УДК 681.322.01

Методика выбора программного обеспечения компьютерных сетей

© А.М. Андреев, Г.П. Можаров

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Рассмотрена процедура синтеза программного обеспечения, представляющая упорядочение набора программных модулей, реализующих множество операций. Определено описание этого набора как трансверсалей семейства алгоритмических признаков программы. Для нахождения наибольшей по весу частичной трансверсали семейства алгоритмических признаков программы используется жадный алгоритм. Подобный подход применен и для оценки надежности (вероятности связности случайного графа, представляемого случайным матроидом) программного обеспечения. Предложена модель надежности сети связи программных модулей.

Ключевые слова: *программное обеспечение, алгоритмические признаки, трансверсаль, частичная трансверсаль, оценка надежности, модель надежности.*

Введение. Основная проблема выбора программного обеспечения (ПО) связана с неполнотой знаний о возможной реализации функций компьютерной сети (КС) для выполнения программы $\mathcal{P}$. Для решения этой проблемы составляется некоторая модель ПО, позволяющая учесть наиболее существенные особенности структуры графа ПО [1–3].

При проектировании ПО необходимо реализовать множество операций (алгоритмов), которые целесообразно представить в виде совокупности более простых составляющих, называемых модулями, и удовлетворить ряду противоречивых требований, например по повышению надежности реализации программ, объему используемой оперативной памяти, уменьшению времени выполнения программ и загрузки каналов между центральным процессором и внешними запоминающими устройствами, по точности вычислений и т. д.

Пусть программное обеспечение $\mathcal{P}$ должно реализовать множество операций $\mathcal{O} = \left\{O_1, ..., O_N\right\}$. Под операцией (алгоритмом) будем понимать, например, решение систем алгебраических и дифференциальных уравнений различного типа, вычисление интегралов, поиск информации по заданным признакам и т. д.

Каждая i-я операция $O_i \in \mathcal{O}$ может быть реализована любой программой из заданного множества программ $P_i = \left\{P_{i1}, ..., P_{ir_i}\right\}$, $i = \overline{1,\ N}$. Каждая из программ P_i характеризуется надежностью, временем реализации, объемом требуемой памяти и/или другими характеристиками. Программное обеспечение $\mathcal{P}$ представляет собой упорядоченный набор программ $P_l = \left\{P_{1l_1},\ ...,\ P_{Nl_N}\right\}$, где P_{il_i} – программа из множества P_i, реализующая i-ю операцию $\mathcal{O}_i$. Поскольку множества P_i и P_j могут пересекаться, то на i-м и j-м местах в наборе P_l может стоять одна и та же программа.

Математическое описание представления алгоритмических признаков. Пусть $\mathcal{O} = \left\{O_1, ..., O_N\right\}$ – множество операций (алгоритмов), а $\mathcal{P} = \left\{P_1, ..., P_M\right\}$ – упорядоченное семейство таких множеств $\mathcal{O}$, что элемент $p_m \in P_m$, $m = \overline{1,\ M}$, где M определяет число алгоритмов (подпрограмм), которые должны быть реализованы ПО. Назовем p_m алгоритмическим признаком. Определим трансверсаль $p = \left(p_1, ..., p_M\right)$, т. е. систему различных представителей семейства $\mathcal{P}$, таким образом, что p_m – элемент множества P_m. Другими словами, трансверсаль – это произвольная последовательность $\left(p_1, ..., p_M\right)$, в которой $p_m \in P_m$ и $p_m \neq p_h$ для $m \neq h$. Частичной трансверсалью будем называть трансверсаль произвольного подсемейства семейства $\mathcal{P}$, т. е. $\overline{p} = \left(p_{i_1}, ..., p_{i_k}\right)$, где $1 \leq i_1 \leq\ ...\ \leq i_k \leq M$ [4–6].

Формально задача представления ПО сводится к отысканию отображения $\mathcal{O} \to \mathcal{P}$, которое задает разбиение множества $\mathcal{O}$ операций на M модулей, соответствующих функциям алгоритма из семейства $\mathcal{P}$.

Положим, требуется выбрать одно из нескольких ПО для решения конкретных алгоритмов. Пусть

$$P_1 = \left\{P_{11}, P_{12}\right\}, P_2 = \left\{P_{21}, P_{22}\right\} \text{ и } P_{11} = P_{21} = P_1^*,\ P_{12} = P_2^*,\ P_{22} = P_3^*.$$

Программа P_1^* реализует операции $\mathcal{O}_1$ и $\mathcal{O}_2$, P_2^* – только $\mathcal{O}_1$, а программа P_3^* – только $\mathcal{O}_2$. Рассмотрим два варианта $\mathcal{P}$:

$$P_1 = \left(P_1^*, P_2^*\right),\ P_2 = \left(P_2^*, P_3^*\right).$$

Предположим, что объем, занимаемый каждой из программ, равен 1, тогда объем, занимаемый программой P_1, равен 1, а объем программы P_2 – 2 (так как $\mathcal{P}\left(P_1\right) = \left\{P_1^*\right\}$, $\mathcal{P}\left(P_2\right) = \left\{P_2^*, P_3^*\right\}\Big)$.

Способ решения задачи проектирования программного комплекса состоит в построении такого множества $\mathcal{P}^* \subseteq \mathcal{P}$, что мощность $\mathcal{P}^*$ значительно меньше мощности $\mathcal{P}$, что облегчает решение задачи проектирования.

Предлагаемая методика включает: определение набора признаков, описывающих ПО, каждый признак имеет численную меру – значение; определение значений признаков, требуемых решаемых алгоритмов; определение степени соответствия значений признаков каждого ПО требованиям алгоритмов; определение относительной важности признаков с точки зрения решаемых алгоритмов.

Однако поиск соответствующего ПО – лишь начальный этап. Необходимо увязать этот процесс с отображением на ПО важных особенностей реализуемых программ. На начальных этапах выбора ПО вряд ли целесообразно каждый раз переписывать программы под конкретное ПО.

Будем считать ПО выбранным, если найдено его описание p и соответствующее ему разбиение $\mathcal{O}$ на модули $\mathcal{P}(\mathcal{O}) = (O_1, ..., O_M)$.

Основные этапы выбора программного обеспечения. Процедура выбора так или иначе связана с оптимизацией по совокупности критериев. Поэтому желательно алгоритмизировать и процессы порождения вариантов, и их оценку [7].

После выбора частичного описания $\overline{p}$ необходимо построить полное описание p признаков ПО. Эта процедура связана с видом разбиения множества $\mathcal{O}$. При этом одинаковые признаки могут оказаться в различных множествах $\{P_1, ..., P_M\}$, как это было показано выше. Более того, множества $\left(P_{i_1}, ..., P_{i_k}\right)$ могут содержать более одного элемента, т. е. не только признаки $\left(p_{i_1}, ..., p_{i_k}\right)$ частичного описания. Это объясняется тем, что такие множества могут быть расширены разработчиком за счет добавления к элементам $\overline{p}$ других, близких к ним по значимости признаков. Последовательностям $\left(\overline{P}_1, ..., \overline{P}_H\right)$ соответствуют разбиения $\left(P_1(\mathcal{O}), ..., P_H(\mathcal{O})\right)$, каждое из которых будем называть начальным разбиением спецификации $\mathcal{O}$. Число возможных описаний не равно числу последовательностей вида $\left(\tilde{p}_1, ..., \tilde{p}_M\right)$, где $\left(\tilde{p}_1, ..., \tilde{p}_M\right) \in \{P_1 \cup P_2 \cup ... \cup P_M\} = \left\{\tilde{p}^1, ..., \tilde{p}^m\right\}$, $\tilde{p}_1 \neq ... \neq \tilde{p}_M$. Последовательность $\left(\tilde{p}_1, ..., \tilde{p}_M\right)$ в общем случае не является трансверсалью признаков ПО.

Решение задачи выбора структуры ПО включает следующие этапы:

• нахождение частичного описания (построение начальных фрагментаций $\{P_1(\mathcal{O}), ..., P_H(\mathcal{O})\}$ и соответствующих последовательностей $(\overline{P}_1, ..., \overline{P}_H)$);

• поиск общей трансверсали признаков для H последовательностей $(\overline{P}_1, ..., \overline{P}_H)$ длины M семейства $\mathcal{P}$, причем общая трансверсаль признаков для семейства $(\overline{P}_1, ..., \overline{P}_H)$ не обязательно существует. Это означает, что на заданном семействе признаков не может быть реализовано ПО с требуемыми свойствами, следовательно, требуется возврат с этого этапа синтеза к предыдущему и изменение вида последовательностей $(\overline{P}_1, ..., \overline{P}_H)$ (возможно, придется изменить и начальные разбиения);

• построение разбиения $\mathcal{P}(\mathcal{O})$, соответствующего найденной общей трансверсали признаков. Необходимость этого этапа обусловлена тем, что семейство $\mathcal{O}' = \{O_1', ..., O_M'\}$, полученное на втором этапе, может содержать пересекающиеся подмножества, а также, возможно, что $\left|O_1' \cup \; ... \; \cup O_M'\right| \le |\mathcal{O}|$, т. е. некоторые объекты программы могут не попасть в подмножества семейства $\mathcal{O}'$. Поэтому следует расширить подмножества $\{O_1', ..., O_M'\}$ до подмножеств $\{\mathcal{O}_1, ..., \mathcal{O}_M\}$ семейства $\mathcal{O}'$ так, чтобы $\mathcal{O}_1 \cup ... \cup \mathcal{O}_M = \mathcal{O}$, а затем построить разбиение $\mathcal{P}(\mathcal{O})$.

Опишем формальные комбинаторные схемы для решения задачи частичного представления множества операций системой ПО.

Пусть $\mathcal{P} = (P_1, ..., P_M)$ – семейство подмножеств на конечном множестве $\mathcal{O}$ признаков ПО, а $\overline{p}$ – семейство частичных трансверсалей ПО. Пара $M = (\mathcal{O}, \overline{p})$ представляет собой матроид [9]. Введем функцию w: $\mathcal{O} \to R^+$, где R^+ – множество неотрицательных действительных чисел, а значение $w(O)$ есть вес элемента $O \in \mathcal{O}$. Упорядочим множество $\mathcal{O} = \{O_1, \; ..., \; O_q\} = \bigcup_{m=1}^{M} P_m$ по невозрастанию весов: $w(O_1) \ge ... \ge w(O_q)$.

Эта задача совсем нетривиальная, а для разнородных, программных компонентов ПО проблема еще больше усложняется [7–10]. Здесь важно, чтобы вес $w(O)$ элемента (алгоритмического признака) был инвариантен относительно будущей реализации ПО. Подобные допущения оправданы лишь при определенных условиях и на ранних стадиях выбора структуры ПО. Задача заключается в нахождении наибольшей по весу частичной трансверсали семейства $\mathcal{P}$, а поскольку $M = (\mathcal{O}, \overline{p})$ – матроид, то для этого можно применить жадный алгоритм.

Общая комбинаторная схема для нахождения частичной трансверсали $\overline{p} \subseteq \mathcal{O}$ с наибольшим весом такова. Просматриваются все q ребер (u_i, v_j) $(q = |\mathcal{O}|)$ двудольного графа H с вершинами $(u_1, ..., u_q)$, $(v_1, ..., v_M)$, причем $O_i \in P_m$, $i = 1, \ ..., \ q$. Находится паросочетание графа H. Затем отыскивается чередующаяся цепь относительно паросочетания с началом в O_i.

В результате применения жадного алгоритма получается наибольшая по весу $\sum\limits_{O \in \mathcal{P}} w(O)$ частичная трансверсаль $\overline{p}$.

Разбиение программы. Допустим, решена задача поиска общей трансверсали признаков ПО и выполнено расширение подмножеств $\{\mathcal{O}_1^{'}, ..., \mathcal{O}_M^{'}\}$ до подмножеств $\{O_1, ..., O_M\}$, причем $O_1 \cup ... \cup O_M = \mathcal{O}$. Процедура отыскания общего описания p архитектуры для H семейств признаков, основанная на поиске максимального нуль-единичного потока в сети $N = (V, B)$ (построение этой сети см. в [3–5]), вовсе не гарантирует получения разбиения спецификации $\mathcal{O}$ согласно найденному представлению. Теперь, на последнем этапе необходимо найти разбиение $\mathcal{P}(\mathcal{O})$, соответствующее описанию p. Это разбиение назовем окончательным определением семейства $\mathcal{O}^{'} = \{\mathcal{O}_1, ..., \mathcal{O}_M\}$ расширенных подмножеств.

Обозначим через $(\overline{O}_1, ..., \overline{O}_n)$ последовательность различных элементов пересечений множеств $\{O_1^{'}, ..., O_M^{'}\}$, а через $\mathcal{S} = \{S_1, ..., S_n\}$ – семейство непустых непересекающихся множеств. При этом имеют место биекции $S_j \to \widetilde{O}_j$, где $j = 1, ..., n$, $\widetilde{O}_j$ – совокупность пересекающихся множеств семейства $\mathcal{O}^{'}$, таких, что $\overline{O}_j$ – элемент пересечения, $\{\widetilde{O}_1, \ ..., \ \widetilde{O}_j, \ ..., \widetilde{O}_n\}$ не обязательно не пересекаются.

Рассмотрим q-элементное множество $\mathcal{O}$, для которого $\mathcal{S}$ является разбиением и $q = \sum\limits_{j=1}^{n} |S_j|$. Упорядочим элементы $\mathcal{O} = \{O_1, ..., O_q\}$ по невозрастанию неотрицательных вещественных весов: $w(O_1) \geq ... \geq w(O_q)$.

Задача оптимального окончательного определения заключается в том, чтобы найти подмножество $\overline{p} \subseteq \mathcal{O}$ с наибольшим весом $w\left(\overline{p}\right) = \sum_{O \in p} w\left(O\right)$, где $\left|\overline{p}\right| = n$. Это означает, что необходимо отобрать n подмножеств $\mathcal{O}$, пересечения которых образуют последовательность $\left(\overline{O}_1, ..., \overline{O}_n\right)$. В результате окончательного определения каждый элемент $\overline{O}_j$ должен быть закреплен только за одним из пересекающихся подмножеств $\widetilde{O}_j$ и исключен из других. При этом сумма весов подмножеств, за которыми закрепляются элементы $\left(\overline{O}_1, ..., \overline{O}_n\right)$, должна быть наибольшей.

Для оптимального определения разбиения необходимо: во-первых, выделить последовательность $\left(\overline{O}_1, ..., \overline{O}_n\right)$, все элементы которой различны; во-вторых, построить семейство $\mathcal{S}$; в-третьих, из множеств S_j семейства $\mathcal{S}$ отобрать n разных элементов, сумма весов которых будет наибольшей [7].

На первом этапе окончательного определения семейства выполним следующие операции. Найдем пересечение M множеств семейства $\mathcal{O}'$:

$$P_M = \left\{\overline{O}_1^M\right\},\ \overline{O}_1^M = \mathcal{O}_1 \bigcap\ ...\ \bigcap \mathcal{O}_M .$$

Далее найдем все пересечения $M-1$ множеств из $\mathcal{O}'$, число которых не превышает числа сочетаний $C_M^{M-i+1} = \left(M, M-1\right)$: $\overline{O}_1^{M-1}$,..., ,..., $\overline{O}_{(M,M-1)}^{M-1}$.

Построим разбиение вида $P_{M-1} = \left\{\overline{O}_1^{M-1} \setminus \overline{O}_1^M, ..., \overline{O}_{(M,M-1)}^{M-1} \setminus \overline{O}_1^M\right\}$. Рекуррентно продолжим построение разбиений, отыскивая на шаге $i \geq 3$ все пересечения $M-i+1$ множеств $\mathcal{O}'$. Их не более $C_M^{M-i+1} = \left(M, M-i+1\right)$, т. е. $\overline{O}_1^{M-i+1}$,..., $\overline{O}_{(M,\, M-i+1)}^{M-i+1}$. Рассмотрим некоторое пересечение $\overline{O}_s^{M-i+1}$, где $1 \leq s \leq \left(M, M-i+1\right)$. Пусть $\overline{O}_{s(M-i+2)}$ – объединение всех пересечений множеств $\mathcal{O}'$, найденных на (i – 1)-м шаге и участвующих в образовании пересечения $\overline{O}_s^{M-i+1}$ на шаге $i = 3,\ ...,\ M-1$. Выполним следующее разбиение:

$$P_{M-i+1} =$$
$$= \left\{\overline{O}_1^{M-i+1} \setminus \overline{O}_{1(M-i+2)}, ..., \overline{O}_s^{M-i+1} \setminus \overline{O}_{s(M-i+2)}, ..., \overline{O}_{(M,M-i+1)}^{M-i+1} \setminus \overline{O}_{(M-i+1)(M-i+2)}\right\}.$$

Поскольку индекс s изменяется от 1 до $\left(M,\ M-i+1\right)$, а число пересечений на (i − 1)-м шаге не превышает количества сочетаний C_M^{M-i+1}, то некоторые подмножества $\overline{O}_{s(M-i+2)}$ являются пустыми.

Наконец, найдем попарные пересечения $\left\{\overline{O}_1^2, ..., \overline{O}_{(M,2)}^2\right\}$, где $\left(M,\ 2\right)=C_M^2$, и построим разбиение вида

$$P_2=\left\{\overline{O}_1^2 \setminus \overline{O}_{13}, ..., \overline{O}_{(M,2)}^2 \setminus \overline{O}_{(M,2)3}\right\},$$

где $\left\{\overline{O}_{13},\ ...,\ \overline{O}_{(M,2)3}\right\}$ – объединения модулей разбиения пересечений из M по три тех подмножеств, которые участвуют в формировании попарных пересечений $\left\{\overline{O}_1^2,\ ...,\ \overline{O}_{(M,2)}^2\right\}$. Результатом первого этапа является семейство разбиений $P=\left\{P_{i_1},\ ...,\ P_{i_r}\right\}$ множества пересечений элементов семейства $\mathcal{O}'=\left\{\mathcal{O}_1,\ ...,\ \mathcal{O}_M\right\}$, где $2\le r\le G$, $1\le i_1\le ...\le i_r\le M-1$. На втором этапе пронумеруем элементы $P_{i_1}\cup\ ...\ \cup P_{i_r}$, определив последовательность $\left(\overline{O}_1,\ ...,\ \overline{O}_n\right)$, где $n=\left|P_{i_1}\right|+\ ...\ +\left|P_{i_r}\right|$. После первого этапа известны пересекающиеся множества, образующие $\widetilde{O}_j$, элементом которых является $\overline{O}_{j},\ j=1,\ ...,\ n$. Сравним каждый элемент $\overline{O}_j$ с множеством S_j, элементы которого находятся во взаимно однозначном соответствии с элементами множества $\widetilde{O}_j$. Если при этом одним и тем же пересекающимся множествам из $\mathcal{O}'$, содержащим $\overline{O}_j \ne \overline{O}_{j'}$, $j,\ j'\in\left\{1,\ ...,\ n\right\}$, соответствуют различные элементы $S_j,\ S_{j'}$, причем $S_j\cap S_{j'}=\varnothing$, то получаем разбиение $\mathcal{S}=\left(S_1,\ ...,\ S_n\right)$. На третьем этапе необходимо решить, какому элементу множества S_j соответствует элемент $\overline{O}_j$. Тогда в итоге объект спецификации $\overline{O}_j$ будет закреплен за одним, и только одним из множеств совокупности $\widetilde{O}_j$.

Использование для выбора надежного программного обеспечения класса случайных матроидов. Надежность ПО определяется как вероятность существования какого-либо множества из заданной системы подмножеств конечного множества, элементы которого отсутствуют независимо с фиксированными вероятностями. В настоящее время ее принято называть надежностью системы множеств, комбинаторной надежностью или даже полиномом надежности [3], так как это есть полином от надежностей ее элементов. В случае одинаковой надежности элементов мы имеем полином от одной переменной [5, 10].

Надежность системы множеств полностью определяется надежностью системы (подсистемы) ее минимальных по включению множеств. Такие множества не вложимы друг в друга. Как известно, систему множеств со свойством невложимости друг в друга называют клаттером [3]. Поэтому надежность системы множеств называют перколяционной вероятностью случайного клаттера [10–12].

Пусть программное обеспечение $\mathcal{P}$ должно реализовать множество операций $\mathcal{O}$, причем $\mathcal{P} = \left\{P_1, \; ..., \; P_M\right\}$ – семейство подмножеств множества $\mathcal{O}$ (необязательно непересекающихся и различных). Назовем каждое P_i из $\mathcal{P}$ программным модулем (ПМ). При выполнении условия Ф. Холла [9, 10] существует система различных представителей (трансверсаль) семейства $\mathcal{P}$, т. е. такое $p \subseteq \mathcal{O}, \; |p| = M$, что при некоторой нумерации $\left(O_1, \; ..., \; O_M\right)$ его элементов $O_i \in P_i$. Если $O \in p$ и $P \in \mathcal{P}$ имеют одинаковый индекс, мы скажем, что O есть представитель $P : O \to P$.

Известно [6, 9, 10], что множество всех трансверсалей системы ПМ $\mathcal{P}$ порождает на множестве $\mathcal{O}$ матроид $M\left(\mathcal{P}\right)$ как совокупность всех его баз. Определим в $\mathcal{O}$ случайное подмножество $\mathcal{F} \subseteq \mathcal{O}$, такое, что события $\left\{O \in \mathcal{F}\right\}$ независимы и имеют заданные вероятности. Пусть $R\left(\mathcal{P}, \; q\right)$ – вероятность того, что $\mathcal{F}$ содержит трансверсаль (где q – заданная вероятность отказа ребра случайного графа). Верхняя оценка надежности ПО может быть вычислена из следующего выражения

$$R\left(M, q\right) \le \prod_{i=1}^{r} \left(1 - q^{|C_i|}\right),$$

где при каждом $i = 1, ..., M$ множество коциклов $\mathcal{C} = \left\{C_1, \; ..., \; C_M\right\}$ порождено базой p [3, 5].

Пусть $p = \left\{P_1, ..., P_M\right\}$ – фиксированная трансверсаль, $O_i \in P_i$ и $\mathcal{C} = \left\{C_1, \; ..., \; C_M\right\}$ – семейство коциклов $M\left(\mathcal{P}\right)$, порожденное базой p. Коциклы C_k можно охарактеризовать как множество всех таких элементов $a \in \mathcal{O}$, что $p - O_k + a$ является базой [4, 5]. Коциклы из порождаемого базой p семейства $\mathcal{C}$ описываются следующим образом.

Элемент $a \in \mathcal{O}$ принадлежит коциклу $C_k \in \mathcal{C}$ тогда и только тогда, когда либо $a = O_k$, либо $a \notin P$ и существует последовательность $\left\{O_{i_1},\ \dots,\ O_{i_s}\right\} \subseteq p - O_k \quad \left(O_{i_j} \to P_{i_j}\right)$, такая что $a \in P_{i_1}$, $O_{i_j} \in P_{i_{j+1}}$ при $1 \le j \le s-1$, и $O_{i_s} \in P_k$.

Такая последовательность позволяет построить новое представительство:

$$a \to P_{i_1},\ O_{i_j} \to P_{i_{j+1}} \quad \text{при } 1 \le j \le s-1,\ O_{i_k} \to P_k,$$

для остальных $O_i \in p$, что является следствием метода чередующихся цепей [6], позволяющего находить коциклы в явном виде.

Верхнюю оценку $R\left\{\mathcal{P},\ q\right\}$ можно получить следующим образом. Присоединим к $\mathcal{O}$ множество индексов $I = \left\{1,\ \dots,\ M\right\}$, нумерующих модули в $\mathcal{P}$, и положим $P_i^{'} = P_i + i$. Тогда I есть трансверсаль семейства $\mathcal{P}^{'} = \left\{P_1^{'},\ \dots,\ P_M^{'}\right\}$, а порождаемое ею семейство коциклов есть само $\mathcal{P}^{'}$. Таким образом,

$$M\left(\mathcal{P}\right) = M\left(\mathcal{P}^{'}\right)\mathcal{O} \text{ и } R\left(\mathcal{P},\ q\right) \le \prod_{i=1}^{m}\left(1 - q^{\left|P_i\right|}\right).$$

Заключение. Синтез ПО определяет способ работы КС посредством разрешенного набора признаков p и разбиения программы $\mathcal{P}\left(\mathcal{O}\right)$ в соответствии с операциями (подпрограммами), реализуемыми ПО для выполнения задачи.

Рассмотренный подход к синтезу ПО включает в себя: поиск ее частичного описания с последующим построением полного набора признаков; отыскание общего описания для разных вариантов начального разбиения; построение разбиения ПО, соответствующего найденному описанию. Этот подход применен для оценки надежности (вероятности связности случайного графа, представляемого случайным матроидом) ПО.

Представленные комбинаторные схемы и методы не претендуют на полноту формализации процесса выбора ПО. Но они могут оказаться полезными при решении задач синтеза ПО.

ЛИТЕРАТУРА

[1] Андреев А.М., Березкин Д.В., Можаров Г.П., Свирин Ил.С. Математическое моделирование надежности компьютерных систем и сетей. *Вестник МГТУ им. Баумана. Сер. Приборостроение*, спец. выпуск «Моделирование и идентификация компьютерных систем и сетей», 2012, с. 3–46.

[2] Андреев А.М., Можаров Г.П., Сюзев В.В. *Многопроцессорные вычислительные системы: теоретический анализ, математические модели и применение*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2011, 334 с.

[3] Полесский В.П. Достижимые границы вероятности полного ранга случайного подматроида. *Пробл. передачи информ.*, 1995, т. 31, № 4, с. 81–99.

[4] Ковалев М.М. *Матроиды в дискретной оптимизации*. 2-е изд., Москва, Едиториал УРСС, 2003, 224 с.

[5] Степанов В.Е. Комбинаторная алгебра и случайные графы. *Теория вероятн. и ее примен.*, 1970, т. XIV, вып. 3, с. 393–420.

[6] Берж К. *Теория графов и ее применения*. Москва, Изд-во иностр. лит., 1962, 320 с.

[7] Топорков В.В. *Модели распределенных вычислений*. Москва, Физматлит, 2004, 320 с.

[8] Форд Л.Р., Фалкерсон Д.Р. *Потоки в сетях*. Москва, Мир, 1966, 276 с.

[9] Холл М. *Комбинаторика*. Москва, Мир, 1970, 424 с.

[10] Edmonds J., Fulkerson D.R. Bottlneck Extremal. J. *Comb. Theory*, 1970, vol. 8, pp. 299–306.

[11] Grimmet G.R. *Percolation Theory. Second edition. A Series of Comprehensive Studies in Math*. Berlin, Springer, 1999, 459 p.

[12] Shier D.R. Network *Reliabilty and Algebraic Structures*. Oxford, Claredon Press, 1991.

Статья поступила в редакцию 24.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Андреев А.М., Можаров Г.П. Методика выбора программного обеспечения компьютерных сетей. *Инженерный журнал: наука и инновации*, 2013, вып.11. URL: http://engjournal.ru/catalog/it/network/1055.html

Андреев Арк Михайлович родился в 1943 г., окончил МВТУ им. Н.Э. Баумана в 1967 г. Канд. техн. наук, доцент кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана. Автор более 100 научных работ в области вычислительных средств, систем управления и обработки сигналов. e-mail: arkandreev@gmail.com

Можаров Геннадий Петрович родился в 1945 г., окончил МВТУ им. Н.Э. Баумана в 1970 г. Канд. техн. наук, доцент кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана. Автор более 60 научных работ в области вычислительных средств, систем управления и обработки сигналов. e-mail: gmojarov@gmail.com