 0.67 | 0.66 | 0.64 | 0.54 | 0.38 | 0.38 |
| 6 | 3 | 0.96 | 0.85 | 0.84 | 0.76 | 0.75 | 0.66 | 0.65 | 0.61 | 0.50 | 0.48 |

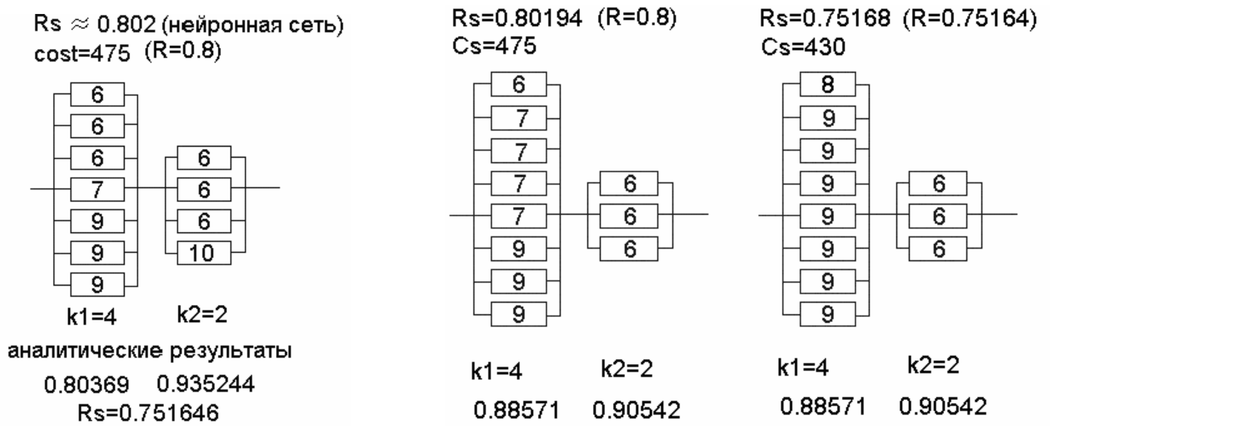
Таблица 2. Стоимость элементов для подсистем (компонентов) примера

| Подсистема | k | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 95 | 86 | 80 | 75 | 61 | 45 | 40 | 36 | 31 | 26 |
| 2 | 2 | 137 | 132 | 127 | 122 | 100 | 59 | 54 | 41 | 36 | 30 |
| 3 | 1 | 118 | 113 | 108 | 59 | 54 | 49 | 45 | 35 | 30 | 25 |
| 4 | 1 | 149 | 84 | 74 | 69 | 64 | 58 | 38 | 31 | 26 | 21 |
| 5 | 2 | 131 | 120 | 103 | 93 | 60 | 43 | 36 | 31 | 26 | 21 |
| 6 | 3 | 149 | 104 | 96 | 79 | 45 | 40 | 35 | 30 | 25 | 20 |

В работе 1 приведены решения для двух задач. Для первой задачи авторы [1] ограничиваются рассмотрением системы, которая состоит из последовательно соединенных первых двух подсистем. Общее количество вариантов для этой задачи составляет $1.9\times10^{9}$. В статье [1] для подсистем приведены как расчетные значения вероятности безотказной работы, так и значения, предсказанные нейронной сетью. Отличия в результатах точных расчетов и предсказанных нейронной сетью приводят к тому, что точное значение вероятности безотказной работы всей системы сильно отличается от целевого значения 0.8 и равно 0.752. Результаты расчетов вероятности безотказной работы методом ортогонализации для данного решения полностью совпали с точными результатами, приведенными в работе [1], что подтверждает правильность метода ортогонализации для расчета комбинаторных подсистем.

Задача оптимизации была решена методом логико-вероятностных вкладов для двух целевых значений системной надежности. В первом случае, в качестве целевой вероятности безотказной работы системы использовалось значение $R=0.8$. Во втором случае, использовалось точное значение вероятности безотказной работы для системы, приведенной в качестве решения этой задачи в работе [1], а именно $R=0.751646$. Ниже представлены полученные решения. Решение, полученное для $R=0.8$, обладает такой же стоимостью

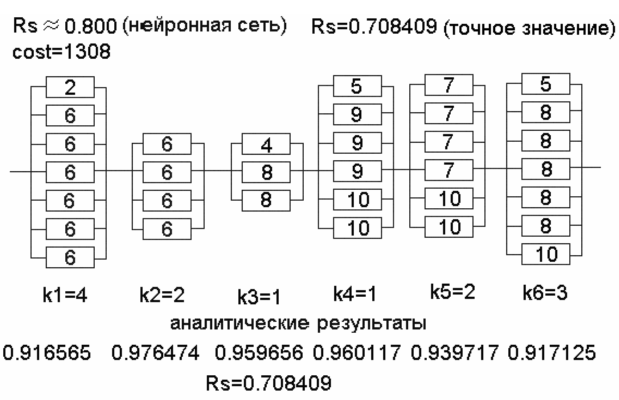
($C_S = 475$) как в работе [1], но большей надежностью, а решение, полученное для $R = 0.751646$, обладает меньшей стоимостью ($C_S = 430$).

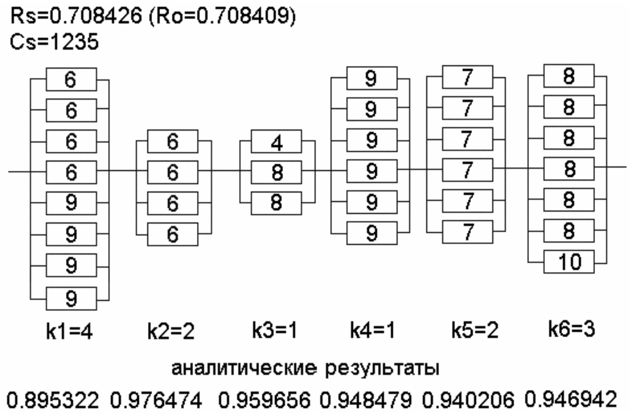



*Решение из работы [1]* *Решение методом вкладов для* $R \geq 0.8$ *и* $R \geq 0.75164$

Различие в решениях связано в первую очередь с тем, что в статье [1] вероятность безотказной работы подсистем $k$ из $n$ не вычисляется точно, а оценивается нейронной сетью. Нельзя не отметить близость элементного состава решений, что позволяет говорить о частичном совпадении результатов. Результаты решения задачи, полученные методом логико-вероятностных вкладов по сравнению с результатами из работы [1] обладают меньшей стоимостью при той же надежности, что позволяет говорить о превосходстве разработанного метода по сравнению с методом из [1] для данного класса задач.

По условиям второй задачи из работы [1] система состоит из шести последовательно соединенных подсистем. Общее количество вариантов для второй задачи составляет $6.9 \times 10^{27}$. Для решения, приведенного в статье Д. Коит и др. [1] было вычислено точное значение вероятности безотказной работы, которое затем использовалось в качестве целевого значения при решении задачи методом логико-вероятностных вкладов. Сравнение результатов также показало частичное совпадение элементного состава двух решений, но в целом расхождения элементного состава более значительны, чем для первой задачи. Это можно объяснить ростом погрешности в оценке нейронной сетью вероятности безотказной работы последовательной системы с ростом количества подсистем, что также отмечают и сами авторы статьи [1].




Результат из работы [1] Решение методом вкладов

В таблице 3 приведены решения для системы, состоящей из шести подсистем с разными требованиями к результирующей надежности системы – от 0.5 до 0.98 с шагом 0.02.

Таблица 3

| N | 1 | 2 | 3 | 4 | 5 | 6 | Po | P | C |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 9 9 9 9 9 9 9 9 | 6 6 6 | 8 8 8 | 9 9 9 9 10 | 7 7 7 7 10 | 8 8 10 10 10 10 10 | 0.50 | 0.50086 | 1000 |
| 2 | 9 9 9 9 9 9 9 9 | 6 6 6 | 8 8 8 | 9 9 9 9 10 | 7 7 7 7 7 | 5 10 10 10 10 10 10 10 10 | 0.52 | 0.52049 | 1020 |
| 3 | 7 9 9 9 9 9 9 9 | 6 6 6 | 8 8 8 | 9 9 9 9 9 | 7 7 7 7 7 | 8 8 8 10 10 10 10 10 | 0.54 | 0.54248 | 1039 |
| 4 | 7 7 9 9 9 9 9 9 | 6 6 6 | 8 8 8 | 9 9 9 9 9 | 7 7 7 7 7 | 8 8 8 8 10 10 10 10 | 0.56 | 0.56002 | 1058 |
| 5 | 6 7 7 9 9 9 9 9 | 6 6 6 | 8 8 8 | 9 9 9 9 9 | 7 7 7 7 7 | 8 8 8 8 8 10 10 10 | 0.58 | 0.58087 | 1082 |
| 6 | 7 7 7 7 9 9 9 9 | 6 6 6 | 8 8 8 | 9 9 9 9 9 10 | 7 7 7 7 7 | 8 8 8 8 8 10 10 10 | 0.60 | 0.60011 | 1107 |
| 7 | 6 6 6 7 9 9 9 9 | 6 6 6 | 8 8 8 | 9 9 9 9 9 10 | 7 7 7 7 7 | 8 8 8 8 8 8 10 10 | 0.62 | 0.62195 | 1132 |
| 8 | 6 6 6 6 9 9 9 9 | 6 6 6 | 5 8 8 | 9 9 9 9 9 10 | 7 7 7 7 7 | 8 8 8 8 8 8 10 10 | 0.64 | 0.64200 | 1156 |
| 9 | 6 6 6 6 6 6 9 9 | 6 6 6 | 5 8 8 | 9 9 9 9 9 10 | 7 7 7 7 7 | 8 8 8 8 8 8 10 10 | 0.66 | 0.66065 | 1184 |
| 10 | 6 6 6 9 9 9 9 9 | 6 6 6 6 | 5 8 8 | 9 9 9 9 9 10 | 7 7 7 7 7 | 8 8 8 8 8 8 10 10 | 0.68 | 0.68069 | 1201 |
| 11 | 6 6 6 6 9 9 9 9 | 6 6 6 6 | 4 8 8 | 9 9 9 9 9 9 | 7 7 7 7 7 | 8 8 8 8 8 8 10 10 | 0.70 | 0.70041 | 1225 |
| 12 | 6 6 7 7 7 7 9 9 | 6 6 6 6 | 8 8 8 8 | 9 9 9 9 9 9 | 7 7 7 7 7 | 8 8 8 8 8 8 8 10 | 0.72 | 0.72139 | 1254 |
| 13 | 6 6 6 6 6 6 8 9 | 6 6 6 6 | 8 8 8 8 | 9 9 9 9 9 9 | 7 7 7 7 7 | 8 8 8 8 8 8 8 10 | 0.74 | 0.74000 | 1279 |
| 14 | 6 6 6 6 6 7 9 9 | 6 6 6 6 | 8 8 8 8 | 9 9 9 9 9 9 | 7 7 7 7 7 7 | 8 8 8 8 8 8 8 10 | 0.76 | 0.76116 | 1305 |
| 15 | 2 6 6 6 9 9 9 9 | 6 6 6 6 | 8 8 8 8 | 9 9 9 9 9 9 | 7 7 7 7 7 7 | 8 8 8 8 8 8 8 8 | 0.78 | 0.78000 | 1333 |
| 16 | 1 6 6 6 9 9 9 9 | 6 6 6 6 | 8 8 8 8 | 9 9 9 9 9 9 10 | 7 7 7 7 7 7 | 8 8 8 8 8 8 8 8 | 0.80 | 0.80002 | 1363 |
| 17 | 1 1 7 9 9 9 9 9 | 6 6 6 6 | 8 8 8 8 | 9 9 9 9 9 9 9 | 7 7 7 7 7 7 | 8 8 8 8 8 8 8 8 | 0.82 | 0.82080 | 1399 |
| 18 | 1 1 7 7 9 9 9 9 | 6 6 6 6 | 8 8 8 8 | 9 9 9 9 9 9 9 | 7 7 7 7 7 7 | 5 5 8 8 8 8 8 8 | 0.84 | 0.84006 | 1438 |
| 19 | 1 1 6 6 9 9 9 9 | 6 6 6 6 | 5 8 8 8 | 9 9 9 9 9 10 10 | 7 7 7 7 7 7 | 5 5 8 8 8 8 8 8 | 0.86 | 0.86022 | 1483 |
| 20 | 1 1 1 1 9 9 | 6 6 6 6 | 5 8 8 8 | 9 9 9 9 9 9 9 9 | 7 7 7 7 7 7 | 5 5 8 8 8 8 8 8 | 0.88 | 0.88012 | 1531 |
| 21 | 1 1 1 1 9 9 | 6 6 6 6 | 8 8 8 8 8 | 9 9 9 9 9 9 9 9 | 7 7 7 7 7 7 10 | 5 5 5 8 8 8 8 8 | 0.90 | 0.90027 | 1583 |
| 22 | 1 1 1 1 9 9 | 6 6 6 6 7 | 8 8 8 8 8 | 9 9 9 9 9 9 9 9 | 7 7 7 7 7 7 7 | 5 5 5 5 8 8 8 8 | 0.92 | 0.92302 | 1667 |
| 23 | 1 1 1 1 1 | 6 6 6 6 6 | 8 8 8 8 8 | 9 9 9 9 9 9 9 9 | 7 7 7 7 7 7 7 | 5 5 5 5 5 5 8 10 | 0.94 | 0.94075 | 1725 |
| 24 | 1 1 1 1 1 | 6 6 6 6 6 | 8 8 8 8 8 8 | 1 9 9 9 9 | 7 7 7 7 7 7 7 | 5 5 5 5 5 5 5 8 | 0.96 | 0.96046 | 1830 |
| 25 | 1 1 1 1 1 9 | 6 6 6 6 6 | 8 8 8 8 8 8 8 | 1 9 9 9 9 9 9 10 | 7 7 7 7 7 7 7 7 | 5 5 5 5 5 5 5 5 | 0.98 | 0.98013 | 2020 |

Для решения этих задач предварительно были найдены все возможные варианты элементного состава для каждой подсистемы, рассчитана их надежность и результаты отсортированы по возрастанию надежности для каждой комбинаторной подсистемы. Количество вариантов для подсистемы 1 – 43471, для подсистемы 2 – 43746, для подсистемы 3 и 4 – 43756, для подсистемы 5 – 43746, для подсистемы 6 – 43691. Время предварительных расчетов и сортировки составило 17 минут 51 секунда на компьютере Intel Pentium M 1.6 МГц, 512 МБ ОЗУ. Для решения каждой оптимизационной задачи первая частная производная вычислялась по формуле (5) в среднем 320 раз, приращение надежности в результате замены подсистемы по формуле (9) рассчитывалось $1.27 \times 10^7$ раз. Среднее время решения одной задачи оптимизации составило 21 секунду.

## 6 Расчет надежности систем работающих по принципу k из n

Построение логической функции.

Логическая функция работоспособности системы функционирующей по принципу $k$ из $n$ может быть представлена совокупностью всех кратчайших путей успешного функционирования. Количество путей определяется количеством бесповторных сочетаний $k$ из $n$ вычисляемых по формуле

$$C_n^k = \frac{n!}{k!(n-k)!} \tag{14}$$

Пути успешного функционирования формируются методом формирования комбинаций, который изложен в [14]. Таким образом, логическая функция работоспособности комбинаторной подсистемы может быть записана в виде:

$$Y_{k/n} = \bigvee_{j=1}^{C_n^k} \Omega_j \tag{15}$$

где $\Omega$ – бесповторная конъюнкция ранга меньшего или равного $n$.

Для перехода от логической функции к вероятностному многочлену используем метод квазиортогонализации по одной переменной. Основное правило квазиортоганализации определяется следующим соотношением [14]:

$$A \cdot i \vee A \cdot B = A \cdot i \vee A \cdot B \cdot \bar{i} \tag{16}$$

Применение правила квазиортогонализации позволяет во всех случаях получить полностью ортогональную булеву функцию без увеличения общего числа ее конъюнкций [14]. Полностью ортогональная логическая функция является формой полного замещения, т.е. позволяет прямые логические переменные заменить вероятностями работоспособности, инверсные переменные – вероятностями неработоспособности, конъюнкции заменить умножением, а дизъюнкции заменить сложением, и, таким образом перейти от логической функции к вероятностному многочлену.

Рассмотрим пример построения логической функции работоспособности и вероятностного многочлена для подсистемы, работающей по принципу 3 из 5. Количество дизъюнкций в логической функции работоспособности, вычисленное по формуле (12) равно 10. В таблице 1 представлены кратчайшие пути успешного функционирования комбинаторной подсистемы. В таблице 2 записана логическая функция работоспособности, полученная из кратчайших путей успешного функционирования методом квазиортогонализации по одной переменной. Для получения логической функции работоспособности для комбинаторных систем, работающих по принципу k из n можно использовать простое мнемоническое правило. А именно, находится максимальный номер элемента входящего в кратчайший путь успешного функционирования, все входящие в путь элементы заменяются прямыми логическими переменными, а все элементы, которые не содержатся в данном пути и чей номер меньше максимального номера элемента для данного пути, заменяются инверсной логической переменной. Таким образом, можно легко получить логическую функцию работоспособности, которая является формой прямого замещения. В таблице 6 записан вероятностный многочлен для вероятности безотказной работы комбинаторной системы, полученный из записанной в таблице 5 логической функции. Логическая функция представлена в форме прямого замещения, позволяющей сразу перейти от логической функции к вероятностному многочлену.

Таблица 4

| $j$ | $C_{3/5} =$ |
|---|---|
| 1 | 1.2.3 |
| 2 | 1.2.4 |
| 3 | 1.2.5 |
| 4 | 1.3.4 |
| 5 | 1.3.5 |
| 6 | 1.4.5 |
| 7 | 2.3.4 |
| 8 | 2.3.5 |
| 9 | 2.4.5 |
| 10 | 3.4.5 |

Таблица 5

| $j$ | $Y_{3/5} =$ |
|---|---|
| 1 | $(x1 \wedge x2 \wedge x3) \vee$ |
| 2 | $\vee (x1 \wedge x2 \wedge \overline{x3} \wedge x4) \vee$ |
| 3 | $\vee (x1 \wedge x2 \wedge \overline{x3} \wedge \overline{x4} \wedge x5) \vee$ |
| 4 | $\vee (x1 \wedge \overline{x2} \wedge x3 \wedge x4) \vee$ |
| 5 | $\vee (x1 \wedge \overline{x2} \wedge x3 \wedge \overline{x4} \wedge x5) \vee$ |
| 6 | $\vee (x1 \wedge \overline{x2} \wedge \overline{x3} \wedge x4 \wedge x5) \vee$ |
| 7 | $\vee (\overline{x1} \wedge x2 \wedge x3 \wedge x4) \vee$ |
| 8 | $\vee (\overline{x1} \wedge x2 \wedge x3 \wedge \overline{x4} \wedge x5) \vee$ |
| 9 | $\vee (\overline{x1} \wedge x2 \wedge \overline{x3} \wedge x4 \wedge x5) \vee$ |
| 10 | $\vee (\overline{x1} \wedge \overline{x2} \wedge x3 \wedge x4 \wedge x5) \vee$ |

Таблица 6

| $j$ | $P_{3/5} =$ |
|---|---|
| 1 | $p1 \cdot p2 \cdot p3 +$ |
| 2 | $+ p1 \cdot p2 \cdot q3 \cdot p4 +$ |
| 3 | $+ p1 \cdot p2 \cdot q3 \cdot q4 \cdot p5 +$ |
| 4 | $+ p1 \cdot q2 \cdot p3 \cdot p4 +$ |
| 5 | $+ p1 \cdot q2 \cdot p3 \cdot q4 \cdot p5 +$ |
| 6 | $+ p1 \cdot q2 \cdot q3 \cdot p4 \cdot p5 +$ |
| 7 | $+ q1 \cdot p2 \cdot p3 \cdot p4 +$ |
| 8 | $+ q1 \cdot p2 \cdot p3 \cdot q4 \cdot p5 +$ |
| 9 | $+ q1 \cdot p2 \cdot q3 \cdot p4 \cdot p5 +$ |
| 10 | $+ q1 \cdot q2 \cdot p3 \cdot p4 \cdot p5 +$ |

## 6 Обсуждение результатов

Представлен разработанный метод решения задачи оптимизации надежности. Результаты решения иллюстративных примеров, взятых из работы [1], показали, что разработанный оптимизационный *метод логико-вероятностных вкладов* позволяет решать высокоразмерные задачи оптимального резервирования и распределения надежности. Решения, полученные методом логико-вероятностных вкладов, превосходят решения, полученных с использованием ранее разработанного в статье [1] метода. Поэтому представляется целесообразным дальнейшее развитие и апробация разработанного метода.

Применение для расчета надежности комбинаторных подсистем метода из работы Можаева А.С. [14], позволило точно и с небольшими вычислительными затратами рассчитать вероятность безотказной работы комбинаторных подсистем, что в свою очередь позволило применить оптимизационный метод логико-вероятностных вкладов для систем содержащих комбинаторные подсистемы k из n.

Использование первых частных производных функции надежности в процессе оптимизации позволяет находить решение оптимизационной задачи за небольшое число шагов, даже в случае задач большой размерности.

ЛИТЕРАТУРА


1. D.W. Coit and A.E. Smith, Solving the redundancy allocation problem using a combined neural network/genetic algorithm approach. *Computers Ops. Res. Vol.23, No. 6, pp 515-526, 1996*
2. D.W. Coit and A.E. Smith, Reliability Optimization of Series-Parallel Systems Using a Genetic Algorithm. *IEEE Transactions On Reliability, Vol. 45, No. 2, 1996*
3. M.S. Chern, On the computational complexity of reliability redundancy allocation in a series system. Ops. Res. Lett. 11, pp 309-315, 1992
4. W. Feller, An Introduction To Probability Theory, Wiley, New York, 1968
5. Рябинин И.А., Черкесов Г.Н. Логико-вероятностные методы исследования надежности структурно-сложных систем. М.: Радио и связь, 1981, - 286 с
6. Рябинин И.А. Надежность и безопасность сложных систем. СПб.: Политехника, 2000. –248с.
7. Можаев А.С. Общий логико-вероятностный метод анализа надежности структурно сложных систем. Уч. пос. Л.:ВМА, 1988. - 68 с.
8. Можаев А.С. Автоматизация моделирования систем ВМФ. Учебник для слушателей ВМФ. Часть 2. Автоматизированное структурно-логическое моделирование систем. ВМА, СПб, 2006
9. Можаев А.С. Программный комплекс автоматизированного структурно-логического моделирования сложных систем (ПК АСМ 2001). // Труды Международной Научной Школы 'Моделирование и анализ безопасности, риска и качества в сложных системах' (МА БРК – 2001). СПб.: Издательство ООО 'НПО 'Омега', 2001, с.56-61.
10. Поспелов Д.А. Логические методы анализа и синтеза схем. М.: Энергия, 1964
11. Рябинин И.А. Надежность и безопасность структурно-сложных систем. – СПб.: Изд-во С.-Петерб. ун-та, 2007. – 276 с.

12. Рябинин И.А., Парфенов Ю.М., Определение «веса» и «значимости» отдельных элементов при оценке надежности сложной системы // Известия АН СССР. Энергетика и транспорт, 1978, №6.
13. Ryabinin I.A. Reliability of Engineering Systems. Principles and Analysis. M.: Mir, 1976
14. Можаев А.С. Автоматизация моделирования систем ВМФ. Учебник для слушателей ВМА. СПб, 2006