

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 24
2018
№ 7

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Гурарий М. М., Жаров М. М., Русаков С. Г., Ульянов С. Л. Минимаксная оптимизация в задачах схмотехнического проектирования 435
- Романов П. С., Романова И. П. Методика принятия решений по выбору наилучшей технической системы по набору ее характеристик 445
- Божко А. Н. Формализация размерных ограничений на проектные решения сборочного передела 454

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Гаевой С. В., Ахмед В. М. А., Фоменков С. А. Использование имитационного моделирования для определения показателей обслуживания кластера . . . 464

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Исаева О. С. Технология построения комплексных моделей в инфраструктуре имитационного моделирования 474
- Ефимов Н. О., Жилияков Е. Г. Распознавание фрагментов изображений рукописного текста. Характерные частотные интервалы 481
- Наумов А. С. Разработка программного обеспечения для реконструкции фрагментированных изображений 487

Главный редактор:

СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.
ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.
КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.
ПОПКОВ Ю. С.,
акад. РАН, д. т. н., проф.
РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.
РЯБОВ Г. Г.,
чл.-корр. РАН, д. т. н., проф.
СОЙФЕР В. А.,
акад. РАН, д. т. н., проф.
СОКОЛОВ И. А.,
акад. РАН, д. т. н., проф.
СУЕТИН Н. В., д. ф.-м. н., проф.
ЧАПЛЫГИН Ю. А.,
акад. РАН, д. т. н., проф.
ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.
ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.
ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.
АНТОНОВ Б. И.
БАРСКИЙ А. Б., д. т. н., проф.
ВАСЕНИН В. А., д. ф.-м. н., проф.
ВАСИЛЬЕВ В. И., д. т. н., проф.
ВИШНЕКОВ А. В., д. т. н., проф.
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.
ДОМРАЧЕВ В. Г., д. т. н., проф.
ЗАБОРОВСКИЙ В. С., д. т. н., проф.
ЗАРУБИН В. С., д. т. н., проф.
КАРПЕНКО А. П., д. ф.-м. н., проф.
КОЛИН К. К., д. т. н., проф.
КУЛАГИН В. П., д. т. н., проф.
КУРЕЙЧИК В. В., д. т. н., проф.
ЛЬВОВИЧ Я. Е., д. т. н., проф.
МАРТЫНОВ В. В., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.
НЕЧАЕВ В. В., к. т. н., проф.
ПОЛЕШУК О. М., д. т. н., проф.
САКСОНОВ Е. А., д. т. н., проф.
СОКОЛОВ Б. В., д. т. н., проф.
ТИМОНИНА Е. Е., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)
ФОМИЧЕВ В. А., д. т. н., проф.
ШИЛОВ В. В., к. т. н., доц.

Редакция:

БЕЗМЕНОВА М. Ю.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.
Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

INFORMATION TECHNOLOGIES

INFORMACIONNYYE TEHNOLOGII

Vol. 24
2018
No. 7

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

CAD-SYSTEMS

- Gourary M. M., Zharov M. M., Rusakov S. G., Ulyanov S. L.** Minimax Optimization in Circuit Design Problems 435
- Romanov P. S., Romanova I. P.** Decision-making Methodology for Choosing the Best Technical System for the Recruitment of its Characteristics 445
- Bozhko A. N.** Formalization of Dimensional Constraints on Design Decisions of the Assembly Stage 454

COMPUTING SYSTEMS AND NETWORKS

- Gaevoy S. V., Ahmed W. M. A., Fomenkov S. A.** Using Simulation to Determine Quality of Service of a Cluster System 464

INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Isaeva O. S.** The Technology of Building Complex Models in the Simulation Modeling Infrastructure 474
- Efimov N. O., Zhiljakov E. G.** Handwritten Text Fragment Recognition. Typical Subband Intervals 481
- Naumov A. S.** Fragmented Image Reconstruction Software Development 487

Editor-in-Chief:

Stempkovsky A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.
Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.
Popkov Yu. S., Member of RAS,
Dr. Sci. (Tech.), Prof.
Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Soifer V. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.
Chaplygin Yu. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.
Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Vasiliev V. I., Dr. Sci. (Tech.), Prof.
Vishnekov A. V., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kureichik V. V., Dr. Sci. (Tech.), Prof.
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Martynov V. V., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Poleschuk O. M., Dr. Sci. (Tech.), Prof.
Saksonov E. A., Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Timonina E. E., Dr. Sci. (Tech.), Prof.
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.
Lysenko A. V.
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ CAD-SYSTEMS

УДК 519.812.3 004.942 519.876.5

DOI: 10.17587/it.24.435-444

М. М. Гурарий, канд. техн. наук, ст. науч. сотр., e-mail: gourary@yandex.ru,
М. М. Жаров, канд. техн. наук, вед. науч. сотр., e-mail: zarov@ippm.ru,
С. Г. Русаков, д-р техн. наук, гл. науч. сотр., e-mail: rusakov@ippm.ru,
С. Л. Ульянов, д-р техн. наук, гл. науч. сотр., e-mail: ulyas@ippm.ru,
Институт проблем проектирования в микроэлектронике РАН, 124365, Москва

Минимаксная оптимизация в задачах схемотехнического проектирования

Рассмотрены направления совершенствования методов минимаксной оптимизации при решении задач проектирования, включающие: способ задания частных критериев в виде произвольной кусочно-линейной выпуклой функции; использование особенностей задачи и алгоритмов схемотехнического моделирования для ускорения процедур оптимизации; принципы построения алгоритма решения линейной минимаксной задачи на шаге оптимизации с учетом возможной многокритериальности.

Ключевые слова: автоматизация проектирования, минимаксная оптимизация, схемотехническое моделирование, линейное программирование, аналоговая схема, итерационные методы, многокритериальные задачи

Введение

В настоящее время разработка интегральных схем выполняется с применением средств САПР. Разработаны и широко внедрены разнообразные программные продукты, пользуясь которыми разработчик схем может получать значения показателей качества схемы, задавая параметры компонентов схемы, технологического процесса, режимов работы и внешних условий ее функционирования [1, 2]. В рамках этих программ у разработчика имеется также возможность получать зависимости показателей работы схемы от ее параметров. Однако проектирование требует значительного времени ввиду необходимости его многократного повторения до удовлетворения заданных требований к схеме.

По мере увеличения степени интеграции БИС все более актуальной становится задача организации автоматического направленного изменения параметров для обеспечения удовлетворения заданным требованиям в отношении показателей качества схемы. Поиск таких параметров может быть осуществлен на основе алгоритмов численной оптимизации [3, 4].

В настоящее время известно много работ, посвященных разработанным или предлагаемым системам проектирования аналоговых схем на основе оптимизационного подхода. Эти системы различаются математической формулировкой оптимизационных задач.

Классическая постановка задачи, когда один из качественных показателей схемы используется как критерий, а остальные показатели — как ограничения, применяется в IDAC3 [5], GPCAD [6], MOJITO [7]. Ряд систем используют критерий в виде взвешенной суммы показателей: OPASYN [8], ASTRX/OBLX [9], FASY [10]. Здесь все показатели включаются в критерий, а ограничения в явном виде не задаются. В системах FPAD [11], OPTIMAN [12], MAELSTROM [13], gaRFeeld [14] часть показателей не включены в критерий, а учитываются в ограничениях. В большинстве перечисленных систем для оценки значений показателей используются явные формульные представления моделей схемы. В то же время среди существенно более мощных и универсальных систем, основанных на методах электрического моделирования, преобладает минимаксный (или близкий ему) критерий. К таким систе-

мам можно отнести EF2ef [15], $\Sigma\Delta$ Modulators [16], DELIGHT.SPICE [17].

Минимаксная оптимизация состоит в выборе параметров схемы, обеспечивающих наибольшее превышение требуемого уровня показателя над его фактическим значением для того показателя, у которого это значение наименьшее. Такой критерий обладает следующими преимуществами над другими известными критериями.

1. По сравнению с заданием критерия в виде аналитической функции (например, суммы квадратов) от недопустимых отклонений минимаксный критерий имеет более ясный смысл для разработчика. Результат "схема удовлетворяет всем требованиям с запасом не менее 10 %" понятнее, чем "среднеквадратичный запас по требованиям равен 10 %".

2. По сравнению с выбором одного показателя как критерия, а остальных в виде ограничений минимакс при невозможности удовлетворения всех требований получает полезный результат в виде отрицательного запаса.

3. По сравнению с комплексным критерием в виде аналитического выражения минимаксный критерий при возможности удовлетворения ограничений на показатели всегда достигает этого.

Используемые в существующих системах алгоритмы минимаксной оптимизации основаны на известных математических методах решения таких задач [18–20]. Эти методы не учитывают особенности расчета показателей схемы с помощью алгоритмов моделирования, поэтому вычислительные затраты часто оказываются избыточными. В математических работах по методам оптимизации явно или неявно предполагается, что затраты на вычисление целевой функции и ограничений несущественны, и поэтому общие затраты определяются числом шагов и собственными затратами алгоритма оптимизации. Отсутствуют работы, в которых бы рассматривалось построение алгоритмов с существенными и разными затратами на вычисление показателей, зависящими от точности их расчета и т.п. Поэтому одной из задач данной работы является разработка направлений развития минимаксных алгоритмов оптимизации с учетом характеристик используемых алгоритмов схемотехнического моделирования.

В работе также предложено распространение известной линейной формы задания част-

ных критериев на случай произвольной кусочно-линейной выпуклой функции. Рассматриваются условия представления минимаксной задачи в многокритериальной форме, приводится описание алгоритма ее решения для линейного случая.

1. Постановка задачи минимаксной оптимизации при проектировании

Задача проектирования электронной схемы состоит в таком выборе ее конфигурации и параметров, чтобы наилучшим образом удовлетворить требования, предъявляемые к схеме. Для этого необходимо определить единый числовой критерий, который позволил бы проводить сравнение различных проектных решений. При этом предполагаются известными:

а) набор показателей схемы: $y = \{y_1, y_1, \dots, y_M\}$;

б) направление изменения (увеличение или уменьшение) каждого из показателей, приводящее к улучшению качества схемы, для определенности будем полагать в дальнейшем, что это — уменьшение;

в) предельно-допустимые значения показателей:

$$y_i \leq \bar{y}_i, i = 1, \dots, M; \quad (1)$$

г) математическая модель схемы, определяющая вектор показателей $y = \tilde{y}(d)$ по заданному вектору параметров: $d = \{d_1, d_2, \dots, d_N\}$.

Имеются два основных варианта постановки оптимизационной задачи:

— оптимизация одного из показателей при ограничениях на остальные:

$$\tilde{y}_k(d) \rightarrow \min, \tilde{y}_i(d) \leq \bar{y}_i \text{ при } i \neq k; \quad (2)$$

— оптимизация комплексного критерия в виде скалярной функции φ :

$$\varphi(\tilde{y}(d)) \rightarrow \min. \quad (3)$$

Применение первого подхода не всегда является эффективным, так как он приводит к тому, что большинство значений показателей находятся на границе допуска. В то же время использование второго подхода на основе выражения (3) при достаточно удачном выборе целевой функции позволяет получать реше-

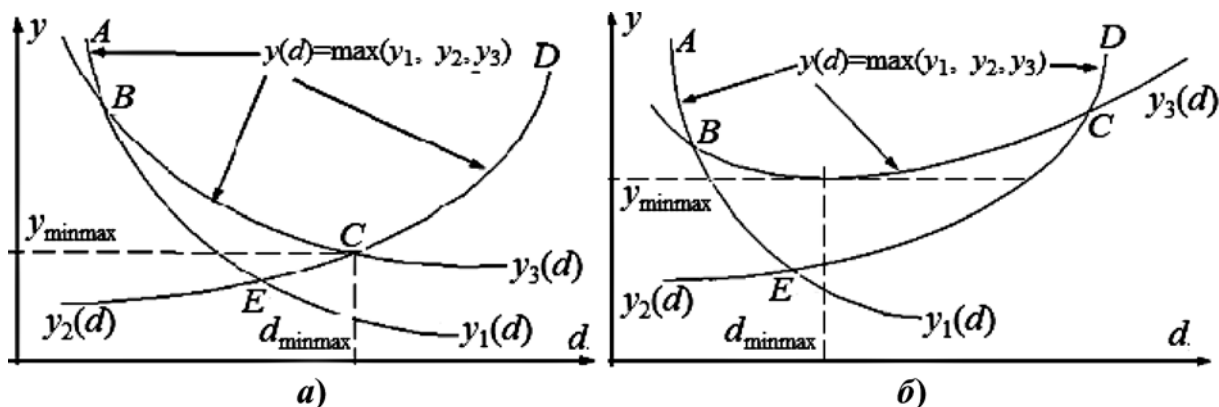


Рис. 1. Одномерные минимаксные задачи с минимумом в точке пересечения критериев (а) и с минимумом по одному показателю (б)

ние, соответствующее улучшению всех качественных показателей. В рамках второго подхода определение функции целесообразно проводить в два этапа. Сначала для каждого показателя определяется его индивидуальная целевая функция (частный критерий), отражающая степень влияния данного показателя на улучшение качества схемы. Эта функция преобразует значение показателя к безразмерной величине и имеет фиксированное (обычно нулевое) значение при предельно-допустимом значении показателя $\varphi_i(\bar{y}_i) = 0$.

Определив частные критерии, можно выбрать какое-либо симметричное выражение от этих величин в качестве целевой функции задачи (3). Для минимаксной задачи оптимизации выражение (3) принимает вид:

$$\max_{1 \leq i \leq M} (\varphi_i(\bar{y}(d))) \rightarrow \min. \quad (5)$$

Отметим, что алгоритмы минимаксной оптимизации более сложные и трудоемкие, чем алгоритмы оптимизации гладких функций. Однако вычислительные затраты в большинстве случаев можно сокращать по сравнению с затратами на гладкую оптимизацию. Объясним это на иллюстративном примере. На рис. 1, а показана минимаксная задача для трех функций одной переменной. Функция максимума $\max(y_1(d), y_2(d), y_3(d))$ представляет собой ломаную линию А—В—С—D, а минимум этой функции находится в точке С — пересечение кривых $y_2(d)$ и $y_3(d)$. Другой вариант такой задачи показан на рис. 1, б, где минимум функции максимума является локальным минимумом функции $y_3(d)$. В этом случае точка минимума определена менее четко, и ее на-

хождение требует большего числа итераций. Такой случай всегда реализуется для критерия в виде аналитического выражения, а для минимаксной задачи вариант рис. 1, б на практике встречается редко, так как обычно каждый из показателей монотонно зависит от параметров, и оптимальное решение представляет собой компромисс между различными требованиями, как это имеет место на рис. 1, а. Поэтому при вычислении функции максимума на шаге оптимизационного процесса можно во многих случаях не вычислять все значения показателей, а ограничиться лишь теми, прогнозируемые значения которых близки к критическому значению. Можно предложить и другие приемы для снижения трудоемкости процесса оптимизации, которые предполагается рассмотреть далее. Конечно, использование подобных особенностей минимаксной задачи приводит к усложнению алгоритма получения решения. Однако при вычислении показателей высокотратными методами схемотехнического моделирования эти усложнения представляются оправданными.

2. Определение частных критериев

Эффективность решения задач проектирования при использовании минимаксного подхода зависит от выбора коэффициентов в выражениях для частных критериев. Эти коэффициенты должны задавать относительную значимость требований к различным показателям. Методика определения коэффициентов, предложенная в работе [17], получила наибольшее распространение. Она используется также в работах [9, 13] и состоит в следующем.

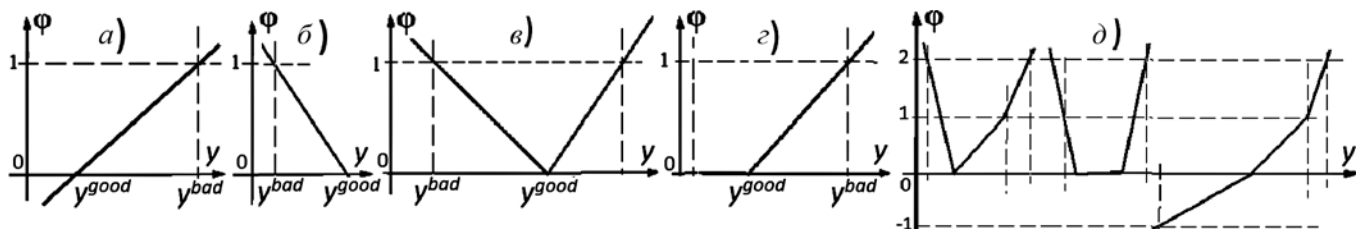


Рис. 2. Вид частных критериев на основе "хороших" и "плохих" значений: возрастающая (а) и убывающая (б) линейные зависимости, двустороннее отклонение (в), "мягкое" ограничение (з), предлагаемые произвольные выпуклые критерии (д)

Пользователь задает для каждого показателя два значения, которые называются "хорошее" и "плохое" ($y_i^{\text{good}}, y_i^{\text{bad}}$). Предполагается, что "хорошее" значение соответствует предельному значению для данного показателя, а "плохое" значение — определенному ухудшению качества схемы, причем степень этого ухудшения одинакова для "плохого" значения любого показателя. Тогда частный критерий для i -го показателя примет вид:

$$\varphi_i(y_i) = (y_i - y_i^{\text{good}})/(y_i^{\text{good}} - y_i^{\text{bad}}). \quad (6)$$

Частный критерий (6) принимает нулевое значение при "хорошем" значении показателя и единичное значение при "плохом", и улучшение качества схемы при изменении одного показателя всегда соответствует уменьшению частного критерия независимо от того, какое при этом должно быть изменение значения самого показателя. Эти свойства проиллюстрированы на рис. 2, а, б.

Если требуется максимальная близость показателя к заданному значению, то можно задать два "плохих" значения, соответствующих отклонениям от "хорошего" значения в большую и меньшую стороны. Тогда частный критерий примет кусочно-линейный вид, показанный на рис. 2, в, что можно записать как

$$\varphi_i(y_i) = \max((y_i - y_i^{\text{good}})/(y_i^{\text{good}} - y_i^{\text{bad}1}), (y_i - y_i^{\text{good}})/(y_i^{\text{good}} - y_i^{\text{bad}2})). \quad (7)$$

Кусочно-линейный вид частного критерия возможен также при отнесении показателя к "мягким" [17] ограничениям, при которых показатель исключается из минимаксного критерия после достижения своего "хорошего" значения. Это эквивалентно заданию частного критерия в кусочно-линейном виде (рис. 2, з).

Мы полагаем, что целесообразно дать пользователю возможность задавать частные кри-

терии не только в виде (6), (7), но и в более общем виде выпуклой кусочно-линейной зависимости, примеры которой представлены на рис. 2, д. Для этого можно отказаться от однозначного соответствия между индексом критерия и индексом показателя — $\varphi_i(y)$, а допустить задание пользователем нескольких частных критериев для одного показателя.

Таким образом, предлагается для каждого k -го частного критерия ψ_k задавать следующие параметры: номер показателя $I(k)$, уровень хорошего значения lev_k (необязательно 0), а также "хорошее" и "плохое" значения показателя $I(k)$: $y_{I(k)}^{\text{good}}, y_{I(k)}^{\text{bad}}$. Тогда k -й критерий (6) можно записать в следующем виде:

$$\psi_k(y_{I(k)}) = lev_k + (y_{I(k)} - y_k^{\text{good}})/(y_k^{\text{good}} - y_k^{\text{bad}}). \quad (8)$$

Максимум из функций (8) можно вычислить, предварительно сгруппировав функции, относящиеся к одному (i -му) показателю:

$$\begin{aligned} & \max_{1 \leq k \leq K} (\psi_k(y_{I(k)}(d))) = \\ & = \max_{1 \leq i \leq M} \left(\max_{k: I(k)=i} (\psi_k(y_i(d))) \right) = \\ & = \max_{1 \leq i \leq M} (\varphi'_i(y_i(d))). \end{aligned} \quad (9)$$

Таким образом, частный критерий φ'_i по показателю i определен в (9) как выпуклая оболочка заданных для него линейных критериев ψ_k . Поэтому может быть реализован любой из вариантов, показанных на рис. 2, д.

3. Методы минимаксной оптимизации и направления их совершенствования для задач схемотехнического проектирования

Для решения локальной минимаксной задачи используются как методы, специально предназначенные для ее решения, так и общие

методы нелинейного программирования. Сведение минимаксной задачи к задаче нелинейного программирования выполняется следующим образом [18].

Пусть минимаксная задача записана в форме (5). Введем дополнительную переменную d_0 , включенную в вектор d , которая всегда должна быть равна функции максимума $d_0 = \max_{1 \leq i \leq M} (\varphi_i(d))$. Тогда задачу (5) можно переписать как

$$f_0(d) = d_0 \rightarrow \min_d \text{ при } f_i(d) = \varphi_i(d) - d_0. \quad (10)$$

К наиболее эффективным методам нелинейного программирования для задач общего вида относятся методы спуска [18, 19]. К методам спуска относятся методы возможных направлений, условного градиента, проекции градиента, метод линейризации [18]. Имеется много вариантов этих методов, общая схема которых реализует последовательное приближение к оптимуму. Эта схема включает вычисление целевой функции, ограничений, а также их частных производных по оптимизируемым переменным. Линейризованные функции и ограничения шага формируют задачу линейного программирования, решение которой соответствует направлению улучшения целевой функции при выполнении ограничений. Направление поиска следующей точки для нелинейной минимаксной задачи (10) определяется из решения линейной минимаксной задачи с линейными ограничениями на значение шага:

$$\max_{1 \leq i \leq M} \left(\varphi_i(d^{(n)}) + \frac{\partial \varphi_i}{\partial d} h \right) \rightarrow \min_h \quad (11)$$

при $h_i \leq h^{\max}$.

Принцип определения следующей точки на основе решения задачи (11) для одномерной задачи в том случае, когда ограничения на значение шага не критичны, показан на рис. 3, где рассматриваются три функции скалярной переменной d — $\varphi_1(d)$, $\varphi_2(d)$, $\varphi_3(d)$. В точке n -й итерации $d^{(n)}$ проводится линейризация задачи, т. е. определяются касательные к кривым $\varphi_i(d)$. Решением линейной минимаксной задачи является точка C , в которой значение параметра ($d^{(n+1)}$) определяет следующую итерацию.

Из рис. 3 видно, что такой способ решения минимаксной задачи вблизи минимакса ана-

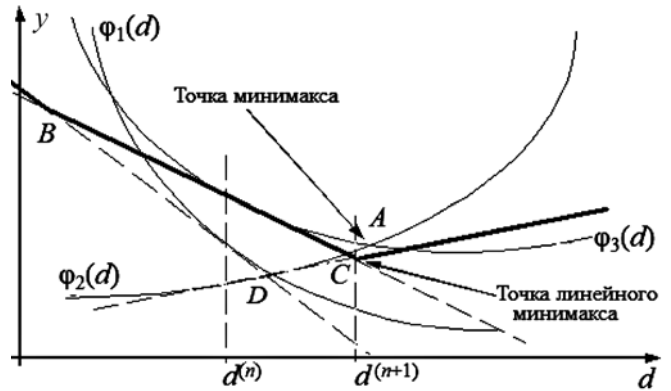


Рис. 3. Иллюстративный пример решения одномерной минимаксной задачи с помощью линейризации

логичен решению уравнения $\varphi_2(d) = \varphi_3(d)$ методом Ньютона. Как известно, метод Ньютона обладает квадратичной скоростью сходимости, которая не достигается в большинстве оптимизационных методов для гладких критериев. Это подтверждает утверждение, что результирующие затраты на решение минимаксной задачи при разумной организации алгоритма ниже затрат на гладкую оптимизацию.

Таким образом, метод последовательного решения линейризованных минимаксных задач должен быть выбран в качестве базисного при разработке общих алгоритмов нелинейной минимаксной оптимизации.

Для разработки наиболее эффективных алгоритмов оптимизации на основе выбранного подхода необходимо учитывать вычислительные затраты как на вычисление функций, входящих в минимаксный критерий, так и на выполнение операций самого алгоритма оптимизации. При разработке различных вариантов метода спуска [20] обычно учитывают только затраты на последнем шаге. Основные усилия направляют на сокращение числа итераций (11) и снижение размерности линейной задачи на каждой итерации.

Однако вычисление значений показателей методами электрического моделирования требует, как правило, значительно большего времени, чем решение даже сравнительно высоко-размерной задачи ЛП, поэтому вряд ли стоит прилагать большие усилия для снижения ее размерности. Что касается сокращения числа итераций, то следует учесть, что на каждой итерации вычисления показателей выполняются не только при определении направления спуска, но и при сокращении шага, поэтому

необходимо принимать во внимание общие затраты на всех шагах.

Помимо затрат на моделирование схемы при разработке алгоритмов оптимизации следует учесть другие особенности алгоритмов электрического моделирования: необходимость проведения отдельных реализаций процесса моделирования для различных групп выходных показателей при заданных значениях параметров; существенный разброс в затратах на различные процедуры моделирования; возможность задания точности расчетов и зависимость от нее времени моделирования; наличие моделей разного уровня с различной точностью и затратами на моделирование.

Для того чтобы в максимальной степени учесть эти особенности для снижения затрат, предполагается включить в разрабатываемые алгоритмы реализацию следующих принципов.

Адаптивность алгоритмов выбора шага. Так как затраты на моделирование, как правило, намного превосходят возможные затраты на операции самого алгоритма, то представляется целесообразным обеспечить хранение всей ранее полученной информации и при выборе очередного шага учитывать предыдущие результаты для наиболее рационального выбора. Многие существующие алгоритмы включают принцип адаптивности, но ограничиваются обработкой информации от последних одного-двух шагов. Здесь предлагается существенно увеличить объем хранимой и обрабатываемой информации.

Асинхронность вычисления показателей. Зависимости показателей, входящих в минимаксный критерий, от параметров, как правило, имеют весьма разную степень нелинейности. Поэтому можно предполагать, что при решении линеаризованной минимаксной задачи для определения следующей точки можно заменить вычисление слабонелинейных показателей и их частных производных применением их линеаризованных зависимостей, полученных в предыдущей точке. Допустимость такого использования зависит также от близости значения этого показателя к текущему минимаксному значению, от близости текущей точки к оптимуму и от вычислительных затрат на получение значения показателя. Должны быть разработаны критерии, учитывающие все приведенные факторы, и проведено их теоретическое и экспериментальное исследование.

Управление точностью моделирования. Текущее значение показателя и его производных целесообразно при определенных условиях заменить значениями, полученными более быстрыми процедурами, но с большей погрешностью. Предполагая, что зависимость времени расчетов от допустимой погрешности заданных показателей известна, можно поставить задачу оптимального выбора расчетной процедуры и требований к ее точности.

5. Минимаксная задача в многокритериальной форме и построение алгоритма ее решения

Формулировка задачи проектирования в минимаксной форме (5) не всегда позволяет получить однозначное и наилучшее решение. Рассмотрим такой случай на следующем примере. Пусть параметры проектируемой схемы представлены номиналами резисторов и емкостей, а ее показатели разбиты на две группы: статические, например напряжение и мощность в рабочей точке, и динамические, например задержки на разных выходах и/или ширина полосы частот. Очевидно, что статические показатели не зависят от значений емкостей, а динамические показатели зависят от всех параметров. Тогда, если минимаксное значение для задачи в целом достигается на статических показателях, то оно не зависит от емкостей, значения которых однозначно не определены. Для их определения нужно решить еще одну минимаксную задачу в отношении динамических показателей как функций от значений емкостей при найденных номиналах резисторов. Проблема заключается в том, чтобы подобное последовательное решение минимаксных задач обеспечивалось автоматически в рамках алгоритма.

Указанное расширение минимаксной задачи относится в работе [21] к многокритериальным задачам с однородными равноценными критериями, для которых даются подробные формально-математические характеристики. Однако рассмотрение практических алгоритмов в [21] ограничивается дискретным случаем. Поэтому ниже приведено общее описание такого алгоритма для решения линеаризованной задачи вида (11).

Рассмотрим линейную минимаксную задачу в следующей форме.

Заданы следующие линейные зависимости показателей от параметров:

$$y_i(x) = \sum_{j=1}^M a_{ij}x_j + b_i, \quad i = 1, 2, \dots, N, \quad M < N. \quad (12)$$

Требуется найти такие значения вектора параметров x ($\dots x_j \dots$), что максимальное значение показателя принимает наименьшее возможное значение:

$$\max_i(y_i(x)) \rightarrow \min_x. \quad (13)$$

Прежде всего отметим, что задача (13) в одномерном случае ($M = 1$) для скалярного аргумента x записывается как

$$\max_{1 \leq i \leq N} (a_i x + b_i) \rightarrow \min_x \quad (14)$$

и имеет простое решение, что наглядно иллюстрирует рис. 4. Соответствующий алгоритм приводить не будем — он достаточно очевиден.

Система (12) с помощью гауссовского исключения переменных x_j может быть приведена к виду:

$$Hx + uy + v = 0; \quad (15)$$

$$\sum_{j=1}^N c_{ij}y_j + d_i = 0, \quad (16)$$

$$i = 1, 2, \dots, K, \quad K = N - M < N,$$

где H — верхняя треугольная матрица, $K = N - M < N$.

Таким образом, задача (13) сведена к задаче

$$\max_{1 \leq i \leq N} (y_i) \rightarrow \min_y \quad (17)$$

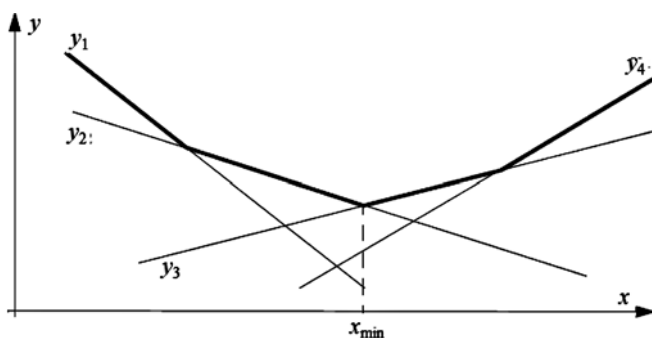


Рис. 4. Решение одномерной линейной минимаксной задачи. Полу жирные линии показывают функцию $\max(y_1(x), y_2(x), y_3(x), y_4(x))$

при линейной связи (16) переменных y . Система (15) используется не при оптимизации (16), (17), а после ее завершения для определения параметров x .

Прежде чем переходить к решению общей задачи (16), (17), рассмотрим ее частный случай для одного уравнения ($K = 1$):

$$\sum_{j=1}^N c_j y_j + d = 0. \quad (18)$$

Если все коэффициенты c_j имеют одинаковый знак, решением задачи (17), (18) будут равные значения показателей: $y_j = -d / \sum_{j=1}^N c_j$, так как уменьшение какого-либо y_j приводит к необходимости увеличить максимальное значение, чтобы сохранить сумму в уравнении (18). Если же в (18) имеются хотя бы два коэффициента c_j с разными знаками, то задача не имеет решения, так как можно улучшать оба показателя, не меняя сумму (18).

Решение общей задачи (16), (17) будем искать следующим образом. Множество переменных y разобьем на K групп, обозначив s_k — множество индексов k -й группы и полагая, что все переменные такой группы имеют одинаковое значение $y_j = z_k, j \in s_k$. Тогда система уравнений с квадратной матрицей относительно общих значений z_k примет следующий вид:

$$\sum_{j=1}^K q_{ik} z_k + d_i = 0, \quad \text{где } q_{ik} = \sum_{j \in s_k} c_{ij}. \quad (19)$$

Систему (19) линейным преобразованием всегда можно привести к единичному диагональному виду: $q_{ii} = 1, q_{ik} = 0$ при $i \neq k$. Такое преобразование соответствующим образом изменит и коэффициенты d_i и c_{ij} . Предполагая сортировку уравнений в (19) по возрастанию d_i и учитывая для c_{ij} условие диагональности матрицы q , получим следующие соотношения:

$$A: d_i > d_{i+1}; \quad B: \sum_{j \in s_i} c_{ij} = 1; \quad C: \sum_{j \in s_k, k \neq i} c_{ij} = 0. \quad (20)$$

Отметим, что в силу диагональности системы (19) ее решение (общие значения переменных из группы) $z_i = -d_{i+1}$ удовлетворяет условию $z_i > z_{i+1}$. К виду, удовлетворяющему условиям (19), (20), можно привести любую систему, задав некоторым образом разбиение

переменных на группы. Для такой системы задача (16), (17) будет решена, если выполнены следующие условия:

$$c_{ij} \geq 0, j \in s_i; \quad (21)$$

$$c_{ij} = 0, j \in s_k, k > i. \quad (22)$$

Условие (21) следует из свойства задачи с одним уравнением (19) — общий знак для всех коэффициентов. Равенство (22) следует из того, что ненулевой коэффициент влияния показателя k -й группы на i -й критерий не может существовать в точке оптимума, так как при этом становится возможным улучшить (снизить) i -й критерий за счет k -го (менее критичного при $k > i$).

Невыполнение условий (21), (22) позволяет определить направление изменения одного из показателей для улучшения одного из критериев. Продемонстрируем принцип построения соответствующего алгоритма для наиболее важного случая невыполнения условия (10) для первого уравнения.

Пусть m — индекс переменной, для которой (10) при $i = 1$ не выполняется, а n — индекс соответствующей группы $c_{1m} < 0, m \in s_n$. Удалим из n -й группы переменную y_m , обозначив s'_n образовавшуюся группу без этой переменной. Для сохранения в "сокращенной" группе s'_n равенства (20) для В разделим n -ю строку (без m -го элемента) на $c'_n = \sum_{j \in s'_n} c_{nj}$: $c'_{nk} = c_{nk}/c'_n$, $d'_n = d_n/c'_n, k = 1, 2, \dots, K$.

Вычтем из каждой другой строки системы n -ю строку, умноженную на величину, которая позволит сохранить равенство (20) для С для остальных групп:

$$c'_{ik} = c_{ik} - c'_{nk}c'_i, d'_i = d_i - d'c'_i,$$

$$i, k = 1, 2, \dots, K, (k \neq n).$$

Тогда систему относительно общих значений, аналогичную (19), можно записать в виде

$$z_i = -d'_i - c_{im}y_m. \quad (23)$$

Полагая, что переменная y_m образует дополнительную $(K + 1)$ -ю группу, запишем для нее $z_{K+1} = y_m$, что можно рассматривать как дополнительное уравнение в (23) с $d'_{K+1} = 0$,

$c_{K+1, m} = -1$. Тогда равенства (23) задают условия для одномерной минимаксной задачи (аналогичной (14)):

$$\max_{1 \leq i \leq K+1} (z_i) = \max_{1 \leq i \leq K+1} (-d'_i - c_{im}y_m) \rightarrow \min_{y_m}. \quad (24)$$

Решив эту задачу (см. рис. 4), получим значение y_m и улучшенное значение критерия. После этого нужно снова сократить число групп и привести систему к исходной форме, удовлетворяющей условиям (20). Для этого учтем, что минимум на рис. 4 соответствует совпадению общих значений переменных из двух групп (обозначим их p и r), и объединим их в одну новую группу, которой присвоим индекс 1, так как ее общее значение максимально среди всех групп. Значение y_m , соответствующее точке пересечения, определяется из равенства $z'_p = z'_r = -d'_p - c_{pm}y_m = -d'_r - c_{rm}y_m$, откуда получаем

$$y_m = (d'_r - d'_p)/(c_{rm} - c_{pm}). \quad (25)$$

Подставив (25) в систему (23), найдем значения всех компонентов вектора критериев z_m . Затем следует провести сортировку уравнений по значениям z_m для выполнения (20 А) и линейное преобразование новой системы (19), чтобы обеспечить выполнение условий (20 В и С).

В том случае, если после получения минимаксного значения имеются группы, содержащие более одного показателя, можно продолжить процесс оптимизации, отбросив показатели из первой группы. Этим автоматически осуществляется последовательное решение минимаксных задач, о которых говорилось в начале раздела. Таким образом, предложенный метод позволяет осуществлять векторную (многокритериальную) оптимизацию.

Заключение

Выполнен анализ проблемы выбора обобщенного критерия качества для схемотехнического проектирования интегральных схем. Показано, что минимаксный критерий предпочтительнее других применяемых критериев.

Рассмотрены вопросы задания частных критериев, входящих в общий минимаксный критерий. Отмечены достоинства методики опре-

деления весовых коэффициентов на основе опыта разработчика. Предложено обобщение этой методики на более широкий класс выпуклых кусочно-линейных функций.

Проведен анализ известных подходов к построению алгоритмов минимаксной оптимизации. Показано, что в качестве базового метода следует выбрать метод последовательного решения линеаризованных минимаксных задач. Предложены основные направления совершенствования алгоритмов минимаксной оптимизации для задач схемотехнического проектирования, учитывающие особенности алгоритмов электрического моделирования и основанные на адаптивности выбора шага, асинхронности вычисления показателей и управлении точностью моделирования.

Показана необходимость учета многокритериальности при схемотехническом проектировании на базе минимаксной оптимизации, и разработаны принципы построения алгоритма решения линейной минимаксной задачи с учетом эффекта многокритериальности.

Предложенные подходы к построению оптимизационных минимаксных алгоритмов предполагается использовать при разработке программного обеспечения для подсистемы проектирования аналоговых интегральных схем.

Список литературы

1. Баталов Б. В., Егоров Ю. Б., Русаков С. Г. Основы математического моделирования больших интегральных схем на ЭВМ. М.: Радио и связь, 1983. 168 с.
2. Lavagno L., Markov I. L., Martin G. E., Scheffer L. K. Electronic Design Automation for Integrated Circuits Handbook. Second Edition. Two Volume Set CRC Press Reference. May 5, 2016. 1430 p.
3. Бененсон З. М., Елистратов М. Р., Ильин Л. К. Моделирование и оптимизация на ЭВМ радиоэлектронных устройств. М.: Радио и связь, 1981. 272 с.
4. Батищев Д. И. Методы оптимального проектирования: учеб. пособие. М.: Радио и связь, 1984. 248 с.
5. Degrauwe M. G. R., Goffart B. L. A. G., Meixenberger C. et al. Towards an Analog System Design Environment // IEEE J. of Solid-State Circuits. 1989. Vol. 24, N. 3. P. 659–670.
6. Hershenson M., Boid S. P., Lee T. H. GPCAD: A Tool for CMOS Op-Amp Synthesis // ICCAD '98 Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design. P. 296–303.

7. McConaghy T., Palmers P., Steyaert M., Gielen G. Trustworthy Genetic Programming-Based Synthesis of Analog Circuit Topologies Using Hierarchical Domain-Specific Building Blocks // IEEE Trans. on Evolutionary Computation. 2011. Vol. 15, N. 4. P. 557–570.

8. Koh H. Y., Seqin C. H., Gray P. R. OPASYN: A Compiler for CMOS Operational Amplifiers // IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems. Feb. 1990. Vol. 9, N. 2. P. 113–125.

9. Ochotta E. S., Rulenbar R. A., Carley L. R. Synthesis of High-Performance Analog Circuits in ASTRX/OBLX // IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems. March 1996. Vol. 15, N. 3. P. 273–292.

10. Torralba A., Chavez J., Franquelo L. G. FASY: A Fuzzy-Logic Based Tool for Analog Synthesis // IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems. July 1996. Vol. 15, N. 7. P. 705–715.

11. Fares M., Kaminska B. FPAD: A Fuzzy Nonlinear Programming Approach to analog Circuit Design // IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems. July 1995. Vol. 14, N. 7. P. 785–793.

12. Gielen G., Walscharts H., Sansen W. Analog Circuit Design Optimization Based on Symbolic Simulation and Simultaneous Annealing // IEEE J. of Solid-State Circuits. 1990. Vol. 25, N. 3. P. 707–713.

13. Krasnicki M., Phelps R., Rutenbar R., Carley L. MAELSTROM: Efficient Simulation-Based Synthesis for Custom Analog Circuits // Proc. DAC99, New Orleans, Louisiana. 1999. P. 945–950.

14. Vancorenland P., De Ranter C., Steyaert M., Gielen G. Optimal RF Design using Smart Evolutionary Algorithms // Proc. 37th Design Automation Conference. Los Angeles, California, 2000. P. 7–10.

15. Leyn F., Daems W., Gielen G., Sansen W. A Behavioral Signal Path Modeling Methodology for Qualitative Insight in and Efficient Sizing of CMOS Opamps // Proc. ACM/IEEE ICCAD97. San Jose, California, 1997. P. 374–381.

16. Medeiro F. Perez-Verdu B., Rodriguez-Vazquez A., Huer-tas J. L. A Vertically Integrated Tool for Automated Design of $\Sigma\Delta$ Modulators // IEEE J. of Solid-State Circuits. 1995. Vol. 30, N. 7. P. 762–772.

17. Nye W., Riley D. C., Sangiovanni-Vincentelli A., Tits A. L. DELIGHT.SPICE: An Optimization-Based System for the Design of Integrated Circuits // IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems. 1988. Vol. 7, N. 4. P. 501–518.

18. Пшеничный Б. Н., Данилин Ю. М. Численные методы в экстремальных задачах. М.: Наука, 1975. 320 с.

19. Карманов В. Г. Математическое программирование: учеб. пособие. 5 изд. М.: ФИЗМАТЛИТ, 2004. 264 с.

20. Демьянов В. Ф., Малоземов В. Н. Введение в минимакс. М.: Наука, 1972. 368 с.

21. Подиновский В. В., Гаврилов В. М. Оптимизация по последовательно применяемым критериям. М.: ЛЕНАНД, 2016. 194 с.

M. M. Gourary, Ph. D., Senior Research Associate, gourary@ippm.ru,
M. M. Zharov, Ph. D., Leading Research Associate, zarov@ippm.ru,
S. G. Rusakov, D. Sc., Principal Research Associate, rusakov@ippm.ru,
S. L. Ulyanov, D. Sc., Principal Research Associate, ulyas@ippm.ru,
Institute for Design Problems in Microelectronics RAS, Moscow, 124365, Russian Federation

Minimax Optimization in Circuit Design Problems

The directions of improvement of minimax methods for circuit design problems are considered. The choices of generalized quality criterion for the circuit design is discussed. It is concluded that the minimax criterion has advantages over other formulations of design targets. New approach to setting of individual objectives for each performance indicator is proposed. The approach extends well-known weighting techniques based on designer's experience to the representation of arbitrary piecewise linear dependences. From the analysis of known minimax optimization algorithms, it was considered that the method of sequential linearized minimax solvings should be the basis of the computational algorithm. Main directions of the development of minimax optimization methods are proposed taking into account specific features of simulation algorithms included in the optimization procedure. The directions are based on the stepsize adaptability, asynchronous performances evaluations and the simulation accuracy control. The possible need for the multi-objective formulation of minimax problem is illustrated by the circuit example. An outline of the linear minimax optimization algorithm taking into account the possible multicriteria is presented.

Keywords: electronic design automation, minimax optimization, circuit simulation, linear programming, analog circuit, iterative methods, multi-objective optimization

DOI: 10.17587/it.24.435-444

References

1. **Batalov B. V., Egorov Ju. B., Rusakov S. G.** *Osnovy matematicheskogo modelirovaniya bol'shikh integral'nykh shem na JeVM* (Basics of Mathematical Modeling of Large Integrated Circuits), Moscow, Radio i svjaz', 1983, 168 p. (in Russian).
2. **Lavagno L., Markov I. L., Martin G. E., Scheffer L. K.** *Electronic Design Automation for Integrated Circuits Handbook*, Second Edition, Two Volume Set May 5, 2016 by CRC Press Reference, 1430 p. (in Russian).
3. **Benenson Z. M., Elistratov M. R., Il'in L. K.** *Modelirovanie i optimizacija na JeVM radioelektronnykh ustrojstv* (RF Devices Computer Simulation and Optimization), Moscow, Radio i svjaz', 1981, 272 p. (in Russian).
4. **Batishhev D. I.** *Metody optimal'nogo proektirovaniya: uchebnoe posobie* (Optimal Design Methods. Textbook), Moscow, Radio i svjaz', 1984, 248 p. (in Russian).
5. **Degrauwe M. G. R., Goffart B. L. A. G., Meixenberger C.** Towards an Analog System Design Environment, *IEEE J. of Solid-State Circuits*, 1989, vol. 24, no. 3, pp. 659–670.
6. **Hershenson M., Boid S. P., Lee T. H.** GPCAD: A Tool for CMOS Op-Amp Synthesis, *ICCAD '98, Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design*, pp. 296–303.
7. **McConaghy T., Palmers P., Steyaert M., Gielen G.** Trustworthy Genetic Programming-Based Synthesis of Analog Circuit Topologies Using Hierarchical Domain-Specific Building Blocks, *IEEE Trans. on Evolutionary Computation*, 2011, vol. 15, no. 4, pp. 557–570.
8. **Koh H. Y., Sequin C. H., Gray P. R.** OPASYN: A Compiler for CMOS Operational Amplifiers, *IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems*, Feb. 1990, vol. 9, no. 2, pp. 113–125.
9. **Ochotta E. S., Rutenbar R. A., Carley L. R.** Synthesis of High-Performance Analog Circuits in ASTRX/OBLX, *IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems*, March 1996, vol. 15, no. 3, pp. 273–292.
10. **Torralba A., Chavez J., Franquelo L. G.** FASY: A Fuzzy-Logic Based Tool for Analog Synthesis, *IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems*, July 1996, vol. 15, no. 7, pp. 705–715.
11. **Fares M., Kaminska B.** FPAD: A Fuzzy Nonlinear Programming Approach to analog Circuit Design, *IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems*, July 1995, vol. 14, no. 7, pp. 785–793.
12. **Gielen G., Walscharts H., Sansen W.** Analog Circuit Design Optimization Based on Symbolic Simulation and Simulated Annealing, *IEEE J. of Solid-State Circuits*, 1990, vol. 25, no. 3, pp. 707–713.
13. **Krasnicki M., Phelps R., Rutenbar R., Carley L.** MAELSTROM: Efficient Simulation-Based Synthesis for Custom Analog Circuits, *Proc. DAC99, New Orleans, Louisiana*, 1999.
14. **Vancorenland P., De Ranter C., Steyaert M., Gielen G.** Optimal RF Design using Smart Evolutionary Algorithms, *Proc. Design Automation Conference*, 2000.
15. **Leyn F., Daems W., Gielen G., Sansen W.** A Behavioral Signal Path Modeling Methodology for Qualitative Insight in and Efficient Sizing of CMOS Opamps, *Proc. ACM/IEEE ICCAD97*, 1997.
16. **Medeiro F. Perez-Verdu B., Rodriguez-Vazquez A., Huertas J. L.** A Vertically Integrated Tool for Automated Design of $\Sigma\Delta$ Modulators, *IEEE J. of Solid-State Circuits*, 1995, vol. 30, no. 7, pp. 762–772.
17. **Nye W., Riley D. C., Sangiovanni-Vincentelli A., Tits A. L.** DELIGHT.SPICE: An Optimization-Based System for the Design of Integrated Circuits, *IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems*, 1988, vol. 7, no. 4, pp. 501–518.
18. **Pshenichnyj B. N., Danilin Ju. M.** *Chislennyye metody v jekstremal'nykh zadachah* (Numerical Methods in extreme problems), Moscow, Nauka, 1975, 320 p. (in Russian).
19. **Karmanov V. G.** *Matematicheskoe programmirovaniye*, Uchebnoe posobie. — 5 izd. (Mathematical Programming), Moscow, FIZMATLIT, 2004, 264 p. (in Russian).
20. **Dem'janov V. F., Malozemov V. N.** *Vvedenie v minimaks* (Introduction in Minimax), Moscow, Nauka, 1972, 368 p. (in Russian).
21. **Podinovskij V. V., Gavrilov V. M.** *Optimizacija po posledovatel'no primenjaemym kriterijam* (Optimization by Sequentially Applied Criteria), Moscow, LENAND, 2016, 194 p. (in Russian).

П. С. Романов, д-р техн. наук, проф., e-mail: romanov_p_s@mail.ru,
Коломенский институт (филиал) ФГБОУ ВО "Московский политехнический университет",
г. Коломна,

И. П. Романова, канд. техн. наук, e-mail: irom84@mail.ru
ФГБОУ ВО "Национальный исследовательский московский государственный
строительный университет" (НИУ МГСУ), г. Москва, 129337

Методика принятия решений по выбору наилучшей технической системы по набору ее характеристик

Рассматривается комплексный подход к принятию решения по выбору наилучшей технической системы по количественным и качественным характеристикам. Оценку вариантов и выбор наилучшего варианта технической системы предлагается проводить на основе методики, построенной с использованием методов спектрального анализа и анализа иерархий. Приводится описание методов, две постановки задачи и алгоритмы решения этими методами. Анализируются недостатки и преимущества указанных методов.

Ключевые слова: *принятие решения, техническая система, метод спектрального анализа, метод анализа иерархий, комплексная методика выбора технической системы*

Введение

Для выбора лучшей технической системы (ТС) из набора имеющихся ТС должностное лицо, отвечающее за этот выбор (или лицо, принимающее решение (ЛПР)), должно решить следующую задачу.

Выбрать один из вариантов технической системы (одну из представленного модельного ряда технических систем) по ее показателям (количественным и/или качественным характеристикам (параметрам)) на основе критериев выбора, установленных ЛПР.

К таким задачам, в частности, относятся:

- выбор рационального комплекта технологического оборудования для передвижных автомобильных ремонтных мастерских [1, 2];
- выбор варианта станции технического обслуживания автомобилей [3];
- определение лучшего варианта системы освещения производственных помещений ремонтных мастерских (РМ) по техническому обслуживанию и текущему ремонту сельскохозяйственной техники [4–7].

В работах [1–7] предложены методики решения указанных задач для случаев, когда ТС характеризуются количественными или качественными параметрами.

Если известны только числовые оценки характеристик технической системы, то более эффективно проводить выбор рационального

варианта ТС с помощью метода спектрального анализа (МСА) [1, 4, 5]. Часто характеристики сложных технических систем являются не только количественными, но и качественными. В этом случае указанную задачу выбора одного из вариантов ТС в качестве наилучшего предлагается в работах [2, 6–8] решать на основе метода анализа иерархий (МАИ). Применение МАИ дает возможность выбрать рациональный вариант ТС из представленного множества альтернатив по совокупности количественных и качественных параметров. Причем задача принятия решения в этом случае является многокритериальной. При этом варианты ТС по количественным характеристикам эксперты сравнивают на качественном уровне попарно, а потом полученные качественные оценки переводят в числовые относительные показатели на основе шкалы относительной важности, предложенной Т. Саати в [9]. В работах [2, 6–8] все результаты получены с использованием согласованных экспертных оценок.

Постановка задачи

В комплексной методике принятия решений по выбору наилучшей ТС предлагается использовать методы спектрального анализа и анализа иерархий, поэтому приведем постановки задач и алгоритмы решений для этих методов. Рас-

смотрение для наглядности проведем на примере выбора автомобиля по его основным эксплуатационным свойствам из группы автомобилей одного класса.

Рассмотрим метод спектрального анализа на основе положений работы [1].

Пусть есть n вариантов ТС (моделей автомобилей), представленных для сравнения. Обозначим их $ТС_i$, $i = \overline{1, n}$. Известны количественные технические характеристики этих систем (моделей автомобилей) X_j , $j = \overline{1, m}$, с их помощью можно выявить предпочтительность того или иного варианта ТС (автомобиля). Исходя из учета одной характеристики X_j предпочтительность варианта технической системы (автомобиля) $ТС_i$ определяют показателем x_{ij} . Первоначально полагают, что все характеристики равнозначны, т.е. $W_i^{(0)} = 1$, $V_j^{(0)} = 1$, где $W_i^{(0)}$, $V_j^{(0)}$ — значения весовых коэффициентов нулевой итерации соответственно для вариантов ТС (автомобилей) и для их характеристик. Требуется по всей совокупности показателей x_{ij} , $i = \overline{1, n}$, $j = \overline{1, m}$, построить ранжированный по предпочтительности ряд вариантов ТС (автомобилей), на основе которого выбрать наилучший из них.

В основе решения данной задачи методом спектрального анализа лежит следующая итерационная процедура:

$$W_i^{(k)} = \sum_{j=1}^m V_j^{(k-1)} \tilde{x}_{ij} / \max_i \sum_{j=1}^m V_j^{(k-1)} \tilde{x}_{ij}; \quad (1)$$

$$V_j^{(k)} = \sum_{i=1}^m W_i^{(k-1)} \tilde{x}_{ij} / \max_j \sum_{i=1}^m W_i^{(k-1)} \tilde{x}_{ij}, \quad (2)$$

где n — число вариантов ТС (автомобилей); m — число технических характеристик; $k = \overline{1, K}$ — номер итерации в итерационном процессе (если $k \rightarrow \infty$, то $W_i, V_j \rightarrow \text{const}$);

$$\tilde{x}_{ij} = \begin{cases} \frac{x_{ij}}{\max\{x_{ij}\}}, & \text{если увеличение } x_{ij} \\ & \text{приводит к улучшению свойств ТС;} \\ \frac{\min\{x_{ij}\}}{x_{ij}}, & \text{если уменьшение } x_{ij} \\ & \text{приводит к улучшению свойств ТС;} \end{cases} \quad (3)$$

$W_i^{(k)}$, $V_j^{(k)}$ — значения весовых коэффициентов k -й итерации для вариантов ТС (автомобилей) и для их характеристик соответственно.

Для решения задачи в [1, 4, 5] предложен следующий алгоритм.

Шаг 1. По формуле (3) рассчитать нормированную матрицу $\{\tilde{x}_{ij}\}$, $i = \overline{1, n}$, $j = \overline{1, m}$.

Шаг 2. По форме табл. 1 построить таблицу для расчетов, куда внести полученные значения $\{\tilde{x}_{ij}\}$, $i = \overline{1, n}$, $j = \overline{1, m}$.

Шаг 3. Для первой итерации ($k = 1$) по формуле (1) провести расчет значений $W_i^{(1)}$, при $V_j^{(0)} = 1$, \tilde{x}_{ij} , $i = \overline{1, n}$, $j = \overline{1, m}$ берут из табл. 1. Полученные значения $W_i^{(1)}$ записать в табл. 1.

Шаг 4. Провести вторую итерацию ($k = 2$). По формуле (2) рассчитать $V_j^{(2)}$, используя данные, полученные в п. 1 и п. 3 алгоритма. Записать их в табл. 1.

5. Далее рассчитать значения $W_i^{(k)}$ для нечетных итераций и $V_j^{(k)}$ — для четных согласно шагам 3, 4 алгоритма. Вычисления необходимо проводить до тех пор, пока значения $W_i^{(k)}$ и $V_j^{(k)}$ не примут значения весовых коэффициентов, полученных на предыдущей итерации. Так пошагово заполняется табл. 1.

Если расчеты проводить на ПЭВМ, то k задают заранее. Значения весовых коэффициентов обычно начинают повторяться при $k \geq 7$ [1, 4], исходя из этого и надо выбирать значение k .

6. Далее провести расчет $W_i^{(k)}$ и $V_j^{(k)}$ по формулам (1), (2), начав с нахождения $V_j^{(1)}$ при условии, что $W_i^{(0)} = 1$, и далее по указанному ранее алгоритму заполнить оставшиеся строки и столбцы в табл. 1.

Таблица 1

Форма записи промежуточных расчетов

W_i	V_j	V_1	...	V_j	...	V_m	$W_i^{(0)}$	$W_i^{(1)}$	$W_i^{(2)}$	$W_i^{(3)}$
W_1	\tilde{x}_{11}	...	\tilde{x}_{1j}	...	\tilde{x}_{1m}	1	$W_1^{(0)}$	$W_1^{(1)}$		$W_1^{(3)}$
...
W_i	\tilde{x}_{i1}	...	\tilde{x}_{ij}	...	\tilde{x}_{im}	1	$W_i^{(0)}$	$W_i^{(1)}$		$W_i^{(3)}$
...
W_n	\tilde{x}_{n1}	...	\tilde{x}_{nj}	...	\tilde{x}_{nm}	1	$W_n^{(0)}$	$W_n^{(1)}$		$W_n^{(3)}$
$V_j^{(0)}$	1	...	1	...	1					
$V_j^{(1)}$							
$V_j^{(2)}$	$V_1^{(2)}$...	$V_j^{(2)}$...	$V_m^{(2)}$					
$V_j^{(3)}$							
$V_j^{(4)}$	$V_1^{(4)}$...	$V_j^{(4)}$...	$V_m^{(4)}$					

7. Получив на какой-то итерации (или заданной итерации) повторяющиеся значения $W_i^{(k)}$, построить на их базе ранжированный ряд вариантов ТС, $i = \overline{1, n}$, технической системы (автомобилей).

8. Выбрать оптимальный вариант ТС (автомобиля) — ТС_o по формуле

$$W_o = \max_i W_i^{(k)},$$

где W_o — весовой коэффициент k -й итерации варианта ТС, имеющий наибольшее значение.

Для решения задачи выбора наилучшей ТС (модели автомобиля) из группы автомобилей, относящихся к одному классу, по основным качественным характеристикам с помощью МАИ, описанному в работе [2], приведем следующую постановку задачи.

Постановка задачи. Покупатель (автопредприятие) изучает возможность покупки автомобиля из модельного ряда автомобилей, относящихся к одному классу. Эксперты [10] определили пять качественных характеристик (критериев), которым должен удовлетворять автомобиль. Задача заключалась в выборе одного из четырех вариантов автомобилей: Ford Focus, Mazda3, Nissan Sentra, Skoda Octavia.

При рассмотрении выбранных моделей автомобилей эксперты [10] предложили следующие качественные характеристики.

Рабочее место водителя. *Сиденье, органы управления, обзор.* Неплохо устроиться можно в каждом автомобиле, однако Nissan слегка отстает от конкурентов, водительское кресло со скромным диапазоном регулировок по длине установлено слишком высоко, руль слишком дешевый, зато у него обзорность отличная, тонкие передние стойки сочетаются с камерой заднего вида. Ford при наличии камеры имеет толстые стойки. Skoda и Mazda имеют хорошую обзорность.

Салон. *Передняя и задняя части, багажник.* К эргономике серьезных претензий нет. Skoda имеет продуманный и просторный салон, особенно на уровне плеч, удобство посадки водителя. Mazda и Ford имеют тесный во всех измерениях задний ряд. Nissan гораздо просторнее. Но в Nissan неудачно (на уровне головы) расположены потолочные ручки, в Skoda отсутствует центральный подголовник. Кроме того, Skoda имеет огромный функциональный багажник.

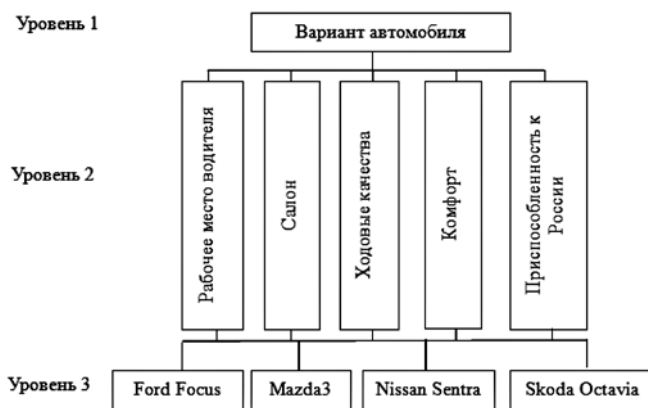
Ходовые качества. *Динамика, тормоза, поведение на дороге.* Ни один из автомобилей не является резвым, особенно Skoda. Mazda страдает плохой информативностью педали тормоза, но имеет хорошую управляемость. Ford обладает эффективными тормозами по сравнению с другими моделями, но у него неоднозначны настройки электроусилителя руля. Nissan — худший по управляемости, но имеет хорошие тормоза.

Комфорт. *Шум, плавность хода, климат.* Mazda — самая шумная в квартете. У Skoda двигатель шумит при разгоне, но при равномерном движении — самый тихий. Nissan — лидер по плавности хода. Mazda является аутсайдером по плавности хода. По показателю климат — лучший автомобиль Ford, так как обладает обогревом руля и лобового стекла. У всех автомобилей, кроме Skoda, — двухзонный климат-контроль, а у нее обычный кондиционер, но она имеет дефлекторы обдува заднего дивана.

Приспособленность к России. *Геометрическая проходимость, сервис, эксплуатация.* У всех машин приличный клиренс. Skoda имеет меньший срок гарантии, чем другие машины, но у нее нет ограничений по пробегу. Ford имеет заливную горловину без пробки. Остальные параметры у машин одинаковые.

Кратко опишем процедуру решения задачи на базе МАИ на примере технических систем — моделей автомобилей для задачи, приведенной выше; подробно методический аппарат описан в [2, 6–8].

На первом этапе проводим декомпозицию и представляем задачу в иерархической форме (графическое представление иерархии приведено на рисунке).



Декомпозиция задачи в иерархию

На данном этапе определяют не только цель — верхний уровень иерархии — вариант автомобиля, но и критерии, по которым проводится выбор. В данном случае выбраны только качественные характеристики автомобилей (критерии): *Рабочее место водителя, Салон, Ходовые качества, Комфорт, Приспособленность к России* (они подробно описаны ранее). На уровне 3 находятся модели (варианты) автомобилей, которые рассматриваются в задаче.

На втором этапе для получения матриц суждений уровней 2 и 3 применяем принцип дискриминации и сравнительных суждений. Последовательно проводим парные сравнения с использованием табл. 2.

В результате получим одну квадратную матрицу для уровня 2 иерархии и пять матриц суждений размерностью 4×4 для уровня 3, так как имеется пять критериев на уровне 2 (табл. 3) и четыре альтернативных варианта автомобилей. В табл. 4 приведен пример одной из матриц для критерия "Ходовые качества".

На третьем этапе вычисляем множество собственных векторов для каждой матрицы уровней 2 и 3 иерархии, а затем нормируем результат к единице, получая тем самым векторы приоритетов для каждой матрицы уровней 2 и 3 иерархии (см. табл. 3, 4). Используем для этого геометрическое среднее.

Таблица 2

Шкала относительной важности [9]

Интенсивность относительной важности	Качественная оценка интенсивности относительной важности	Пояснения
1	Равная важность	Равный вклад двух объектов (деятельности) в цель
3	Умеренное превосходство одного над другим	Опыт и суждения дают легкое превосходство одного объекта (вида деятельности) над другим
5	Существенное или сильное превосходство	Опыт и суждения дают сильное превосходство одного объекта (вида деятельности) над другим
7	Значительное превосходство	Один объект (вида деятельности) имеет настолько сильное превосходство, что оно становится практически значительным
9	Очень сильное превосходство	Очевидность превосходства одного объекта (вида деятельности) над другим подтверждается наиболее сильно
2, 4, 6, 8	Промежуточные решения между двумя соседними суждениями	Применяются в компромиссном случае
Обратные величины приведенных выше чисел	Если при сравнении одного объекта (вида деятельности) с другим получено одно из приведенных выше чисел (например, 3), то при сравнении второго объекта (вида деятельности) с первым получим обратную величину (т.е. 1/3)	

Таблица 3

Матрица попарных сравнений для уровня 2 (для уровня критериев)

Общее удовлетворение вариантом	Ходовые качества	Приспособленность к России	Рабочее место водителя	Салон	Комфорт	Вектор приоритетов	Y_i
Ходовые качества	1	2	3	5	7	0,423	2,173
Приспособленность к России	1/2	1	3	5	7	0,320	1,641
Рабочее место водителя	1/3	1/3	1	2	3	0,134	0,674
Салон	1/5	1/5	1/2	1	2	0,076	0,385
Комфорт	1/7	1/7	1/3	1/2	1	0,047	0,235
	$\lambda_{\max} = 5,107$		ИС = 0,027		ОС = 0,024		

Матрица попарных сравнений для критерия "Ходовые качества" для уровня 3 (уровня вариантов)

Ходовые качества	Ford Focus	Mazda3	Nissan Sentra	Skoda Octavia	Вектор приоритетов	Y_i
Ford Focus	1	1	1	2	0,291	1,245
Mazda3	1	1	1/2	1/2	0,173	0,732
Nissan Sentra	1	2	1	1	0,291	1,173
Skoda Octavia	1/2	2	1	1	0,245	1,028
	$\lambda_{\max} = 4,177$		ИС = 0,059		ОС = 0,066	

Таблица 5

Таблица средних значений СИ [9]

Порядок матрицы	1	2	3	4	5	6	7	8	9	10
СИ	0,00	0,00	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49

На четвертом этапе вычисляем собственное значение матрицы суждений, индекса согласованности и отношения согласованности для каждой матрицы уровней 2 и 3 иерархии. Собственное значение каждой из матриц суждений λ_{\max} определим по формуле (результаты приведены в табл. 3, 4):

$$\lambda_{\max} = \sum_{i=1}^n Y_i,$$

где значения Y_i получено перемножением каждой из матриц суждений на свой вектор приоритетов (приведены в табл. 3, 4).

Вычисление индекса согласованности ИС проводим по формуле

$$\text{ИС} = (\lambda_{\max} - n)/(n - 1),$$

где n — число сравниваемых элементов (в данной задаче для уровня 2 $n = 5$, для каждой из матриц уровней 3 $n = 4$).

Надо заметить, что для обратно симметричной матрицы всегда $\lambda_{\max} \geq n$. Результаты для уровня 2 приведены в табл. 3, в табл. 4 представлены результаты вычислений для одного из критериев уровня 3.

Рассчитаем отношение согласованности ОС по формуле: $\text{ОС} = \text{ИС}/\text{СИ}$, где значение случайного индекса СИ берется из табл. 5 [9]. Результаты вычислений приведены в табл. 3, 4.

На пятом этапе применяем иерархический синтез для взвешивания собственных векторов

весами критериев и вычисляем сумму по всем соответствующим взвешенным компонентам собственных векторов уровня иерархии, лежащего ниже. Для получения обобщенных или глобальных приоритетов вариантов ТС (автомобилей) в матрице локальные приоритеты располагаются по отношению к каждому критерию, каждый столбец векторов умножается на приоритет соответствующего критерия и результат складывается вдоль каждой строки. Результаты представлены в табл. 6.

На шестом этапе находим согласованность всей иерархии, перемножая каждый индекс согласованности на приоритет соответствующего критерия и суммируя полученные числа. Отношения согласованности для всех матриц уровней 2 и 3 не превышают 10 % (согласно [9] ОС должно быть около или менее 10 %), поэтому можно утверждать, что мнения экспертов при оценке выбранных для оценки автомоби-

Таблица 6

Глобальные приоритеты

Модели автомобилей	Вектор приоритетов критериев оценки автомобилей					Обобщенные (глобальные) приоритеты
	0,423	0,320	0,134	0,076	0,047	
Ford Focus	0,291	0,286	0,286	0,096	0,347	0,276
Mazda3	0,173	0,143	0,286	0,096	0,074	0,168
Nissan Sentra	0,291	0,286	0,143	0,249	0,406	0,272
Skoda Octavia	0,245	0,286	0,286	0,558	0,173	0,284

лей критериев, а также при сравнении вариантов моделей автомобилей по этим критериям согласованы. Согласованность всей иерархии составила 3 %.

Результаты решения

Приведем результаты решения примера по выбору автомобиля из группы автомобилей, относящихся к одному классу, по их основным эксплуатационным свойствам:

- *количественным характеристикам:* снаряженная масса; время разгона автомобиля до 100 км/ч; максимальная скорость; расход топлива; мощность двигателя; крутящий момент; объем багажника (исходные данные для четырех моделей автомобилей одного класса Ford Focus, Mazda3, Nissan Sentra, Skoda Octavia, взятые из работы [10], приведены в табл. 7);
- *качественным характеристикам:* рабочее место водителя; салон; ходовые качества; комфорт; приспособленность к России (подробное их описание приводится ранее при рассмотрении МАИ).

На базе исходных данных, приведенных в табл. 7, на основе метода спектрального анализа, с помощью компьютерной программы (разработана в программном продукте Delphi 7, число итераций $k = 10$) получены следующие результаты по сравнительной оценке вариантов автомобилей и количественных характеристик, используемых для сравнения.

В результате вычислений был получен ранжированный ряд вариантов автомобилей: Nissan Sentra, Skoda Octavia, Mazda3, Ford Focus (итоговые показатели: 1; 0,9822; 0,9798; 0,9759). При этом характеристики вариантов системы

освещения по важности в результате вычислений ранжируются следующим образом: Снаряженная масса, Крутящий момент, Максимальная скорость, Время разгона до 100 км/ч, Мощность двигателя, Расход топлива, Объем багажника (итоговые показатели: 1,00; 0,9951; 0,9911; 0,9854; 0,9592; 0,9123; 0,8713).

Опираясь на оценки экспертов по указанным ранее качественным характеристикам четырех моделей автомобилей, приведенные в [10], с помощью компьютерной программы, составленной на основе алгоритма решения задачи на базе МАИ в программной среде Turbo Delphi, был получен следующий ранжированный ряд вариантов автомобилей: Skoda Octavia, Ford Focus, Nissan Sentra, Mazda3 (итоговые показатели: 0,284; 0,276; 0,272; 0,168). Решение на основе МАИ приведено ранее при рассмотрении процедуры вычислений по указанному методу. Полученные результаты хорошо согласованы и совпадают с результатами экспертной оценки автомобилей, приведенной в [10].

Отметим следующие недостатки и достоинства методов спектрального анализа и анализа иерархий, полученные в ходе исследований (см. [11, 12]).

Преимущество метода спектрального анализа состоит в отсутствии субъективных оценок при подборе исходных показателей и в расчетах.

Рекомендуется его применять при сравнении по количественным характеристикам вновь разрабатываемых ТС, "вес" характеристик которых еще не определен однозначно, а также для сравнительного анализа ТС по числовым показателям, когда не требуется определять ранжированный ряд этих показателей.

Недостатки МСА: его применение возможно только для однозначно заданных числовых

Таблица 7

Основные характеристики автомобилей [10]

Модель автомобиля	Характеристики						
	Снаряженная масса, кг	Время разгона до 100 км/ч, с	Максимальная скорость, км/ч	Расход топлива, л	Мощность двигателя, кВт	Крутящий момент, Н·м	Объем багажника, л
Ford Focus	1284	11,8	195	8,7	92	159	388
Mazda3	1287	11,6	191	7,4	88	150	384
Nissan Sentra	1267	11,3	184	8,1	86	158	488
Skoda Octavia	1250	12,0	190	9,0	81	155	520

характеристик, отсутствует учет важности характеристик. Хотя при использовании МСА имеется возможность учета важности характеристик ТС при проведении расчетов, если эту важность заранее задать с привлечением экспертов, но тогда важность характеристик станет субъективной величиной.

Преимущества МАИ. Метод анализа иерархий применим при многовариантном выборе по совокупности характеристик. Данные характеристики, во-первых, могут быть как количественными, так и качественными или набором тех и других; во-вторых, имеют заданную важность ("вес").

К недостатку МАИ можно отнести субъективные качественные оценки важности критериев и оценки приоритета того или иного варианта ТС по каждому из выбранных критериев. В последующих вычислениях по МАИ субъективизм отсутствует. Кроме того, для определения согласованности мнений экспертов вычисляют отношения согласованности.

Таким образом, МСА и МАИ имеют недостатки и достоинства и их использование определяется условиями конкретной задачи выбора наилучшего варианта ТС.

Вместе с тем, покупателю технической системы (например, автомобиля) хотелось бы иметь количественную интегральную оценку по всей совокупности характеристик, на базе которой можно провести выбор наилучшей ТС (наилучшего автомобиля) из предложенной совокупности.

В работе [13] рассмотрены различные свертки критериев, которые были использованы для принятия решения по выбору наилучшего варианта ТС по совокупности ее характеристик.

Для нивелирования недостатков и усиления достоинств методов предлагается комплексный подход, позволяющий принимать решение по выбору наилучшего варианта ТС по количественным и качественным характеристикам на основе двух методов.

В данном случае процедура получения интегральной количественной оценки по каждому из вариантов технической системы и порядок построения ранжированного ряда вариантов этой системы будут следующими.

1. Определяем характеристики предложенных вариантов технических систем, как количественные, так и качественные, по которым будем выбирать наилучшую техническую систему.

2. Проводим ранжирование вариантов ТС и их количественных характеристик на основе метода спектрального анализа. В результате расчетов получаем ранжированный ряд вариантов ТС и ранжированный ряд важности рассматриваемых характеристик.

3. Выполняем нормирование полученных результатов ранжированного ряда вариантов ТС. Для нормирования применяем подход, используемый в методе анализа иерархий. Получаем вектор приоритетов для вариантов ТС по количественным характеристикам.

4. Проводим ранжирование вариантов ТС и их только качественных характеристик на основе МАИ. В результате расчетов получаем вектор приоритетов качественных характеристик (критериев оценки вариантов) и вектор глобальных приоритетов вариантов ТС, полученный для качественных характеристик ТС, на базе последнего получаем ранжированный ряд вариантов ТС.

5. Суммируем полученные векторы приоритетов для количественных и качественных характеристик для каждого варианта ТС.

6. Строим ранжированный ряд вариантов ТС на основе следующего критерия: чем больше сумма соответствующих оценок, полученных в п. 5 алгоритма, тем выше ранг варианта.

7. Анализируем полученные результаты и делаем выводы о наилучшем варианте ТС (чем выше ранг, тем лучше вариант ТС) и о важности той или иной характеристики в принятии решения.

Проведем дальнейшие вычисления на основе данного алгоритма для выбранного примера по выбору лучшего автомобиля из четырех предложенных к рассмотрению. Учтем, что пункты 1, 2, 4 алгоритма уже выполнены, поэтому проведем нормирование полученных результатов ранжированного ряда автомобилей по количественным характеристикам. В результате нормирования был получен ранжированный ряд автомобилей: Nissan Sentra, Skoda Octavia, Mazda3, Ford Focus (итоговые показатели: 0,2539; 0,2494; 0,2488; 0,2478). После выполнения п. 5, 6 алгоритма получим следующий ранжированный ряд автомобилей: Skoda Octavia; Nissan Sentra; Ford Focus; Mazda3 (итоговые показатели: 0,533; 0,526; 0,524; 0,417).

Проанализируем полученные результаты. Если рассматривать ранжированный ряд, полученный на основе МАИ (Skoda Octavia, Ford

Focus, Nissan Sentra, Mazda3), то учет количественных характеристик вывел на 2-е место NissanSentra, сместив на 3-е место Ford Focus, но преимущество Nissan Sentra перед Ford Focus незначительно. Остальные автомобили остались на своих местах. Лучшим по количественным и качественным характеристикам является автомобиль Skoda Octavia.

Надо отметить, что результаты оценки зависят от выбранной совокупности характеристик (количественных и качественных). Важно правильно выбрать из большого числа характеристик основные количественные и наиболее важные качественные показатели.

Заключение

Выбор конкретных характеристик технических систем для сравнительной оценки вариантов является субъективным и зависит от руководителя организации (предприятия) или покупателя данной ТС. Так, например, покупатель автомобиля к основным качественным характеристикам, важным лично для него, может отнести дизайн или цвет автомобиля, а не вместимость багажника, или геометрическую проходимость.

Совместное применение методов спектрального анализа и анализа иерархий при выборе наилучшего варианта технической системы позволяет несколько снизить субъективизм в принятии решения, особенно в условиях, когда требуется быстрая оценка альтернативных вариантов.

Предлагаемая комплексная методика принятия решений по выбору наилучшего варианта технической системы по совокупности ее количественных и качественных характеристик может быть использована не только для решения задачи выбора лучшего автомобиля из имеющегося набора, но и для выбора любых других технических систем.

Список литературы

1. Романов П. С. К выбору рационального комплекта технологического оборудования для передвижных автомобильных ремонтных мастерских // Вестник Коломенского государственного педагогического института. 2008. № 1 (5). С. 129–135.

2. Романов П. С. Обоснование выбора рационального комплекта технологического оборудования для передвижных автомобильных ремонтных мастерских в условиях неопределенности // Вестник Коломенского государственного педагогического института. 2009. Вып. № 1 (7). С. 91–96.

3. Романов П. С. Методика выбора варианта станции технического обслуживания автомобилей в рыночных условиях // Вестник Московского государственного областного социально-гуманитарного института. 2011. Т. 1, № 11. С. 147–151.

4. Кафиев И. Р., Романов П. С. Выбор рационального варианта устройства освещения мастерских по ремонту сельскохозяйственных машин // Вестник Башкирского государственного аграрного университета. 2013. № 2 (26). С. 83–86.

5. Кафиев И. Р., Романов П. С. К вопросу проектирования устройства освещения ремонтных мастерских сельскохозяйственных предприятий с использованием энергосберегающих технологий // Достижения современной науки в области энергосбережения. Материалы Первой международной научно-практической конференции. Чебоксары: ЧСХА, 2013. С. 145–148.

6. Кафиев И. Р., Романов П. С. Методика выбора проекта системы освещения для ремонтных мастерских сельскохозяйственных предприятий в рыночных условиях // Вестник Башкирского государственного аграрного университета. 2014. № 2. С. 78–82.

7. Кафиев И. Р., Романов П. С., Романова И. П. Методика принятия решения по выбору проекта системы освещения для ремонтных мастерских сельскохозяйственных предприятий // Вестник Башкирского государственного аграрного университета. 2015. № 1 (33). С. 82–89.

8. Кафиев И. Р., Романов П. С., Романова И. П. К вопросу выбора проекта системы освещения для ремонтных мастерских сельскохозяйственных предприятий в условиях неопределенности // Современные проблемы науки и образования в техническом вузе. Материалы II Международной научно-практической конференции. Стерлитамак: УГАТУ, 2015. С. 162–167.

9. Саати Т., Кернс К. Аналитическое планирование. Организация систем / Пер. с англ. М.: Радио и связь, 1991. 224 с.

10. Тестирование Ford Focus, Mazda3, Nissan Sentra, Skoda Octavia // За рулем. 2015. № 10 (1012). С. 92–93.

11. Кафиев И. Р., Романов П. С., Романова И. П. Сравнительная оценка методов принятия решений по выбору проекта системы освещения для ремонтных мастерских сельскохозяйственных предприятий // Вестник Башкирского государственного аграрного университета. 2015. № 3 (35). С. 48–52.

12. Кафиев И. Р., Романов П. С., Романова И. П. Сравнение двух методик принятия решений на примере выбора рационального проекта системы освещения для ремонтных мастерских сельскохозяйственных предприятий // Российский электронный научный журнал. 2015. № 3 (17). С. 19–35.

13. Романов П. С. Обоснование путей построения автоматизированных систем управления артиллерийскими формированиями на основе новых информационных технологий. Коломна: КГПИ, 2005. 398 с.

P. S. Romanov, D. Sc., Professor, e-mail: romanov_p_s@mail.ru,
Kolomna Institute (branch) of Federal State Budgetary Educational Institution
of Higher Education "Moscow Polytechnical University" (Kolomna),

I. P. Romanova, Ph. D., e-mail: irom84@mail.ru,
Federal State Budgetary Educational Institution of Higher Education "National Research Moscow State
University of Civil Engineering" ((NRU MSUCE)), Moscow, 129337, Russian Federation

Decision-making Methodology for Choosing the Best Technical System for the Recruitment of its Characteristics

The article discusses an integrated approach to decision-making for choosing the best technical system in the quantitative and qualitative characteristics. The evaluation of options and selection of the best variant of technical system it is proposed to carry out based on the methods constructed using methods of spectral analysis and analytic hierarchy process. The description of the methods, two formulations of the problem and its solution algorithms these methods. Analyzes the advantages and disadvantages of these methods. Indicates that combined use of methods of spectral analysis and analytic hierarchy process in selecting the best variant of technical system allows to somewhat reduce the subjectivity in decision-making, especially in conditions that require rapid evaluation of alternatives, and to mitigate the disadvantages and enhance the advantages of these methods. The method may find application in the selection of the best variant of technical system for the totality of its quantitative and qualitative characteristics in various sectors of the economy.

Keywords: the decision, technical system, the method of spectral analysis, analytic hierarchy process, a comprehensive methodology for the selection of the technical system

DOI: 10.17587/it.24.445-453

References

1. **Romanov P. S.** K vyboru racional'nogo kompleksa tehnologicheskogo oborudovaniya dlja peredvizhnyh avtomobil'nyh remontnyh masterskih, *Vestnik Kolomenskogo gosudarstvennogo pedagogicheskogo instituta*, 2008, no. 1 (5), pp. 129–135 (in Russian).

2. **Romanov P. S.** Obosnovanie vybora racional'nogo kompleksa tehnologicheskogo oborudovaniya dlja peredvizhnyh avtomobil'nyh remontnyh masterskih v usloviyah neopredelennosti, *Vestnik Kolomenskogo gosudarstvennogo pedagogicheskogo instituta*, 2009, no. 1 (7), pp. 91–96 (in Russian).

3. **Romanov P. S.** Metodika vybora variant stancii tehničeskogo obsluzhivaniya avtomobilej v rynochnykh usloviyah, *Vestnik Moskovskogo gosudarstvennogo oblastnogo social'no-gumanitarnogo instituta*, 2011, no. 11, pp. 147–151 (in Russian).

4. **Kafiev I. R., Romanov P. S.** Vyor racional'nogo variant ustrojstva osveshhenija masterskih po remontu sel'skohozjajstvennyh mashin, *Vestnik Bashkirskogo gosudarstvennogo agrarnogo universiteta*, 2013, no. 2 (26), pp. 83–86 (in Russian).

5. **Kafiev I. R., Romanov P. S.** K voprosu proektirovaniya ustrojstva osveshhenija remontnyh masterskih sel'skohozjajstvennyh predpriyatij s ispol'zovaniem jenergosberegajushhih tehnologij, *V sbornike: "Dostizhenija sovremennoj nauki v oblasti jenergosberezhenija"*. *Materialy pervoj mezhdunarodnoj nauchno-praktičeskoj konferencii*. Cheboksary, CSHA, 2013, pp. 145–148 (in Russian).

6. **Kafiev I. R., Romanov P. S.** Metodika vybora proekta sistemy osveshhenija dlja remontnyh masterskih sel'skohozjajstvennyh predpriyatij v rynochnykh usloviyah *Vestnik Bashkirskogo gosudarstvennogo agrarnogo universiteta*, 2014, no. 2, pp. 78–82 (in Russian).

7. **Kafiev I. R., Romanov P. S., Romanova I. P.** Metodika prinjatija reshenija po vyboru proekta sistemy osveshhenija dlja remontnyh masterskih sel'skohozjajstvennyh predpriyatij, *Vestnik Bashkirskogo gosudarstvennogo agrarnogo universiteta*, 2015, no. 1 (33), pp. 82–89 (in Russian).

8. **Kafiev I. R., Romanov P. S., Romanova I. P.** K voprosu vybora proekta sistemy osveshhenija dlja remontnyh masterskih sel'skohozjajstvennyh predpriyatij v usloviyah neopredelennosti (To the choice of project lighting system for repair shops of agricultural enterprises in conditions of uncertainty), *V sbornike: Sovremennye problemy nauki i obrazovaniya v tehničeskom vuze Materialy II Mezhdunarodnoj nauchno-praktičeskoj konferencii*, Sterlitamak, UGATU, 2015, pp. 162–167 (in Russian).

9. **Saati T., Kerns K.** *Analiticheskoe planirovanie*, Moscow, Radio and communication, 1991. 224 p. (in Russian).

10. **Testirovanie** Ford Focus, Mazda3, Nissan Sentra, Skoda Octavia (Testing the Ford Focus, Mazda3, Nissan Sentra, Skoda Octavia), *Za rulem*, 2015, no. 10 (1012), pp. 92–93 (in Russian).

11. **Kafiev I. R., Romanov P. S., Romanova I. P.** Sravnitel'naja ocenka metodov prinjatija reshenij po vyboru proekta sistemy osveshhenija dlja remontnyh masterskih sel'skohozjajstvennyh predpriyatij, *Vestnik Bashkirskogo gosudarstvennogo agrarnogo universiteta*, 2015, no. 3 (35), pp. 48–52 (in Russian).

12. **Kafiev I. R., Romanov P. S., Romanova I. P.** Sravnenie dvuh metodik prinjatija reshenij na primere vybora racional'nogo proekta sistemy osveshhenija dlja remontnyh masterskih sel'skohozjajstvennyh predpriyatij, *Rossijskij jelektronnyj nauchnyj zhurnal*, 2015, no. 3 (17), pp. 19–35 (in Russian).

13. **Romanov P. S.** Obosnovanie putej postroenija avtomatizirovannyh sistem upravlenija artilerijskimi formirovanijami na osnove novykh informacionnyh tehnologij, Kolomna: KGPI, 2005. 398 p. (in Russian).

А. Н. Божко, канд. техн. наук, доц., abozhko@inbox.ru,
Московский государственный технический университет им. Н. Э. Баумана

Формализация размерных ограничений на проектные решения сборочного передела

Рассматривается влияние системы конструкторских размерных цепей на допустимые последовательности сборки и разбиения изделия на сборочные единицы. При сборке изделий должны выполняться правила связности и замкнутости конструкторских размерных цепей. Показано, что допустимые последовательности сборки представляют собой линейные продолжения иерархического порядка, который индуцирован размерной схемой на множестве деталей изделия. Даны численные оценки ограничений, которые размерные схемы накладывают на проектные решения сборочного этапа. Оказалось, что цепной способ простановки размеров является самым жестким, координатный — самым "толерантным". Разработано математическое описание принципа замкнутости размерной цепи. Показано, что для локализации размерной задачи требуется совпадение двух отношений эквивалентности на множестве деталей изделия. На основе полученных результатов сформулированы два принципа согласования размерной и механической структур изделия. Полученные результаты могут быть использованы для генерации рациональных размерных схем в интегрированных CAD/CAM-системах.

Ключевые слова: размерная цепь, последовательность сборки, разбиение на сборочные единицы, иерархический порядок, линейный порядок, размерный граф, отношение толерантности, отношение эквивалентности

Введение

Система конструкторских размеров — это важная часть проекта любого сложного изделия. Совокупность размеров и связанных с ними конструктивных элементов (деталей, поверхностей, осей и пр.) образует размерную структуру изделия. Размерная структура оказывает значительное влияние на содержание сборочных работ и экономические характеристики производства. Например, длинные размерные цепи затрудняют сборку с применением экономичных методов полной и неполной взаимозаменяемости. Размерные цепи с большим числом общих звеньев могут быть причиной конфликтов, когда реализация одной цепи влечет за собой потерю точности другой. В проектной практике эти свойства размерной структуры объективированы в форме двух правил, которые обычно называют *принципами кратчайшей размерной цепи* и *минимального числа пересечений*.

От размерной структуры изделия зависят проектные решения сборочного передела. В работе [1] отмечается, что анализ конструкторских размерных цепей помогает понять взаимосвязи деталей, составляющих изделие и ее сборочные единицы, найти рациональные методы достижения точности замыкающих зве-

ньев, проверить корректность простановки размеров и допусков, **наметить последовательность сборки изделия и ее сборочных единиц**. В самом деле, простановка конструкторских размеров индуцирует на множестве деталей изделия отношения порядка и эквивалентности. Эти отношения не должны противоречить упорядоченности деталей в процессе сборки изделия и его разбиению на сборочные единицы. Два простых примера, для которых это требование не выполняется, приведены на рис. 1.

Размерная структура создается на этапе проектирования, а реализуется в процессе сборки изделия, когда ревизия размерной схемы влечет за собой большие экономические и временные издержки. Поэтому в состав CAD/CAM-систем должны входить средства, которые позволяют оценить корректность размерной схемы по критериям сборочного производства и выработать рекомендации по настройке размерной и механической структур изделия. В данной статье предложены оценки жесткости и принципы согласования системы конструкторских размерных цепей. Полученные результаты могут быть использованы для математического и методического обеспечения интегрированных систем, реализующих современные парадигмы проектирования (например, Design for manufacturing, Design for assembly, Concurrent

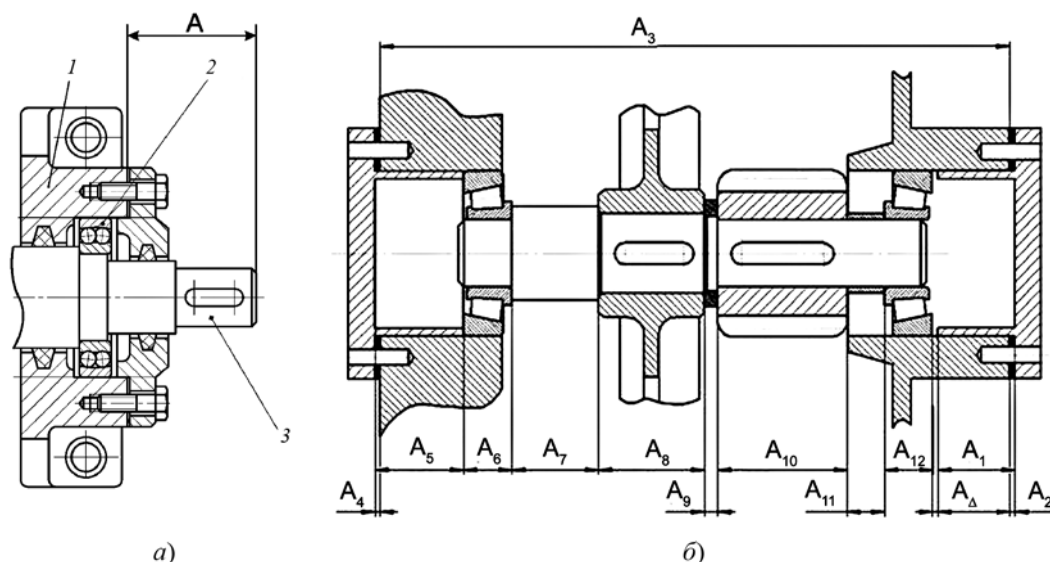


Рис. 1. Примеры конструкций, у которых не согласованы размерная и механическая структуры

engineering), которые требуют согласованной и параллельной работы конструкторов и технологов над проектом сложного изделия [2].

1. Конструкторские размерные цепи

Конструкторские размерные цепи разделяют на поддетальные и сборочные [3]. Первые задают размерные связи между геометрическими элементами деталей, вторые определяют координацию и точность изделия. Составляющими звеньями сборочной размерной цепи могут быть собственные размеры деталей, размеры между разными деталями, поддетальные размерные цепи и производные сборочные размерные цепи. Замыкающие звенья, их номиналы и допуски определяют по условиям эксплуатации или сборки изделия. Замыкающими звеньями служат линейные или угловые размеры между геометрическими элементами (поверхностями, осями и др.), принадлежащими различным деталям. В статье рассматриваются только линейные сборочные размерные цепи и их влияние на основные проектные решения сборочного передела: последовательность сборки и декомпозицию изделия на сборочные единицы.

Замыкающее звено конструкторской размерной цепи получается автоматически после монтажа деталей, которые служат носителями составляющих размеров. Это значит, что размерная схема индуцирует на множестве деталей порядковое отношение, которое не должно противоречить последовательности сборки и

разбиению изделия на сборочные единицы. На рис. 1 приведены примеры двух конструкций, у которых это требование не выполняется, т. е. размерная и механическая структуры изделий не являются согласованными.

В примере на рис. 1, а размер A является составляющим. Легко видеть, что после сборки фрагмента, состоящего из колеса 1, подшипника 2 и вала 3, возникнет неразрешимая (замкнутая) размерная цепь, которую образуют размер A и габаритные размеры трех данных деталей.

В примере на рис. 1, б размерная цепь с замыкающим размером A_{Δ} состоит из 12 звеньев (A_1, \dots, A_{12}). Это очень длинная цепь, которая не может быть реализована экономичными методами полной или неполной взаимозаменяемости. Кроме того, носители данной цепи принадлежат всему изделию, что делает невозможным локализацию и решение точностных задач в пределах одной сборочной единицы.

В работах [4–9] обсуждаются различные аспекты сложной проблемы синтеза рациональных размерных схем в системах автоматизированного проектирования. В статьях [4, 5] предложены способы генерации размерной схемы по трехмерной геометрической модели изделия. Оптимальное распределение допуска на замыкающее звено по звеньям размерной цепи рассматривается в работе [6]. Статья [7] посвящена синтезу трехмерной геометрической модели изделия с учетом заданной размерной схемы. В работе [8] предложен метод моделирования конструкторских размерных

цепей с помощью аппарата теории графов. Проблема синтеза размерной схемы, которая заведомо может быть реализована с помощью методов полной и неполной взаимозаменяемости, обсуждается в статье [9].

Рассмотренные работы по автоматизированному синтезу размерной структуры сложных изделий имеют ряд общих недостатков:

- основное внимание уделено подетальным конструкторским цепям;
- для генерации рациональных размерных структур используются только классические критерии минимального числа звеньев и минимального числа пересечений связанных цепей;
- не обсуждается влияние размерной структуры сложного изделия на допустимые последовательности сборки и декомпозиции на сборочные единицы.

2. Базовые правила реализации размерной схемы

Основные взаимные связи между размерной схемой и проектными решениями сборочного этапа сформулируем в виде двух основных правил реализации.

1. *Правило связности* "растущей" размерной цепи. В сборочной операции можно реализовать такой составляющий размер конструкторской сборочной размерной цепи, который инцидентен собранному фрагменту изделия.

2. *Правило замкнутости*. Детали, которые являются носителями составляющих звеньев сборочной размерной цепи, должны принадлежать одной сборочной единице (СЕ).

Составляющие звенья сборочных размерных цепей разделим на два типа. К первому типу относятся размеры, соединяющие геометрические элементы одной детали, например размеры $A_1, A_2, A_4, \dots, A_{12}$ на рис. 1, б. Ко второму типу относятся размеры, координирующие элементы разных деталей изделия (A_3 на рис. 1, б).

Пусть все звенья размерной цепи относятся к первому типу. В этом случае любая последовательность их реализации размеров индуцирует очевидную упорядоченность деталей, которые являются их носителями.

3. Формализация размерных ограничений на последовательности сборки изделия

Представим размерную схему в виде графа, в котором ребра соответствуют размерам, а вер-

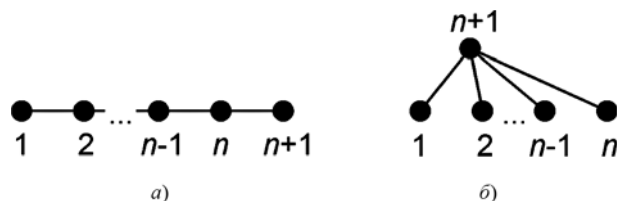


Рис. 2. Размерные графы:

a — цепного метода; b — координатного метода

шины описывают их инцидентность (контакты поверхностей, совпадение осей и др.). Рассмотрим два размерных графа, соответствующих основным методам простановки размеров: цепному (рис. 2, a) и координатному (рис. 2, b).

Обозначим CR_n^1 — число всех последовательностей реализации $(n + 1)$ -вершинной цепи (рис. 2, a), которые удовлетворяют правилу связности, а TR_n^1 — число всех таких последовательностей для $(n + 1)$ -вершинного дерева (рис. 2, b). Здесь единичный верхний индекс означает принадлежность всех элементов размерной цепи к первому типу.

Легко видеть, что любая последовательность реализации размеров дерева сохраняет связность цепи в корне, поэтому $TR_n^1 = n!$.

Подсчитаем CR_n^1 . Число допустимых реализаций $(n + 1)$ -вершинной цепи зависит от выбора первого ребра. Если исполнение начать с левого (правого) ребра, то допустимой является только одна последовательность звеньев, которая описывается порядком вершин $3, 4, \dots, n$ ($n - 2, n - 3, \dots, 1$ — при выборе правого ребра). Пусть первым будет размер, соответствующий любому внутреннему ребру цепи. Тогда цепь разделится на две подцепи, реализацию которых можно продолжить единственным способом: левая подцепь исполняется справа налево, а правая — в противоположном направлении. Это превращает подцепи в упорядоченные множества. Пусть $r_i = (i, i + 1)$ — ребро, которое соединяет вершины с номерами i и $i + 1$. На рис. 3 показаны упорядочения размеров, порожденные выбором первого звена.

Обозначим $L(r_i)$ — число допустимых реализаций размерных звеньев цепи (см. рис. 2, a) при условии, что первым исполняется размер r_i (рис. 3). Очевидно, что $CR_n^1 = \sum_{i=1}^n L(r_i)$. Для расчета CR_n^1 воспользуемся методом перечисления числа линейных продолжений дискретной суммы упорядоченных множеств, описанным в работе [10].

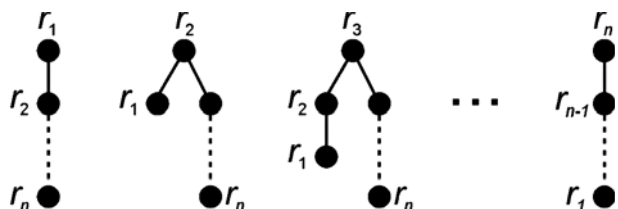


Рис. 3. Варианты упорядочения размеров, которые порождаются выбором первого звена

Пусть (P_i, \leq_i) , $i = \overline{1, r}$, — семейство упорядоченных множеств.

Определение. Упорядоченное множество (P, \leq) называется дискретной суммой упорядоченных подмножеств (P_i, \leq_i) , $i = \overline{1, r}$, если $P = \bigcup_{i=1}^r P_i$ и $\forall a, b \in P$, $a \leq b$ тогда и только тогда, когда $\forall a, b \in P$, $a \leq b$ [11].

Число линейных продолжений упорядоченного множества можно найти по формуле

$$N = \frac{m!}{m_1! m_2! \dots m_r!} N_1 N_2 \dots N_r, \quad (1)$$

где $m_i = |P_i|$, $m = |P| = \sum_{i=1}^r m_i$, а N_i — число линейных продолжений упорядоченного множества (P_i, \leq_i) , $i = \overline{1, r}$ [10].

Используя формулу (1), найдем число линейных продолжений упорядоченного множества, общий вид которого показан на рис. 4. В терминах теории графов это корневое дерево, каждое поддерево которого является цепью.

Все линейные продолжения данного порядка начинаются с корневой вершины. Если убрать эту вершину из дерева, то получим

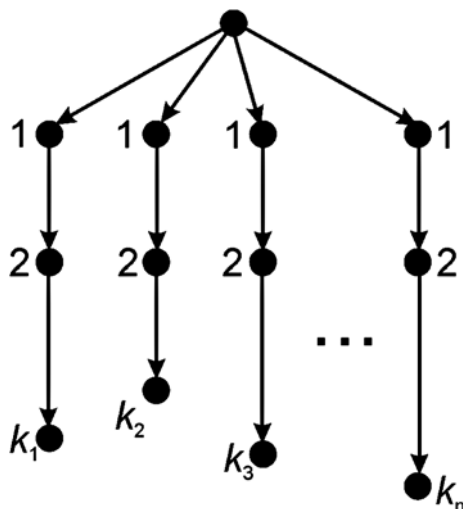


Рис. 4. Иерархический частичный порядок

дискретную сумму r упорядоченных подмножеств, каждое из которых представляет собой цепь. Любая цепь имеет только одно линейное продолжение, поэтому $N_1 = N_2 = \dots = N_r = 1$, поэтому формула (1) примет вид

$$N = \frac{(k_1 + k_2 + \dots + k_n)!}{k_1! k_2! \dots k_n!}, \quad (2)$$

где k_i , $i = \overline{1, n}$, — число вершин в i -й ветви, растущей из корня дерева.

Теперь $\forall i = 2, \dots, n-1$ $L(r_i)$ можно найти как число линейных продолжений упорядоченного множества, которое представляет собой дискретную сумму двух линейных порядков, состоящих из $i-1$ и $n-i$ вершин, т. е. $k_1 = i-1$, а $k_2 = n-i$. Подставив эти значения в формулу (2), получим $L(r_i) = \frac{(n-1)!}{(i-1)!(n-i)!}$. Отсюда

да $CR_n^1 = \sum_{i=2}^{n-1} \frac{(n-1)!}{(i-1)!(n-i)!} + 2$. Внесем одну единицу под операцию суммирования, получим

$CR_n^1 = \sum_{i=1}^{n-1} \frac{(n-1)!}{(i-1)!(n-i)!} + 1$ (теперь суммирование начинается с единицы). Подстановка $j = i-1$

даст $CR_n^1 = \sum_{j=0}^{n-2} \frac{(n-1)!}{j!(n-j-1)!} + 1$. Представим

единицу в виде $1 = \frac{(n-1)!}{(n-1)!0!}$ и внесем это сла-

гаемое под знак суммирования. В результате получим полную сумму биномиальных коэффициентов для числа $n-1$, которая равна

$$CR_n^1 = \sum_{j=0}^{n-1} \frac{(n-1)!}{j!(n-j-1)!} = 2^{n-1}. \quad (3)$$

Рассмотрим вариант сборочной размерной цепи, в которой все составляющие размеры относятся ко второму типу, т. е. они соединяют геометрические элементы разных деталей. Если во множестве деталей, образующих сборочную размерную цепь, выбрать базовую деталь (т. е. деталь, которая устанавливается первой), то данная цепь индуцирует на множестве иерархическое отношение частичного порядка, а допустимые последовательности сборки представляют собой линейные продолжения этого порядка. Выразим данное утверждение в точных терминах.

Пусть $A \subseteq X = \{x_i\}_{i=1}^n$ — множество деталей изделия X , которые своими геометрическими элементами участвуют в формировании сборочной размерной цепи. Для любых $x_k, x_l \in A$

обозначим $x_k R x_l$ ситуацию, когда две детали x_k и x_l связаны размером. Пусть $z \in A$ — деталь, которая устанавливается первой в данной размерной цепи. Детали x и y состоят в отношении иерархического частичного порядка xTy тогда и только тогда, когда существует последовательность $x_{i_1}, x_{i_2}, \dots, x_{i_r} \in A$ такая, что $\forall k = \overline{1, r-1} x_{i_k} R x_{i_{k+1}}, x_{i_1} = z, x_{i_r} = y$ и $\exists t \in \{2, r-1\} | x = x_{i_t}$. Иерархически упорядоченные множества такого типа представляют в виде корневых деревьев, в которых корнем служит базовая деталь, а дуги представляют отношение предшествования, индуцированное размерными связями. На рис. 5 показаны примеры двух иерархических порядков, которые порождаются цепным и координатным методами простановки размеров.

Обозначим $T(r)$ — иерархический порядок, который порождается на множестве A выбором корневой вершины с номером r ; $N(T(r))$ — число линейных продолжений порядка $T(r)$; CR_n^2 — число реализаций n -вершинной цепи; а TR_n^2 — число допустимых реализаций n -вершинного дерева, удовлетворяющих правилу связности "растущей" цепи. Верхний индекс означает, что все звенья размерных цепей принадлежат ко второму типу. Эти числа можно найти по формулам $CR_n^2 = \sum_{r=1}^n N(T(r))$ и $TR_n^2 = \sum_{r=1}^n N(T(r))$.

Легко видеть, что каждому иерархическому порядку $T(r)$, порожденному цепью, можно поставить в соответствие одно и только одно корневое дерево семейства, показанного на рис. 3. Поэтому справедливо равенство

$$CR_n^2 = \sum_{r=1}^n N(T(r)) = \sum_{i=1}^n L(r_i) = 2^{n-1}. \quad (4)$$

Рассмотрим иерархический порядок T (рис. 5, б), который порождает координатный

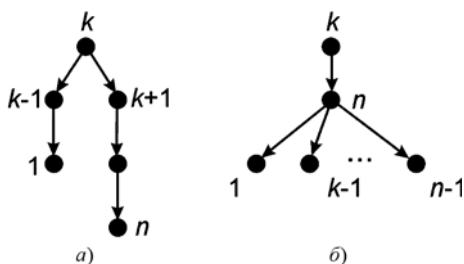


Рис. 5. Иерархические порядки T для цепного (а) и координатного (б) методов простановки размеров

способ простановки размеров. Легко видеть, что у данного дерева с корнем в вершине k существует $(n-1)!$ линейных продолжений. Кроме того, реализацию размерной цепи можно начать с любой из деталей с номерами от 1 до $n-1$, что дает еще $(n-1)(n-2)!$ линейных порядков. Общее число последовательностей, согласованных с данной размерной структурой, равно

$$TR_n^2 = (n-1)! + (n-1)(n-2)! = 2(n-1)!. \quad (5)$$

Числа $CR_n^1, CR_n^2, TR_n^1, TR_n^2$ являются измерителями силы ограничений, которые размерная схема изделия накладывает на допустимые последовательности сборки изделия. Для $n = 4, 5, \dots$ выполняются соотношения $CR_n^1 = CR_n^2 < TR_n^2$ и $TR_n^2 < TR_n^1$. Если $CR_2^1 = CR_2^2 = TR_2^1 = TR_2^2$, то, как легко видеть, $\lim_{n \rightarrow \infty} \frac{CR_n^1}{TR_n^1} = 0$ и $\lim_{n \rightarrow \infty} \frac{TR_n^1}{TR_n^2} = 0$. Найдем $\lim_{n \rightarrow \infty} \frac{CR_n^1}{TR_n^1}$. Представим CR_n^1 в виде суммы $CR_n^1 = \sum_{i=2}^{n-1} \frac{(n-1)!}{(i-1)!(n-i)!} + 2$, тогда

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{CR_n^1}{TR_n^1} &= \lim_{n \rightarrow \infty} \frac{\sum_{i=2}^{n-1} \frac{(n-1)!}{(i-1)!(n-i)!} + 2}{2(n-1)!} = \\ &= \lim_{n \rightarrow \infty} \left(\sum_{i=2}^{n-1} \frac{1}{2(i-1)!(n-i)!} + \frac{1}{(n-1)!} \right) = 0. \end{aligned}$$

Приведенные соотношения показывают, что цепной способ простановки размеров — самый жесткий. Он отсекает большое число последовательностей, для которых не выполняется правило связности "растущей" размерной цепи. Из всех рассмотренных самым "толерантным" является координатный способ образмеривания с поддетальными составляющими размерами. Разница между размерными схемами быстро растет с увеличением мощности размерной цепи.

Чаще всего в конструкторской практике используют так называемый комбинированный способ простановки размеров. В этом случае размеры могут ставиться от основной базы, промежуточных баз и образовывать размерные цепи произвольной длины. Размерные цепи, которые порождает комбинированный способ, описываются деревьями общего вида. Оценим жесткость ограничений, которые накладывает комбинированная размерная схема на возмож-

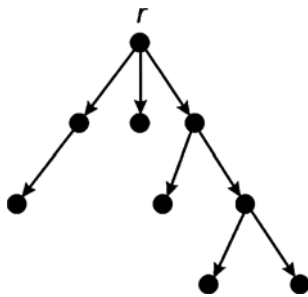


Рис. 6. Иерархический порядок $T(r)$, который порождается на дереве выбором корневой вершины r

ные последовательности сборки, по числу возможных линейных продолжений. Рассмотрим размерную схему, у которой все составляющие размеры принадлежат ко второму типу.

Как и ранее, выбор базовой детали, с которой начинается реализация размерной схемы, превращает размерное дерево в иерархически упорядоченное множество. Обозначим $T(r)$ — иерархический порядок, который порождает на дереве выбор корня r (рис. 6), а PR_n^2 — число допустимых реализаций размерной схемы, удовлетворяющих правилу "растущей" размерной цепи. Это число, как и ранее, равняется сумме линейных продолжений иерархических порядков $T(r)$, взятой по всем корневым вершинам r , т. е. $PR_n^2 = \sum_{r=1}^n N(T(r))$.

В работе [10] показано, что $N(T)$ — число линейных продолжений иерархического порядка можно найти по формуле

$$N(T) = \frac{|T|!}{|T| \times |T_1| \times |T_2| \times \dots \times |T_r|}, \quad (6)$$

где $|T|$ — число вершин дерева T ; $|T_i|$, $i = \overline{1, r}$ — число вершин во всех поддеревьях, содержащих более одной вершины.

Пусть T_1, T_2, T_3 — три n -вершинных иерархических порядка разного типа, т. е. $|T_1| = |T_2| = |T_3|$; T_1 — цепь, полученная выбором корня в любой внутренней вершине (рис. 7, а), T_2 —

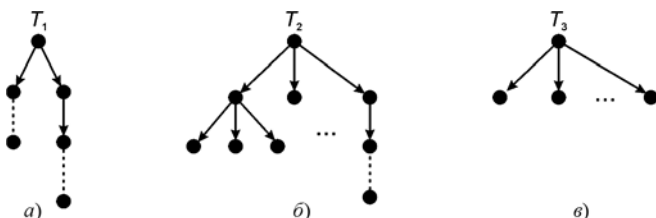


Рис. 7. Примеры различных иерархических порядков

дерево произвольной структуры (рис. 7, б) и T_3 — одноуровневое дерево (рис. 7, в).

Из формулы (6) следуют неравенства $N(T_1) \leq N(T_2) \leq N(T_3)$, связывающие числа линейных продолжений иерархических порядков разного типа. Почленное суммирование данных неравенств по всем n корневым вершинам дает следующие соотношения:

$$CR_n^2 \leq PR_n^2 \leq TR_n^2. \quad (7)$$

Формула (7) показывает, что самые жесткие ограничения на возможные последовательности сборки накладывает цепной способ простановки размеров, самые мягкие — координатный. Комбинированная система образмеривания занимает промежуточное положение между данными размерными схемами.

Пусть размерная схема является смешанной, т. е. в нее входят как поддетальные размеры, так и размеры, соединяющие геометрические элементы разных деталей. Пример такой схемы приведен на рис. 1, б. Очевидно, что любая смешанная размерная схема также может быть представлена в виде дерева. Правило связности "растущей" размерной цепи должно выполняться и в этом случае, поэтому любая допустимая последовательность реализации размеров смешанной размерной цепи является линейным продолжением древесного (иерархического) порядка, заданного на множестве деталей-носителей.

Опуская громоздкие формальные выкладки, скажем, что для любых размерных цепей (первого, второго типов и смешанных) равной мощности выполняется соотношение $CR_n \leq PR_n \leq TR_n$.

Формулы (4)–(7) дают оценки числа последовательностей сборки изделия, согласованных с размерной схемой. Эти числа достаточно велики, нижняя граница составляет $CR_n = 2^{n-1}$, где n — число вершин размерного графа. В большинстве случаев размерная цепь соединяет только часть деталей изделия. Любая перестановка деталей, не входящих в размерную цепь, сохраняет правило связности растущей цепи. Это значит, что число последовательностей сборки, согласованных с размерной схемой, превышает нижнюю границу, равную CR_n . Более точно, пусть N — число деталей изделия. Тогда число согласованных последовательностей больше, чем $(N - n)!CR_n$.

Механическая структура изделия (множество деталей, связанных соединениями и сопряжениями) накладывает значительно более сильные ограничения на допустимые последовательности сборки. Легко привести примеры структур, для которых существует только одна такая последовательность [12]. Это позволяет сформулировать следующий принцип согласования механической и размерной структур изделия.

Утверждение 1. Сборочная размерная цепь согласована с механической структурой изделия по критерию собираемости, если для каждой последовательности сборки изделия, удовлетворяющей условиям базирования, выполняется правило связности "растущей" размерной цепи.

4. Формализация размерных ограничений на сборочные декомпозиции

Система сборочных размерных цепей влияет не только на возможные последовательности сборки изделия. От этого проектного решения зависит способ разбиения изделия на сборочные единицы. В самом деле, любая сборочная размерная цепь решает определенную точностную задачу, которая формулируется по условиям эксплуатации или для обеспечения собираемости изделия или его сборочных единиц. В процессе сборки требуется не только реализовать сборочную размерную цепь, но и осуществить контрольные мероприятия, гарантирующие достижение заданной точности замыкающего звена. Эти технологические и метрологические операции выполняют на отдельных рабочих местах над группой деталей, которые должны быть собраны независимо от остальных фрагментов изделия. Поэтому все детали, которые являются носителями составляющих звеньев сборочной цепи, должны входить в одну сборочную единицу.

Выразим это утверждение в точных математических терминах. Пусть, как и ранее, $X = \{x_i\}_{i=1}^n$ — множество деталей изделия. Рассмотрим бинарное отношение $RC \subseteq X \times X$ на множестве X , в котором $xRCy$ тогда и только тогда, когда детали x и y некоторыми своими размерами или геометрическими элементами участвуют в формировании сборочной размерной цепи.

Легко убедиться, что для отношения RC выполняются следующие свойства:

- $xRCx \forall x \in X$ (рефлексивность);
- $xRCy \Rightarrow yRCx \forall x, y \in X$ (симметричность).

Из этого следует, что RC представляет собой отношение толерантности [11].

Обозначим $\tau(RC)$ — транзитивное замыкание отношения RC . Это минимальное бинарное отношение, включающее в себя RC и обладающее свойством транзитивности. Транзитивное замыкание можно найти по формуле $\tau(RC) = RC \cup RC^2 \cup \dots \cup RC^n$, где n — мощность носителя (число деталей изделия) [11].

Отношение $\tau(RC)$ обладает свойствами рефлексивности, симметричности и транзитивности, т. е. является отношением эквивалентности на множестве X . Известно, что любое бинарное отношение эквивалентности разбивает свой носитель на совокупность непересекающихся подмножеств, элементы которых попарно эквивалентны друг другу [11]. Классы эквивалентности отношения $\tau(RC)$ — это множества деталей, замкнутые относительно сборочных размерных цепей. В них входят детали, собственные размеры которых образуют одну сборочную размерную цепь или являются звеньями производных или связанных размерных цепей.

Пусть SD — некоторая r -уровневая схема декомпозиции изделия $X = \{x_i\}_{i=1}^n$ на сборочные единицы (рис. 8). Обозначим S_i^k , $i = \overline{1, p_k}$, $k = \overline{1, r}$, — совокупность сборочных единиц, принадлежащих k -му уровню иерархии, p_k — число сборочных единиц на данном уровне. Будем рассматривать множество деталей и собранное изделие X как тривиальные разбиения на сборочные единицы, т. е. $S_i^1 = x_i$, $i = \overline{1, n}$; $p_1 = n$ и $S^r = X$, $p_r = 1$.

На любом уровне $k = \overline{1, r}$ совокупность $S^k = \{S_1^k, S_2^k, \dots, S_{p_k}^k\}$ образует разбиение мно-

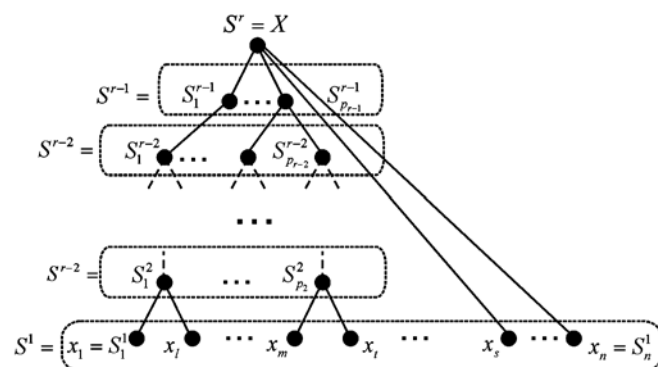


Рис. 8. Многоуровневая схема декомпозиции изделия на СЕ

жества $D^k = \bigcup_{i=1}^{p_k} S_i^k$; $D^k \subseteq X$ и $D^1 = D^r = X$. Элементы разбиения можно рассматривать как классы бинарного отношения эквивалентности $\lambda^k \subseteq D^k \times D^k$. Более точно это условие можно определить как $x\lambda^k y \Leftrightarrow \exists i | x, y \in S_i^k$.

На эквивалентностях $\lambda^k, k = \overline{1, r}$, в свою очередь зададим бинарное отношение подразбиения (измельчения) \triangleleft следующим образом:

$$\lambda^q \triangleleft \lambda^t \Leftrightarrow \forall i \in \{1, \dots, p_q\} \exists j \in \{1, \dots, p_t\} | S_i^q \subseteq S_j^t.$$

Легко проверить, что это отношение обладает свойствами:

- $\lambda^k \triangleleft \lambda^k, \forall k \in \{1, \dots, r\}$ — рефлексивности;
 - $\lambda^q \triangleleft \lambda^t, \lambda^t \triangleleft \lambda^q \Rightarrow \lambda^t = \lambda^q$ — антисимметричности;
 - $\lambda^q \triangleleft \lambda^t, \lambda^t \triangleleft \lambda^s \Rightarrow \lambda^q \triangleleft \lambda^s$ — транзитивности,
- т. е. является отношением нестрогого порядка [11].

Обозначим $\lambda = \bigcup_{k=1}^r \lambda^k$ — множество всех эквивалентностей, порожденных многоуровневой декомпозицией на сборочные единицы. Это множество вместе с заданным на нем отношением подразбиения станет упорядоченным множеством (λ, \triangleleft) .

Упорядоченное множество (λ, \triangleleft) обладает двумя важными свойствами. Во-первых, оно имеет наибольший элемент (порядковую единицу). Единицей множества (λ, \triangleleft) является эквивалентность λ^k , состоящая из одного класса X . Во-вторых, оно является древесным порядком, поскольку таковыми являются любые схемы декомпозиции SD (рис. 8).

В общем случае (λ, \triangleleft) может не быть линейным порядком (цепью), поскольку сборочные единицы нижних уровней могут входить в сборочные единицы верхних уровней, минуя промежуточные вхождения. Например, часто возникает ситуация, когда отдельные детали участвуют в сборке самого изделия, например детали x_s, \dots, x_n на рис. 8.

Образует объединенное множество отношений $\delta = \bigcup_{k=1}^r \lambda^k \cup \tau(RC)$, элементами которого служат эквивалентности, индуцированные разбиением изделия на СЕ и простановкой конструкторских размеров.

Рассмотрим множество δ вместе с порядком, который создает на нем отношение подразбиения. Обозначим это упорядоченное множество

(δ, \triangleleft) . Множество (δ, \triangleleft) является связным и, поскольку $\tau(RC) \triangleleft \lambda^r$, имеет наибольший элемент λ^r . В общем случае множество (δ, \triangleleft) может не быть древесным порядком, но в нем элемент $\tau(RC)$ покрывает только один элемент $\lambda^i \in \lambda$. Если бы существовали различные λ^i и λ^j такие, что $\tau(RC) \triangleleft \lambda^i$ и $\tau(RC) \triangleleft \lambda^j$, то $\lambda^i \cap \lambda^j \neq \emptyset$ (символ \triangleleft означает порядковое покрытие). Это, очевидно, противоречит древесной структуре упорядоченного множества (λ, \triangleleft) .

Используя введенную формализацию, можно дать точное описание согласованности системы сборочных размерных цепей и принятого варианта декомпозиции на сборочные единицы.

Утверждение 2. Сборочная размерная цепь согласована с механической структурой изделия по критерию расчленяемости, если в многоуровневом разбиении SD найдется такой уровень с номером $i, 1 < i < r$, для которого выполняется равенство $\lambda^i = \tau$.

Если выполняется утверждение 2, то поставленная конструктором точностная задача решена наиболее экономным способом, поскольку в сборочные единицы вошли только те детали, которые структурируют заданные конструктором размерные цепи изделия.

Пусть не существует i такого, что $\lambda^i = \tau$. Тогда найдется λ^j такой, что $\tau \triangleleft \lambda^j$. В этом случае хорошим согласованием можно считать такой вариант разбиения λ^j , для которого справедливо $f_r(\lambda^j) \rightarrow \max$. В этом случае сборка СЕ, включающих в себя конструкторские размерные цепи, будет выполнена на ранних стадиях производственного процесса. Поэтому общая сборка изделия будет максимально освобождена от технических мероприятий по контролю, пригонке и регулировке размерных цепей.

Заключение

1. Рассмотрено влияние системы конструкторских размерных цепей на основные проектные решения сборочного передела: последовательность сборки и декомпозицию изделия на сборочные единицы.

2. Показано, что сборка изделия должна быть организована таким образом, чтобы для конструкторских размерных цепей выполнялись два базовых правила: связность и замкнутость размерной цепи.

3. Разработана формализация правила связности и показано, что последовательность

сборки изделия должна представлять собой линейное продолжение иерархического порядка, порожденного размерной цепью на множестве деталей.

4. Рассчитаны интенсивности ограничений, которые накладывают на допустимые последовательности сборки различные способы приостановки конструкторских размеров (цепной, координатный и комбинированный).

5. Разработана формализация правила замкнутости и показано, что для согласования механической и размерной структур изделия по критерию расчленяемости требуется совпадение двух отношений эквивалентности на множестве деталей изделия.

Список литературы

1. Маталин А. А. Технологии машиностроения. М.: Лань, 2015. 512 с.
2. Anderson D. M. Design for Manufacturability: How to Use Concurrent Engineering to Rapidly Develop Low-Cost, High-Quality Products for Lean Production. CRC Press. 2014. 434 p.
3. Емельянов С. Г., Рудской А. М., Учаев П. Н. и др. Размерный анализ в машиностроении. М.: ТНТ, 2012. 332 с.
4. Gao Zhenbo, Wang Jing, Cao Yanlong, Yang Jiangxin. Automatic Generation of 3D Assembly Dimension Chain Based on Feature Model // Procedia CIRP. Vol. 43. 2016. P. 70–75. DOI:10.1016/j.procir.2016.02.012.
5. Heng Wang, Ruxin Ning, Yan Yan. Simulated toleranced CAD geometrical model and automatic generation of 3D dimension chains // The International Journal of Advanced Manufacturing Technology. Vol. 29, Iss. 9–10. 2006. P. 1019–1025. DOI: 10.1007/s00170-005-2617-x.
6. Payam Haghighi, Prashant Mohan, Jami J. Shah and Joseph K. Davidson. Automatic Detection and Extraction of Tolerance Stacks in Mechanical Assemblies // ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 2B: 40th Design Automation Conference Buffalo, New York, USA. 2014. P. 353–363. DOI: 10.1115/DETC2014-35315.
7. Pradeep K. Singh, Satish C. Jain, Pramod K. Jain. Advanced optimal tolerance design of mechanical assemblies with interrelated dimension chains and process precision limits // Computers in Industry. Vol. 56, Iss. 2. 2005. P. 179–194. DOI: 10.1016/j.compind.2004.06.008.
8. Ya Zhang, Lin Zhang. Graph Theory Solution Method to Solve The Complex Assembly Dimension Chain // Proc. of the 2017 7th International Conference on Mechatronics, Computer and Education Informationization (MCEI 2017). 2017. DOI: 10.2991/mcei-17.2017.140.
9. Zhenbo Gao, Zhexing Wang, Zijian Wu, Yanlong Cao. Study on Generation of 3D Assembly Dimension Chain. Procedia CIRP. 2015. Vol. 27. P. 163–168. DOI: 10.1016/j.procir.2015.04.060.
10. Розен В. В. Цель — оптимальность — решение (математические модели принятия оптимальных решений). М.: Радио и связь, 1982. 168 с.
11. Гуров С. И. Булевы алгебры, упорядоченные множества, решетки: Определения, свойства, примеры. М.: Либроком, 2013. 352 с.
12. Божко А. Н. Алгебраические модели сборки изделий // Наука и образование. МГТУ им. Н. Э. Баумана. Электрон. журн. 2016. № 12. DOI: 10.7463/1216.0852565.

A. N. Bozhko, Assistant Professor, e-mail: abozhko@inbox.ru,
Bauman Moscow State Technical University, Moscow

Formalization of Dimensional Constraints on Design Decisions of the Assembly Stage

The paper considers the influence of the system of design dimensional chains on permissible sequences of assembling and splitting the product into assembly units. When assembling products, the rules of connectivity and closure of design dimensional chains must be fulfilled. It is shown that admissible assembly sequences are linear extensions of a hierarchical order that is induced by a dimensional scheme on a set of parts of the product. Numerical estimates of the strength of constraints are given, which dimensional schemes impose design solutions of the assembly stage. It turned out that the chain method of dimensioning is the most rigid one, coordinating — the most "tolerant" one. A mathematical description of the principle of the closure of a dimensional chain is developed. It is shown that the localization of a dimensional problem requires the coincidence of two equivalence relations on the set of parts of the product. On the basis of the results obtained, two principles for reconciling the dimensional and mechanical structures of the product are formulated. The obtained results can be used to generate rational dimensional schemes in integrated CAD/CAM-systems.

Keywords: dimensional chain, assembly sequence, splitting into assembly units, hierarchical order, linear order, dimensional graph, tolerance relation, equivalence relation

DOI: 10.17587/it.24.454-463

References

1. **Matalin A. A.** *Tekhnologii mashinostroyeniya* (Technology of mechanical engineering), Moscow, Lan, 2015, 512 p. (in Russian).
2. **Anderson D. M.** Design for Manufacturability: How to Use Concurrent Engineering to Rapidly Develop Low-Cost, High-Quality Products for Lean Production. CRC Press, 2014, 434 p.
3. **Yemelyanov S. G., Rudskoy A. M., Uchayev P. N.** and etc. *Razmernyy analiz v mashi-nostroyenii* (Dimensional analysis in machine building), Moscow, TNT, 2012, 332 p. (in Russian).
4. **Gao Zhenbo, Wang Jing, Cao Yanlong, Yang Jiangxin.** Automatic Generation of 3D Assembly Dimension Chain Based on Feature Model, *Procedia CIRP*, 2016, vol. 43, pp. 70–75. DOI: 10.1016/j.procir.2016.02.012.
5. **Heng Wang, Ruxin Ning, Yan Yan.** Simulated toleranced CAD geometrical model and automatic generation of 3D dimension chains, *The International Journal of Advanced Manufacturing Technology*, 2006, vol. 29, iss. 9 – 10, pp. 1019–1025. DOI: 10.1007/s00170-005-2617-x.
6. **Payam Haghighi, Prashant Mohan, Jami J. Shah and Joseph K. Davidson.** Automatic Detection and Extraction of Tolerance Stacks in Mechanical Assemblies, *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 2B: 40th Design Automation Conference Buffalo, New York, USA, 2014, pp. 353–363. DOI: 10.1115/DETC2014-35315.
7. **Pradeep K. Singh, Satish C. Jain, Pramod K. Jain.** Advanced optimal tolerance design of mechanical assemblies with interrelated dimension chains and process precision limits, *Computers in Industry*, 2005, vol. 56, iss. 2, pp. 179–194. DOI: /10.1016/j.compind.2004.06.008.
8. **Ya Zhang, Lin Zhang.** Graph Theory Solution Method to Solve The Complex Assembly Dimension Chain, *Proceedings of the 2017 7th International Conference on Mechatronics, Computer and Education Informationization (MCEI 2017)*, 2017, DOI: 10.2991/mcei-17.2017.140.
9. **Zhenbo Gao, Zhexiong Wang, Zijian Wu, Yanlong Cao.** Study on Generation of 3D Assembly Dimension Chain, *Procedia CIRP*, 2015, vol. 27, pp. 163–168. DOI: 10.1016/j.procir.2015.04.060.
10. **Rozen V. V.** *Tsel – optimalnost – resheniye (matematicheskiye modeli prinya-tiya optimalnykh resheniy)* (The goal – optimality – a solution (mathematical models for the adoption of optimal solutions)), Moscow, Radio i svyaz, 1982, 168 p. (in Russian).
11. **Gurov S. I.** *Bulevy algebrы, uporyadochennyye mnozhestva, reshetki: Opredeleniya, svoystva, primery* (Boolean algebras, ordered sets, lattices: Definitions, properties, examples), Moscow, Librokom, 2013, 352 p. (in Russian).
12. **Bozhko A. N.** *Algebraicheskiye modeli sborki izdeliy* (Algebraic models of assembly of products), *Nauka i obrazovaniye. MGTU im. N. E. Baumana. Elektron. zhurn.*, 2016, no. 12. DOI: 10.7463/1216.0852565. (in Russian).



Институт прикладной математики им. М. В. Келдыша РАН
проводит с 17 по 22 сентября 2018 г.

XX Всероссийскую конференцию Научный сервис в сети Интернет



Конференция посвящена основным направлениям и тенденциям использования интернет-технологий в современных научных исследованиях. Основная цель конференции — предоставить возможность для обсуждения, апробации и обмена мнениями о наиболее значимых результатах, полученных ведущими российскими учеными за последнее время в данной области деятельности.

ТЕМАТИКА КОНФЕРЕНЦИИ

- Научные исследования и интернет, интернет-представительство научных организаций и проектов.
- Решение задач и обработка данных на суперкомпьютерах центров коллективного пользования.
- Интернет-проекты в области параллельных вычислений, математическое моделирование, вычислительные сервисы.
- Интернет-проекты для биомедицины.
- Модели и методы построения поисковых систем и систем навигации в интернете, технологии и системы распределенного хранения и обработки данных,
- Технологии и опыт построения информационных систем и баз данных, документации и результатов эксперимента на основе интернет-технологий.
- Цифровые библиотеки и библиографические базы, семантический веб, наукометрия в интернете.
- Онлайн-научная публикация, открытая наука, живая публикация, онлайн-рецензирование, мультимедийные иллюстрации.
- Популярный научный интернет, онлайн-энциклопедии, история науки в интернете.
- Интернет-активность ученого, персональная страница, профили ученого в библиографических базах, аттестация в интернете.
- Системное и инструментальное программное обеспечение, языки и модели программирования, формальные методы для интернет-технологий.

Сайт конференции: <http://agora.guru.ru/abrau2018>

С. В. Гаевой, канд. техн. наук, ст. преподаватель каф. ЭВМиС, e-mail: gaevserge@mail.ru;
В. М. А. Ахмед, аспирант каф. САПРиПК, e-mail: wesamalsofi@gmail.com;
С. А. Фоменков, д-р техн. наук, проф., проф. каф. САПРиПК, e-mail: saf@vstu.ru,
Волгоградский государственный технический университет, г. Волгоград, 400005

Использование имитационного моделирования для определения показателей обслуживания кластера

Важным способом анализа нагрузки вычислительного кластера является моделирование его работы (обслуживания входящих заданий) с использованием модели входящей нагрузки (модели самих входящих заданий). Нами уже построена сама имитационная модель кластера и определенный набор моделей входящей нагрузки. В данной работе рассмотрены особенности этой модели, ее программной реализации и процедура поиска конфигурации кластера на примере кластера UniLu-GAIA.

Ключевые слова: нагрузки вычислительных систем, модели входящей нагрузки, немасштабируемые задачи, время выполнения заданий, имитационное моделирование, стохастическая аппроксимация, показатели обслуживания, вычислительные кластеры, программа для моделирования

Введение

Проблема рационального выполнения параллельных и высокопроизводительных вычислений сейчас достаточно актуальна [1]. В частности, стоит вопрос оптимального балансирования нагрузки и подбора оптимальной производительности [2] вычислительных кластеров (ВК). Одним из возможных путей решения этой задачи является моделирование работы ВК — в том числе и имитационное [3]. Последнее требует построения математической модели ВК и поступающей на нее (т.е. входной) нагрузки [4].

В данной работе рассмотрим модель работы ВК, которая была построена нами ранее, особенности ее программной реализации и вопросы конфигурирования ВК, которые могут быть решены с помощью этих модели и программы.

1. Описание уже построенной модели ВК

Введем следующие определения из работы [2]. Число вычислительных машин, на ко-

торых выполняется задание, называется его шириной. Будем считать, что ширина задания определяется в момент его создания, что является довольно частым допущением [4—8] (немасштабируемые задания).

Длиной задания назовем время его выполнения на указанном в момент создания числе узлов (то есть на ширине). Вопрос изменения ширины не поднимается. Площадь задания назовем произведение длины на ширину. Очевидно, что площадь — это сложность задания. Она же представляет собой суммарное машинное время обслуживания. Этот параметр необходим для упрощения аппроксимации, так как многие математические модели в этом случае дают результат лучше. Далее эта особенность используется. Отметим, что авторы существующих публикаций по теме данной работы часто используют иную терминологию.

Реальный ВК построен из вычислительных машин, которые обслуживают поступающие задания [2, 3]. В базовой модели используется представление такого ВК в виде обслуживающего блока с единой неприоритетной, не ограниченной по размеру очередью. Это обеспечивает обслуживание всех входящих зада-

ний. При завершении исполнения очередного задания очередь просматривается вся от начала и до конца для выбора (извлечения) заданий на выполнение. При этом извлекаемых из нее заданий может быть несколько — ведь ушедшее из очереди (т.е. исполненное) задание может освободить не только один, но и несколько каналов (узлов) обслуживания. Первое обнаруженное задание, для выполнения которого имеется достаточное число свободных машин, ставится на исполнение. Затем просмотр очереди продолжается со следующего за ним задания в поисках второго. Пример распределения заданий по узлам ВК дан на рис. 1.

На нем представлено параллельное исполнение заданий с ширинами 1, 2, 5 и 8. При этом каждое из заданий может занимать произвольные узлы. Даже при наличии заданий в очереди часть узлов может простаивать. В нашем примере в очереди могут быть зада-

Узел 0 (задание 8, часть 1 из 5)	Узел 1 (свободен)	Узел 2 (задание 8, часть 3 из 5)
Узел 3 (задание 5, часть 1 из 1)	Узел 4 (задание 11, часть 1 из 2)	Узел 5 (задание 13, часть 4 из 8)
Узел 6 (свободен)	Узел 7 (задание 8, часть 2 из 5)	Узел 8 (свободен)
Узел 9 (задание 13, часть 7 из 8)	Узел 10 (задание 13, часть 1 из 8)	Узел 11 (задание 13, часть 5 из 8)
Узел 12 (задание 8, часть 4 из 5)	Узел 13 (свободен)	Узел 14 (задание 11, часть 2 из 2)
Узел 15 (задание 13, часть 3 из 8)	Узел 16 (задание 13, часть 2 из 8)	Узел 17 (задание 8, часть 5 из 5)
Узел 18 (задание 13, часть 6 из 8)	Узел 19 (свободен)	Узел 20 (задание 13, часть 8 из 8)

Рис. 1. Пример распределения заданий по узлам ВК

ния шириной шесть и более, которым на рассматриваемый момент времени, показанный на рисунке, нет места на исполнение, т.е. нет достаточного числа узлов ВК. Более узкое задание, которому такого места хватает, должно быть извлечено из очереди, несмотря на то что оно пришло позднее стоящих впереди него. Поскольку порядок обслуживания заданий может не соответствовать порядку их прихода (поступления в очередь), то номера заданий на рис. 1 идут не подряд.

Таким образом, среди всех заданий, которые можно поставить на исполнение, приоритет принадлежит пришедшим ранее.

Введем показатели очереди на основе показателей содержащихся в ней заданий. Длиной очереди назовем число находящихся в ней заданий, шириной (сложностью, площадью) очереди — сумму ширин (сложностей, площадей) входящих в нее заданий. Аналогично определим понятия длины, ширины и площади всей вычислительной системы, т. е. ВК.

Также мы столкнулись с одной особенностью. Существуют два средних времени ожидания в очереди. Ввиду того что в очередь попадают не все задания, а лишь их определенный процент, среднее время ожидания можно рассчитывать двумя способами:

1) для всех заданий, учитывая нулевое время ожидания, не попавших в очередь (будем обозначать ее без штриха);

2) только для попавших в очередь (со штрихом).

2. Постановка задачи

В данной работе ставится цель — продемонстрировать построенную модель ВК и основы ее программной реализации на примере предсказания работы кластера UniLu-GAIA при изменении, например, ширины кластера или производительности его узлов. Этот кластер выбран потому, что логи (журналы) его работы есть в открытом доступе [8] и работа может быть повторена (хотя природа исполняемых заданий конфиденциальна и неизвестна). Логи — списки выполненных кластером заданий, где присутствуют как минимум время прихода, длина и ширина заданий. Будет использован лог UniLu-Gaia-2014-2.swf, который принадлежит кластеру шириной 2004 на момент фиксации лога [9].

3. Описание стохастических моделей

В целях стохастического моделирования было усовершенствовано средство, описанное в работах [10, 11]. В качестве допустимой погрешности имитационного моделирования было взято значение 5 % как рекомендуемое во многих исследованиях [12]. В соответствии с Центральной предельной теоремой [12] это требует более 40 испытаний на проверку каждого из вариантов.

Сама процедура моделирования уже реализована нами в названной выше детерминированной модели [10, 11]. Стохастическая модель может использовать уже созданную модель, если заменить детерминированные данные нагрузки стохастическими. Это позволит нам рассмотреть различные варианты прихода заданий, а также заменить большой список входящих заданий простой математической моделью их генерации — моделью входящей нагрузки (МВН).

Каждая модель нагрузки состоит из двух частей: модели времен приходов заданий и модели обслуживания (определение длины, ширины и площади). Обсуждение этих моделей кратко дается в работах [13, 14].

Начнем с самого простого варианта, когда возможны аппроксимация времени прихода заданий A и аппроксимации времени обслуживания заданий B , B , где A — аппроксимация некоторым законом распределения интервала между приходами заданий, B — аппроксимация законом распределения площади (при указании " \wedge " — длины) заданий. Ширина моделируется отдельно по вероятности ее появления. Зная ширину и длину (площадь), мы можем определить и площадь (длину).

Вместо A и B будут использованы аппроксимации конкретными законами распределения.

Задания различной ширины могут иметь весьма различные характеристики распределения длины/площади. Поэтому имеет смысл выделить отдельные законы для интервалов ширины [4, 7]. В самом простом случае выделяем для каждой ширины свое распределение длины/площади задания. Таким образом, получаем отдельный поток заданий для каждой из возможных ширин. Обозначим этот вариант значком "\$" перед обозначением закона распределения площади: $\$B$.

В силу того что в каждом из таких входных потоков будет только одна ширина, длина бу-

дет пропорциональна площади и в отдельной аппроксимации не нуждается.

Другой вариант разделения — выделить в отдельную группу каждую ширину, равную степени двойки. Это имеет смысл, так как согласно [5] в логах доминируют именно задания, ширина которых является степенью двойки, — даже тогда, когда к этому нет технических предпосылок. Такие работы, как [6], выделяют еще и другие доминирующие (но слабее, чем степени двойки) ширины заданий, например кратные десяти. В других работах [15], наоборот, пытаются уйти от этой тенденции. Интервалы ширин между степенями двойки выделим также в отдельные группы: по одной группе на каждый интервал. То есть будут группы 1, 2, 3, 4, 5—7, 8, 9—15, 16, 17—31, 32, 33—63, 64 и т.д. Обозначим такое выделение групп знаком "&": $\&B$ и $\&B$.

Практика (как наша, так и [4, 7]) показывает, что при разделении входного потока на несколько потоков по ширине часто можно повысить качество аппроксимации нагрузки; получаемая МВН лучше описывает нагрузку.

Аналогичное разделение можно провести и для входных времен заданий. Выделим несколько входных потоков, в каждый из которых приходят задания с ширинами, относящимися к определенному интервалу. Используем аналогичные принципы разбиения и обозначения. Получатся $\&A$, $\&A$.

В работе [5] предлагается анализировать входной поток как нестационарный. Необходимость этого была подтверждена в нашей работе [16]. В данной работе будет рассмотрено изменение интенсивности потока заявок в течение недели. За начало недели мы взяли полночь с воскресенья на понедельник. Будем полагать, что интенсивность прихода заданий остается постоянной в течение получаса (как в работе [5]). Это предположение вводится исключительно для упрощения вычисления интегралов, при получении самих формул его не задействуют.

Под приведенной интенсивностью подразумевается

$$\overline{\lambda(t)} = \lambda(t)\overline{\lambda}, \quad (1)$$

где $\overline{\lambda(t)}$ — приведенная интенсивность прихода заданий; $\lambda(t)$ — интенсивность; $\overline{\lambda}$ — средняя интенсивность в течение недели.

Для генерации интервала между событиями нестационарного потока исказим временную

шкалу. Интервалом времени между реальными точками t_0 и t_1 временной шкалы будем считать значение интеграла

$$L(t, t_0) = \int_{t_0}^t \overline{\lambda(t)} dt. \quad (2)$$

Назовем его "случайным приведенным временем" между приходами заданий. Такой поток будет стационарным, и его можно аппроксимировать обычным способом, а потом вернуться к изначальной шкале времени.

Обозначим такую модель знаком "~" перед обозначением входного потока, например, $\sim A$. Здесь сразу же надо оговорить одну важную особенность. Обозначения $\sim A$ и $\$A$ не одинаковы. В первом случае подразумеваем введение единой интенсивности для всех входных потоков, а во втором — что каждый поток получает свою собственную интенсивность.

Таким образом, получаем следующие варианты приходов: $A, \sim A, \$A, \&A, \sim \$A, \sim \&A, \$\sim A, \&\sim A$ и варианты обслуживания: $B, \$B, \&B, B, \&B$. Сочетание этих двух моделей дает модель нагрузки. Будем обозначать его (сочетание) через знак слеша, например $\&\sim A/\&B$. Итого здесь имеем 40 комбинаций приходов заданий и параметров их обслуживания.

4. Законы распределений

Введем обозначения: $E(X)$ — математическое ожидание величины X , $VAR(X)$ — ее дисперсия, $stDev(X)$ — среднее квадратичное отклонение, $cov(X) = \frac{stDev(X)}{E(X)}$ — коэффициент вариации. Если мы имеем дело с оценкой момента, то будем писать над ней горизонтальную черту.

Методами аппроксимации являются метод моментов (ММ) и метод наибольшего правдоподобия (МНП). Вариант ММ предполагает определение параметров распределения по моментам, т. е. число оценок моментов должно равняться числу параметров распределения. Вариант МНП подразумевает максимизацию функции правдоподобия, которая часто выполняется численно.

В качестве законов распределения случайных после уточнения выводов [13, 14] величин были взяты следующие (в различных источниках встречаются разные обозначения, поэтому приводим используемые нами обозначения).

1. M — экспоненциальное распределение:

$$pdf(x) = \lambda e^{-\lambda x}; \quad cdf(x) = 1 - e^{-\lambda x}; \\ E(X) = stDev(X) = \frac{1}{\lambda}.$$

Для этого распределения оценки ММ и МНП совпадают и дают

$$\lambda = \frac{1}{E(X)}.$$

Решение для МНП возможно аналитически.

2. Γ — гамма-распределение:

$$pdf(x) = \lambda \frac{(\lambda x)^{v-1}}{\Gamma(v)} e^{-\lambda x}; \\ cdf(x) = \frac{\gamma(v, \lambda x)}{\Gamma(v)} = P(v, \lambda x); \\ E(X) = \frac{v}{\lambda}; \quad VAR(X) = \frac{v}{\lambda^2},$$

где $\Gamma(x)$ — гамма-функция Эйлера; $\gamma(x, y)$ — неполная нижняя гамма-функция; $P(x, y)$ — нормированная нижняя гамма-функция.

Оценки ММ (Γ_μ) имеют вид

$$\lambda = \frac{\overline{E(X)}}{VAR(X)}; \quad v = \lambda \overline{E(X)}.$$

Оценки МНП (Γ_λ) найти труднее. Из равенства частных производных логарифмической функции правдоподобия нулю удалось получить

$$\lambda = \frac{v}{E(X)}, \quad \Psi(v) = \overline{E(\ln X)} - \ln(\overline{E(X)}) + \ln v,$$

где $\Psi(v)$ — дигамма-функция; $\overline{E(\ln X)}$ — оценка среднего значения логарифма случайной величины.

Стоит отметить, что обе оценки дают одно и то же математическое ожидание, которое совпадает с оценкой ($E(X) = \overline{E(X)}$), но остальные моменты будут различаться в общем случае.

Все указанные выше распределения будем считать простыми в противоположность гиперраспределениям, которые следуют далее.

3. $H(n)$ — гиперэкспоненциальное распределение:

$$pdf(x) = \sum_{i=1}^n \alpha_i \lambda_i e^{-\lambda_i x}; \quad cdf(x) = 1 - \sum_{i=1}^n \alpha_i e^{-\lambda_i x}; \\ 1 \geq \alpha_i \geq 0; \quad \sum_{i=1}^n \alpha_i = 1; \quad cov(X) \geq 1,$$

где n — число ветвей, веток распределения (задается перед аппроксимацией как часть вида распределения).

В нашей работе [17] было введено упрощение этого распределения для ММ в двуветочном случае ($n = 2$). Обозначим его H_{μ} . МНП с численным решением применим в любом случае, поэтому обозначим этот вариант как $HГ(2)$, где n — число ветвей, веток распределения.

4. $HГ(n)$ — гипергамма-распределение:

$$pdf(x) = \sum_{i=1}^n \alpha_i \lambda_i \frac{(\lambda_i x)^{v_i-1}}{\Gamma(v_i)} e^{-\lambda_i x};$$

$$cdf(x) = \sum_{i=1}^n \alpha_i P(v_i, \lambda_i x); \quad \sum_{i=1}^n \alpha_i = 1,$$

$$1 \geq \alpha_i \geq 0, \quad cov(X) \in (0; \infty),$$

где n — число ветвей, веток распределения (задается перед аппроксимацией как часть вида распределения).

В нашей работе [18] были введены два упрощения для двуветочного случая: $HГ_{\mu}$ для ММ и $HГ_{\lambda}$ для МНП. МНП с численным решением применим в любом случае, поэтому обозначим этот вариант как $HГ(n)$, где n — число ветвей, веток распределения.

5. Расширяемость программной разработки

Представленная модель реализована в виде программной разработки [19], которая построена с использованием паттернов программирования, что позволяет быстро расширять написанную модель без серьезных изменений

программного кода. В качестве языка реализации выбран язык Java как сочетание производительности и удобства разработки. Основными паттернами для нас являются Адаптер, Команда, Стратегия, Фабрика [20].

На рис. 2 дана UML-диаграмма характеристик кластера. В нашем случае для характеристик используется класс NodeChar. Он содержит в себе два интерфейса.

Интерфейс TaskChooser отвечает за выбор задания из очереди, постановку задачи на очереди или исполнение. Таким образом, описанная нами процедура работы с очередью является лишь одной из реализаций этого интерфейса. Создав другую реализацию, можно реализовать свою собственную политику обработки очереди.

Второй интерфейс TaskStreamFactory отвечает за создание конкретного входящего потока заданий. Фабрика необходима, так как при стохастическом моделировании рассматривается большое число исходов и необходимо произвольное число потоков заданий. Сам поток представлен интерфейсом TaskStream. Использование интерфейсов позволяет при необходимости создать любой свой собственный поток входящих заданий, не ограничиваясь не только нашими МВН, но и подходом к генерации нагрузки в целом (например, она может поступать из готового файла без вмешательства программы).

На рис. 3 представлена UML-диаграмма TaskStreamFactory, а на рис. 4 — UML-диаграмма производимого фабрикой интерфейса TaskStream. В нашей реализации этой пары DefaultTaskStreamFactory производит Default

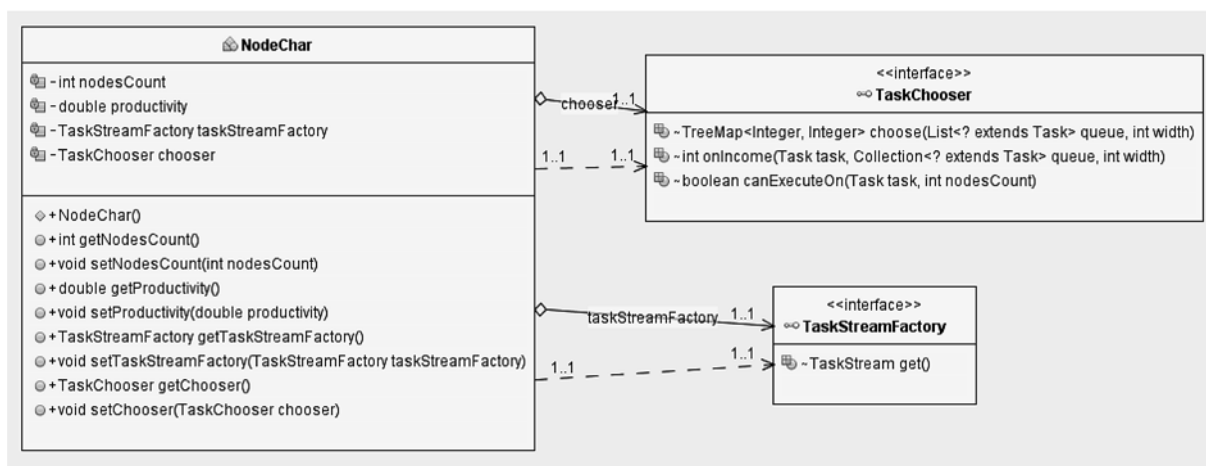


Рис. 2. UML-диаграмма характеристик узла (NodeChar)

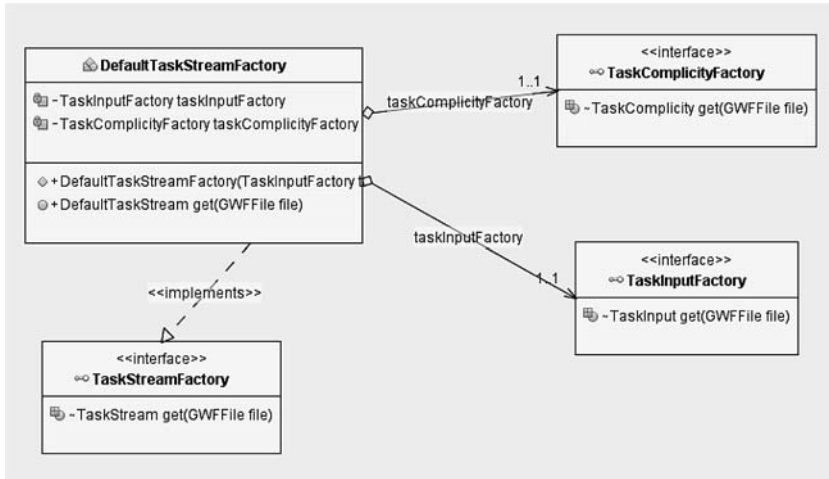


Рис. 3. UML-диаграмма TaskStreamFactory

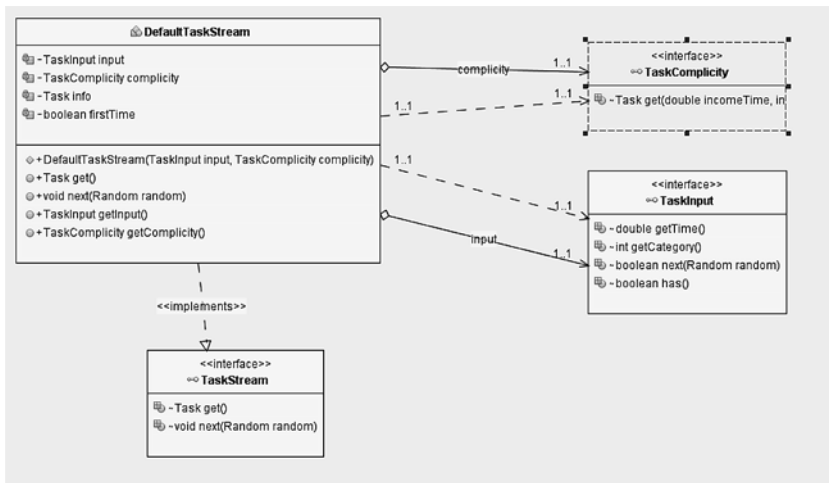


Рис. 4. UML-диаграмма TaskStream

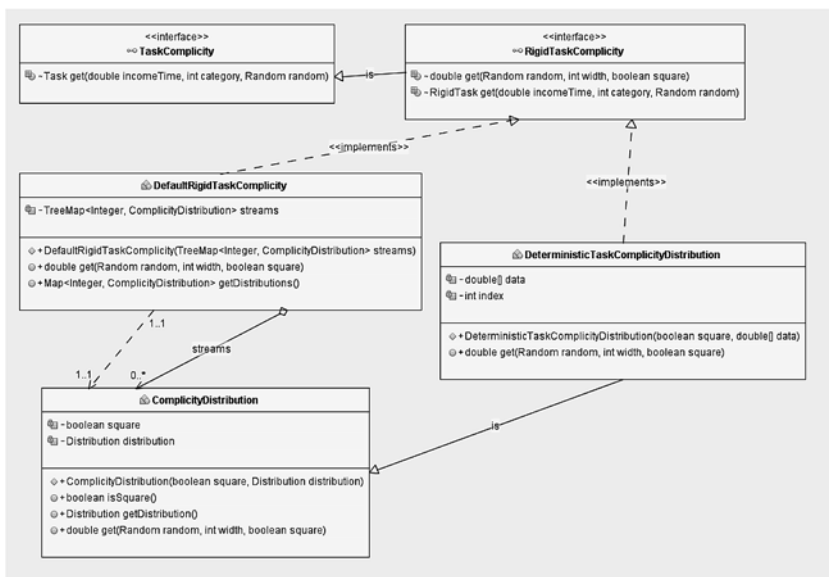


Рис. 5. UML-диаграмма интерфейса TaskComplicity

TaskStream: вся МВН делится на модели прихода заданий и модель обслуживания, и именно эти классы выполняют объединение.

Модель прихода соответствует паре интерфейсов TaskInputFactory и TaskInput, а модель обслуживания — паре TaskComplicityFactory и TaskComplicity. Написав свою реализацию для любой пары интерфейсов, можно воспроизвести свои модели прихода или обслуживания.

На рис. 5 показаны реализации TaskComplicity. В настоящий момент все реализации сводятся к реализации RigidTaskComplicity, который оперирует немасштабируемыми заданиями. При необходимости код программы может быть быстро расширен, например, интерфейсом MoldableTaskComplicity с реализациями для масштабируемых заданий по тем правилам масштабирования, которые нужны реализующему.

6. Моделирование

После проверки модели были найдены несколько наиболее подходящих МВН для кластера UniLu-GAIA [21]. В данной работе мы выберем модель $\lambda \sim N(2)/\Gamma \lambda^2$.

Сначала рассмотрим альтернативные вариации ширины кластера UniLu-GAIA. Максимальная ширина задания, которая встречается в логах данного кластера, — 516. Значит, минимальная ширина кластера для обслуживания этого лога — тоже 516. Поэтому вполне резонно должен быть задан вопрос, не является ли текущая ширина кластера 2004 избыточной или, наоборот, необходимо нарастить ширину кластера, потому что попавшие в очередь задания ждут там очень долго.

Используем моделирование, чтобы определить это. Дадим небольшой запас кластеру по ширине.

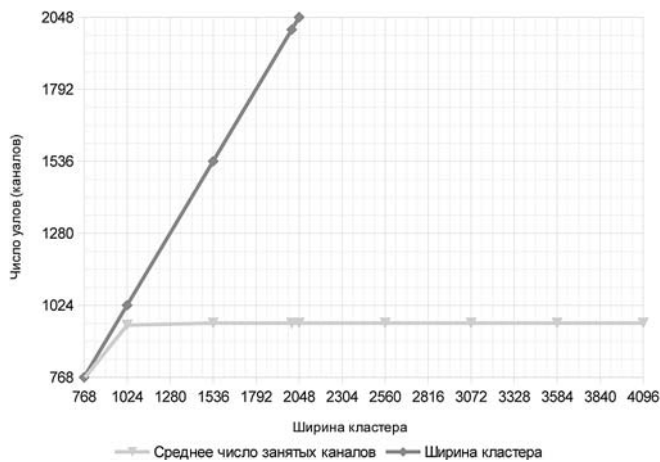


Рис. 6. Использование каналов кластера UniLu-GAIA при различной ширине кластера

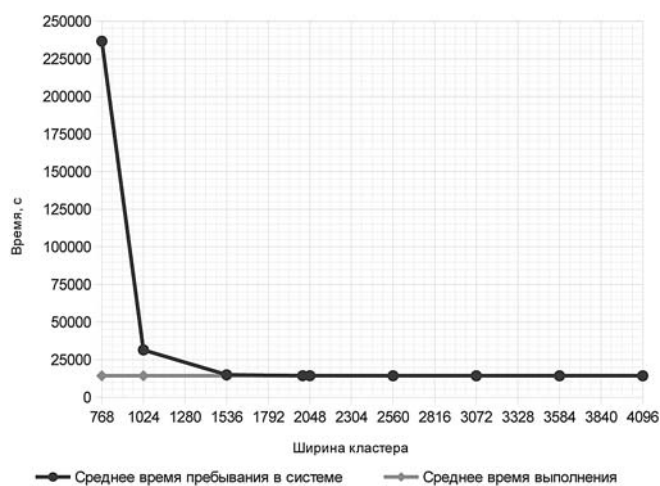


Рис. 7. Среднее время пребывания заданий в кластере UniLu-GAIA при различной ширине кластера

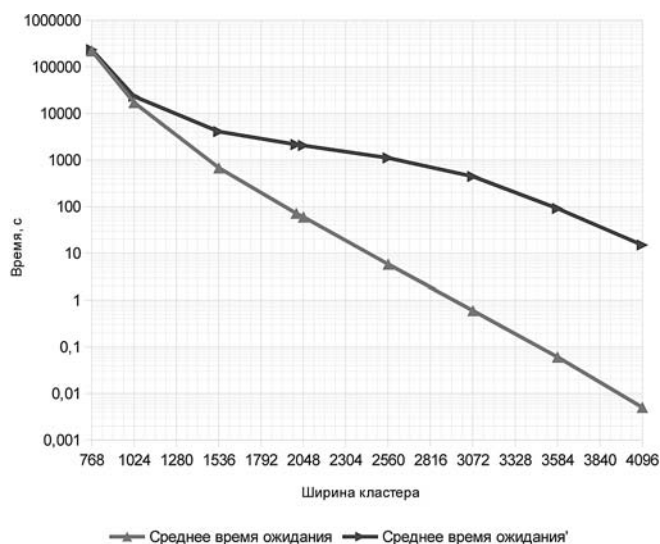


Рис. 8. Соотношение времен ожидания

Будем ориентировать на степени двойки, поэтому минимальной шириной будет 768. Также рассмотрим 1024, 1536, 2048, 2560, 3072, 3584, 4096.

На рис. 6 показано использование кластера при различной ширине.

Очевидно, что кластеру необходимы около 960 каналов, чтобы обслужить входящий поток заданий. В противном случае он просто не будет успевать обслуживать все задания. Это требование накладывается самим потоком входящих заданий.

Наиболее важным параметром для нас является среднее время пребывания в системе. Именно этот параметр говорит, сколько времени прошло с момента отправки задания на исполнение до получения результата. Соотношение времени ожидания в очереди и времени выполнения интересует нас лишь опосредованно.

Согласно рис. 7 при ширине кластера примерно до 960 наблюдается резкое увеличение времени пребывания в системе, так как кластер просто не справляется с таким потоком заданий.

Где-то при ширине примерно 1500 наблюдается практически совпадение графиков времени пребывания в системе и времени выполнения. Значит, среднее время ожидания всех заданий становится пренебрежимо малым по сравнению со временем обслуживания. Время выполнения составляет около 4...4,5 ч (изменить время выполнения изменением ширины кластера нельзя, так как задания не масштабируются).

Но даже при таком малом среднем значении времени ожидания всех заданий среднее время ожидания, взятое лишь для заданий, попавших в очередь, может быть сопоставимым со временем обслуживания. Иными словами, если задание попадает в очередь, оно попадает туда надолго. На рис. 8 представлены эти соотношения.

При ширине примерно в 1500 среднее время ожидания всех заданий составляет около 11 мин. При времени выполнения 4...4,5 ч им действительно можно пренебречь, но при этом среднее время ожидания заданий, попавших в очередь, составляет около 65 мин, что уже около четверти времени исполнения и отброшено быть не может. В такой ситуации в очередь попадает примерно каждое шестое задание (рис. 9). Значит, каждое шестое задание ждет около часа, а остальные пять исполняются сразу. Отсюда и среднее время ожидания всех заданий 11 мин. Поэтому и необходимы

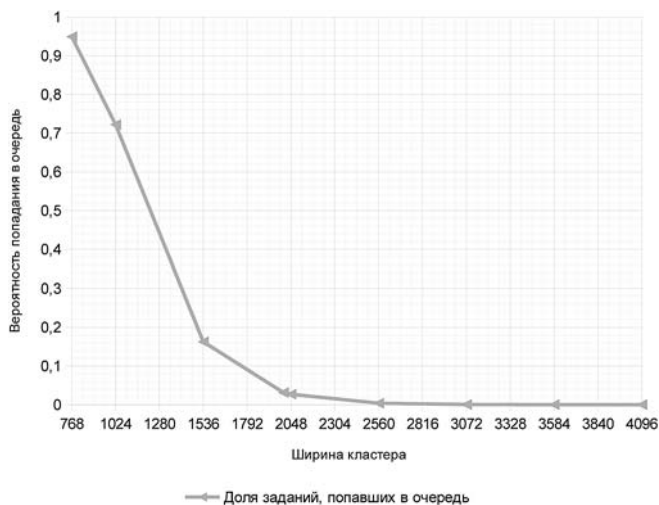


Рис. 9. Вероятность попадания в очередь

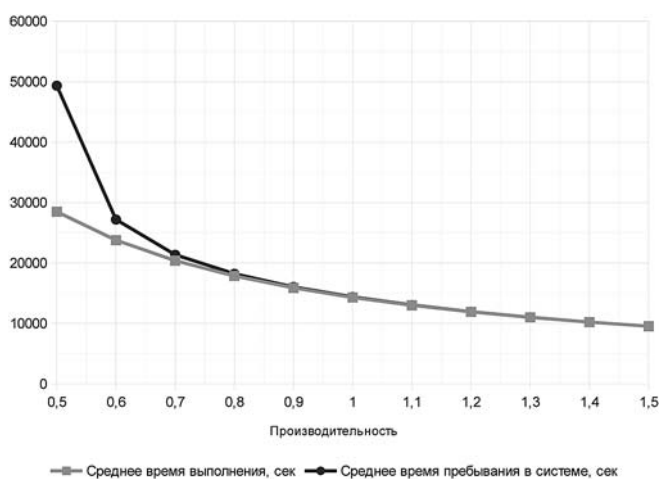


Рис. 10. Зависимость временных характеристик задания от производительности

два времени ожидания, чтобы оценить происходящее.

Увеличение ширины кластера до 2004 сокращает среднее время ожиданий попавших в очередь до 36 мин. При этом в очередь попадает примерно каждое 30-е задание, так что в среднем задания бывают в очереди около 1...1,3 мин. И администраторы кластера выбрали именно эту ширину.

Для сравнения наращивание ширины примерно до 3000 сократит среднее время ожиданий попавших в очередь до 7,5 мин, при этом в очередь попадет каждое 2000-е задание и среднее время ожидания всех заданий составит меньше секунды, но это на 50 % дороже по узлам. При ширине примерно в 4000 задания, попавшие в очередь, ждут там в среднем 15 с. Два других показателя слишком малы, чтобы их можно было учитывать.

Таким образом, наилучшей шириной кластера следует выбрать число, близкое к 2000. На ширинах кластера до 1000 кластер даже не способен справиться с потоком заданий, поэтому их нельзя рассматривать.

Следующий важный вопрос — это определение производительности узлов кластера. За единицу производительности возьмем текущую производительность кластера и отметим увеличение/уменьшение этой величины (на графике — доли единицы). Здесь делается допущение, что у всех заданий времена выполнения будут изменяться обратно пропорционально производительности. В общем случае такое допущение неверно, но без знания природы заданий невозможно отказаться от него.

Ответ на вопрос подходящей производительности также способен дать имитационное моделирование. В качестве примера будем снова рассматривать кластер UniLu-GAIA. На рис. 10 изображено изменение параметров обслуживания в зависимости от производительности. За единичную производительность взята текущая производительность кластера.

В данном случае уже нет такого явного порога изменения характеристик, который был при изменении ширины кластера. Но можно понять, что где-то при производительности, равной 0,7—0,8 от текущей и выше, время ожидания в очереди становится пренебрежимо малым по сравнению со временем обслуживания. Рекомендовать какую-то конкретную производительность здесь нельзя: все определяется финансовыми и техническими возможностями организации, содержащей кластер.

Заключение

Таким образом, было создано средство моделирования вычислительных кластерных систем, которое позволяет получить характеристики обслуживания и предсказать их для некоторой конфигурации [22]. Средство является расширяемым за счет использования паттернов программирования, в частности выделения необходимых интерфейсов.

В качестве примера было проведено моделирование работы кластера UniLu-GAIA, на котором были показаны конкретные характеристики обслуживания и приняты конкретные решения.

Список литературы

1. **Sinisterra M. M., Henaio T. M. D., López E. G. R.** Clúster de balanceo de carga y alta disponibilidad para servicios web y mail. URL: <https://dialnet.unirioja.es/descarga/articulo/4364562.pdf> (дата обращения 04.01.2018).
2. **Аветисян А. И., Гайсарян С. С., Грушин Д. А., Кузюрин Н. Н., Шокуров А. В.** Эвристики распределения задач для брокера ресурсов Grid. URL: <http://www.citforum.ru/nets/digest/grid/index.shtml> (дата обращения 04.01.2018).
3. **GridMe:** Grid modeling environment. Google code. URL: <https://code.google.com/p/gridme/> (дата обращения 04.01.2018).
4. **Jann J., Pattnaik P., Franke H., Wang F., Skovira J., Riordan J.** Modeling of Workload in MPPs // Job Scheduling Strategies for Parallel Processing: Lect. Notes Comput. Sci. Springer-Verlag. 1997. Vol. 1291. P. 95–116.
5. **Lublin U., Feitelson D. G.** The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Jobs // The Rachel and Selim Benin School of Computer Science and Engineering. URL: <http://www.cs.huji.ac.il/~feit/papers/Rigid01TR.pdf> (дата обращения 04.01.2018).
6. **The Feitelson 1996 Model** // The Rachel and Selim Benin School of Computer Science and Engineering. URL: http://www.cs.huji.ac.il/labs/parallel/workload/m_feitelson96/ (дата обращения 04.01.2018).
7. **The Jann et al 1997 Model** // The Rachel and Selim Benin School of Computer Science and Engineering. URL: http://www.cs.huji.ac.il/labs/parallel/workload/m_jann97/ (дата обращения 04.01.2018).
8. **Logs of Real Parallel Workloads from Production Systems** // The Rachel and Selim Benin School of Computer Science and Engineering. URL: <http://www.cs.huji.ac.il/labs/parallel/workload/logs.html> (дата обращения 04.01.2018).
9. **НРС @ Uni.lu.** URL: <https://hpc.uni.lu/systems/gaia/> (дата обращения 04.01.2018).
10. **Гаевой С. В., Аль-Хадша Ф. А. Х., Лукьянов В. С.** Детерминированная имитационная модель кластеров грид-системы, обслуживающих задания // Вестник компьютерных и информационных технологий. 2014. № 6. С. 39–43.
11. **Гаевой С. В., Аль-Хадша Ф. А. Х., Фоменков С. А., Лукьянов В. С.** Детерминированная имитационная модель кластеров грид-системы для сравнения эффективности использования эвристик распределения заданий // Прикаспийский журнал: управление и высокие технологии. 2014. № 2. С. 148–157.
12. **Фоменков С. А., Камаев В. А., Орлова Ю. А.** Математическое моделирование системных объектов: учеб. пособ. (гриф). Доп. УМО вузов по университетскому политехн. образованию. Волгоград, 2014. 335 с.
13. **Гаевой С. В., Аль-Хадша Ф. А. Х., Фоменков С. А.** Аппроксимация времени выполнения заданий на примере вычислительного кластера LPC EGEE 2004 // Известия ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". 2014. № 12 (139). С. 135–141.
14. **Гаевой С. В., Аль-Хадша Ф. А. Х.** Моделирование работы вычислительного кластера на примере LANL CM5 // SCI-ARTICLE.RU: электронный периодический научный журнал. 2013. № 3 (ноябрь). С. 304–313. URL: http://sci-article.ru/stat.php?i=modelirovanie_raboty_vychislitelnogo_klastera_na_primere_LANL_CM5 (дата обращения 04.01.2018).
15. **Downey A. B.** A Parallel Workload Model and Its Implications for Processor Allocation // The Rachel and Selim Benin School of Computer Science and Engineering. URL: <http://alldowney.com/research/allocation/> (дата обращения 04.01.2018).
16. **Ахмед В. М. А., Гаевой С. В., Фоменков С. А.** Влияние нестационарности входного потока заданий на обслуживание в кластерной системе // Вестник компьютерных и информационных технологий. 2017. № 11 (161). С. 44–52.
17. **Гаевой С. В., Ахмед В. М. А., Быков Д. В., Фоменков С. А.** Сокращение времени аппроксимации логов вычислительного кластера с использованием методов моментов на гиперэкспоненциальном распределении // Прикаспийский журнал: управление и высокие технологии. 2017. № 1. С. 94–105.
18. **Ахмед В. М. А., Гаевой С. В., Фоменков С. А.** Сокращение времени аппроксимации нагрузки вычислительного кластера с использованием упрощения гипер-гамма-распределения // Вестник Воронежского института высоких технологий. 2017. № 4 (23). С. 52–58.
19. **Гаевой С. В., Ахмед В. М. А., Фоменков С. А.** ВолгГТУ. Средство аппроксимации и имитационного моделирования вычислительных нагрузок (SWFJParser. JDSBroker). Свидетельство о гос. регистрации программы для ЭВМ № 2017619355 от 24 августа 2017 г. Российская Федерация. 2017.
20. **Freeman E., Bates V., Sierra K., Robson E.** Head First Design Patterns. A Brain-Friendly Guide. O'Reilly Media. 2009. 688 p.
21. **Гаевой С. В., Ахмед В. М. А., Быков Д. В., Фоменков С. А.** Аппроксимация потока заданий на примере вычислительного кластера UniLu-Gaia // Известия ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". 2017. № 8 (203). С. 96–102.
22. **Gaevoy S. V., Ahmed W. M. A., Fomenkov S. A., Kolesnikov S. G.** Methods of Modeling Incoming Jobs Stream used on the Computing Cluster UniLu-Gaia // Journal of Engineering and Applied Sciences. 2017. Vol. 12, Issue 24. P. 7548–7554. URL: <http://medwelljournals.com/abstract/?doi=jeasci.2017.7548.7554>. DOI: 10.3923/jeasci.2017.7548.7554

S. V. Gaevoy, Ph. D. (Engineering), Senior Lecturer, e-mail: gaevserge@mail.ru;

W. M. A. Ahmed, Post-Graduate Student, e-mail: wesamalsofi@gmail.com;

S. A. Fomenkov, D. Sc. (Engineering), Professor, Professor of "CAD" department, e-mail: saf@vstu.ru,
Volograd State Technical University, Volograd, 400005, Russian Federation

Using Simulation to Determine Quality of Service of a Cluster System

In this paper computing clusters (CC) are considered. They are used to execute incoming jobs. There is such a CC in our university and we need to predict its service characteristics at executing several workloads. An important method to analyze parallel workloads is modeling execution of those systems by using parallel workload models (PWM). A simulation model of cluster has already been created at our department. We have already proposed many PWMs, but all these PWMs use

a continuous variable approximation. This approximation can be done either by method of moments (MM), or maximum likelihood method (MLM). The latter gives the more accurate results but consumes much time. It was empirically proved in our and third-party papers. In this paper the goal is to demonstrate the already built CC model and the basics of its implementation. The implementation uses software design patterns and can adapted to many computing systems. We have chosen Java language because it has good performance and is convenient for the development. The program is cross-platform. Everyone can use, for example, Open JDK to compile it for free. The parallel workload of a cluster UniLu-GAIA is used because it is in free access and thus the computing can be repeated by everyone (although the nature of the tasks being executed there is confidential and unknown). The log being used belongs to the cluster with width 2004 at the moment of executing.

Keywords: parallel workload, parallel workload model, rigid job, job length, simulation, stochastic approximation, quality of service, computing cluster, simulation utility, workload approximation, continuous variable distribution, design pattern

DOI: 10.17587/it.24.464-474

References

1. **Sinisterra M. M.** [et al.]. Cluster de balanceo de carga y alta disponibilidad para servicios web y mail [Cluster of load balancing and high availability for web and mail services]. Available at: <https://dialnet.unirioja.es/descarga/articulo/4364562.pdf>
2. **Avetisyan A. I.** [et al.]. *Evrstiki raspredeleniya zadach dlya brokera resursov Grid* [Heuristics of job distribution for Grid resource broker]. Available at: <http://www.citforum.ru/nets/digest/grid/index.shtml> (in Russian).
3. **GridMe:** Grid modeling. Available at: <https://code.google.com/p/gridme/>
4. **Jann J.** [et al.]. Modeling of Workload in MPPs. Job Scheduling Strategies for Parallel Processing: Lect. Notes Comput. Sci. ed. by D. G. Feitelson, L. Rudolph. Springer-Verlag, 1997, vol. 1291. P. 95–116.
5. **Lublin U., Feitelson D. G.** The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Jobs. The Rachel and Selim Benin School of Computer Science and Engineering. Available at: <http://www.cs.huji.ac.il/~feit/papers/Rigid01TR.pdf>
6. **The Feitelson** 1996. The Rachel and Selim Benin School of Computer Science and Engineering. Available at: http://www.cs.huji.ac.il/labs/parallel/workload/m_feitelson96/
7. **The Jann et al** 1997 Model. The Rachel and Selim Benin School of Computer Science and Engineering. Available at: http://www.cs.huji.ac.il/labs/parallel/workload/m_jann97/
8. **Logs of Real Parallel Workloads from Production Systems.** The Rachel and Selim Benin School of Computer Science and Engineering. Available at: <http://www.cs.huji.ac.il/labs/parallel/workload/logs.html>
9. **HPC @ Uni.lu.** Available at: <https://hpc.uni.lu/systems/gaia/>
10. **Gaevoy S. V., Al-Hadsha F. A. H., Luk'yanov V. S.** *Determinirovannaya imitatsionnaya model klusterov grid-sistemy, obsluzhivayuschikh zadaniya* [Deterministic simulation model of clusters of a Grid-system executing jobs]. *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Herald of computer and information technologies], 2014, no. 6, pp. 39–43 (in Russian).
11. **Gaevoy S. V., Al-Hadsha F. A. H., Fomenkov S. A., Luk'yanov V. S.** *Determinirovannaya imitatsionnaya model' klusterov grid-sistemy dl'ya sravneniya effektivnosti ispol'sovaniya evristik raspredeleniya zadaniy* [Deterministic simulation model of clusters of a Grid-system for comparison of heuristics for task distribution]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2014, no. 2, pp. 148–157 (in Russian).
12. **Fomenkov S. A., Kamaev V. A., Orlova Yu. A.** *Matematicheskoye modelirovanie sistemnykh ob'yektov* [Mathematical modeling of system objects], Volgograd, 2014. 335 p. (in Russian).
13. **Gaevoy S. V., Al-Hadsha F. A. H., Fomenkov S. A.** *Approksimatsia vremeni vypolneniya zadaniy na primere vychislitel'nogo klustera LPC EGEE 2004* [Approximation of job execution time discovering computing cluster LPC EGEE 2004]. *Izvestiya VolgGTU, seriya "Aktualnye problemy upravleniya, vychislitel'noy tekhniki i informatiki v tekhnicheskikh sistemakh"* [Newspaper of VSTU, Series "Actual Problems of Management, Computing Hardware and Informatics in Engineering Systems"], Volgograd, 2014, no. 12 (139), pp. 135–141 (in Russian).
14. **Gaevoy S. V., Al-Hadsha F. A. H.** *Modelirovanie raboty vychislitel'nogo klustera na primere LANL CM5* [Simulation of computing cluster discovering LANL CM5]. *SCI-ARTICLE.RU: elektronnyy periodicheskiy nauchnyy zhurnal* [SCI-ARTICLE.RU: Electronic periodical scientific magazine], 2013, no. 3 (Nov), pp. 304–313. Available at: http://sci-article.ru/stat.php?i=modelirovanie_raboty_vychislitel'nogo_klustera_na_primere_LANL_CM5 (in Russian).
15. **Downey A. B.** *A Parallel Workload Model and Its Implications for Processor Allocation.* The Rachel and Selim Benin School of Computer Science and Engineering. Available at: <http://allendowney.com/research/allocation/>
16. **Ahmed W. M. A., Gaevoy S. V., Fomenkov S. A.** *Vliyanie nestacionarnosti vhodnogo potoka zadaniy na obsluzhivanie v klaster'noy sisteme* [Incoming Job Streams Nonstationary Nature Influence Streams on the Service in a Cluster System]. *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Herald of computer and information technologies], 2017, no. 11 (161), pp. 44–52 (in Russian).
17. **Gaevoy S. V., Ahmed W. M. A., Bykov D. V., Fomenkov S. A.** *Sokrashhenie vremeni approksimatsii logov vychislitel'nogo klustera s ispol'zovaniem metodov momentov na giperjeksponentsial'nom raspredelenii* [Reducing the approximation time of cluster workload by using method of moments on Hyperexponential distribution]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2017, no. 1, pp. 94–105 (in Russian).
18. **Ahmed W. M. A., Gaevoy S. V., Fomenkov S. A.** *Sokrashhenie vremeni approksimatsii nagruzki vychislitel'nogo klustera s ispol'zovaniem uproshheniya giper-gamma-raspredeleniya* [Reducing the Approximation Time of Cluster Workload by Using Simplified Hypergamma Distribution]. *Vestnik Voronezhskogo instituta vysokikh tekhnologiy* [Herald of Voronezh Institute of High Technologies], 2017, no. 4 (23), pp. 52–58 (in Russian).
19. **Gaevoy S. V., Ahmed W. M. A., Fomenkov S. A.** *VSTU. Svid. o gos. registratsii programmy dl'ya JeVM № 2017619355 ot 24 avgusta 2017 g. Rossijskaja Federatsiya. Sredstvo approksimatsii i imitatsionnogo modelirovaniya vychislitel'nykh nagruzok (SWFJParser. JDSBroker)* [Certificate of state registration of the computer program no. 2017619355. August 24, 2017. Russian Federation. Utility for approximation and simulation of parallel workloads (SWFJParser.JDSBroker)] (in Russian).
20. **B. Bates** etc. *Head First Design Patterns. A Brain-Friendly Guide.* O'Reilly Media, 2009, 688 p.
21. **Gaevoy S. V., Ahmed W. M. A., Bykov D. V., Fomenkov S. A.** *Approksimatsiya potoka zadaniy na primere vychislitel'nogo klustera UniLu-Gaia* [The Approximation of Task Stream by Using the Computing Cluster UniLu-Gaia]. *Izvestiya VolgGTU, seriya "Aktualnye problemy upravleniya, vychislitel'noy tekhniki i informatiki v tekhnicheskikh sistemakh"* [Newspaper of VSTU, Series "Actual Problems of Management, Computing Hardware and Informatics in Engineering Systems"], Volgograd, 2017, no. 8 (203), pp. 96–102 (in Russian).

О. С. Исаева, канд. техн. наук, ст. науч. сотр., e-mail: isaeva@icm.krasn.ru,
Институт вычислительного моделирования Сибирского отделения Российской академии наук —
обособленное подразделение ФИЦ КНЦ СО РАН, Красноярск

Технология построения комплексных моделей в инфраструктуре имитационного моделирования

Предложена технология построения комплексных имитационных моделей, позволяющая объединить модели, построенные по стандарту SMP2 (Simulation model portability), виртуальные приборы и логические модели, методы функционирования которых задаются в базах знаний. Технология основана на формализации моделей, содержащей структурно-параметрическое и функциональное представления. Она позволяет формировать семантические конструкции в терминах предметной области, что обеспечивает удобство построения моделей и наглядность представления моделируемой системы в целом. Технология апробирована в программном обеспечении, предназначенном для имитационного моделирования спутниковых систем.

Ключевые слова: космический аппарат, бортовая аппаратура, имитационное моделирование, стандарт Simulation model portability (SMP2), инфраструктура имитационного моделирования

Введение

Разработка сложных проектов в аэрокосмической отрасли предполагает обеспечение высокого уровня взаимодействия различных производителей. В случае европейских проектов кооперация основывается на стандартах Европейского космического агентства, которые задают требования к нормативным документам, информационным базам, программным компонентам для всех этапов жизненного цикла производства космической техники [1]. Стандарты определяют существенную роль моделирования при реализации космических проектов. Принципы имитационного моделирования технических объектов, состоящих из подсистем, изготавливаемых разными производителями, заложены в стандарте "Simulation Model Portability" (текущая редакция SMP2) [2]. Системы моделирования, реализующие стандарт SMP2, называют инфраструктурами имитационного моделирования (Simulation Infrastructure). Существует целый ряд инфраструктур моделирования, в их числе: SimSAT — Европейского космического агентства [3], SimTG — Astrium

Satellites [4], симулятор центра управления полетами SWARMSIM [5] и др.

Интеграция моделей и технологий, построенных на основе SMP2, выполняется в различных научно-исследовательских проектах. Примером успешного применения SMP2 является франко-германский проект разработки спутников дистанционного зондирования для мониторинга парникового газа — MERLIN [6]. Проект объединяет несколько компаний, в их числе: Airbus, CNES и Thales Alenia Space. Для реализации проекта выполнена интеграция имитационных моделей, разрабатываемых исследователями на четырех различных SMP2 платформах имитационного моделирования. Применение в данном проекте SMP2 позволило выполнить стандартизацию всех компонентов пространственной системы, сделать их более надежными, снизить затраты и оптимизировать график разработки. Положительный опыт подобных интеграций показывает актуальность и обоснованность ориентации программного обеспечения имитационного моделирования на стандарт SMP2.

В отличие от существующих решений, основанных на стандарте SMP2, для имитаци-

онного моделирования функционирования бортовой аппаратуры для предприятия-разработчика спутниковых систем потребовалось объединить в комплексной модели не только SMP2-модели различных производителей, но и логические модели, в основе которых лежат базы правил, а также виртуальные приборы, моделирующие физические свойства бортовой аппаратуры. Применение в комплексной модели логических подсистем и объединение их с другими моделями позволит обеспечить существенные преимущества для конструкторов бортовой аппаратуры на этапе проектирования и испытания бортовой аппаратуры.

Целью данной работы является создание технологии интеграции разнородных имитационных моделей, построенных на основе стандарта SMP, баз знаний и виртуальных приборов. Технология предназначена для построения комплексных имитационных моделей функционирования бортовых систем космического аппарата и может применяться для поддержки задач конструирования, подготовки и проведения испытаний и анализа результатов. В работе решены следующие задачи: определены функциональные задачи для построения комплексной модели бортового оборудования космических систем; предложено формальное представление комплексной имитационной модели; разработан алгоритм интеграции разнородных имитационных моделей; построены примеры комплексных моделей функционирования бортовой аппаратуры.

Реализация предложенных моделей и методов выполнена специалистами Института вычислительного моделирования СО РАН [7].

1. Задачи построения комплексной модели

Модели функционирования космических систем предназначены для имитации взаимодействия различных бортовых устройств, моделирования выполнения ими команд управления, формирования и анализа телеметрической информации и др. Способы реализации имитационных моделей и степень их детализации определяются спецификой решаемых задач. Часть моделей может быть представлена схематично, только на уровне логики функционирования, а другая часть может быть реализована более детально с учетом физических характеристик, свойств передаваемых сигналов, линий связи и пр. В качестве примера рассмотрим задачу моделирования внешнего командно-программного управления бортовыми системами. Такая модель применяется в наземных испытаниях бортовой аппаратуры. Элементы модели (рис. 1): БУ БКУ — бортовой комплекс управления; БЦВК — бортовой цифровой вычислительный комплекс; МИ КИС — интерфейсный модуль командно-измерительной системы космического аппарата; НКУ — наземный комплекс управления; КУ — команда управления, типы коммутационных интерфейсов: РЕЛ — релейный, 232 — RS-232, ИМП — импульсный, 422 — RS-422, ВЧ — высокочастотный интерфейс.

Имитатор наземного комплекса управления выполняет передачу команды в имитатор бортовой аппаратуры командно-измерительной системы космического аппарата, который анализирует полученную команду, при необходимости ее выполняет и формирует квитанцию

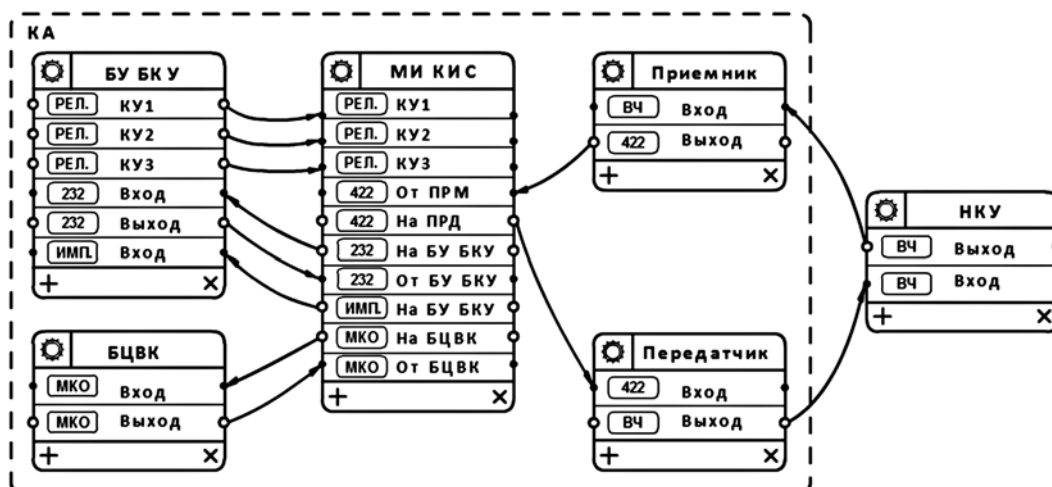


Рис. 1. Модель командно-программного управления бортовой аппаратурой

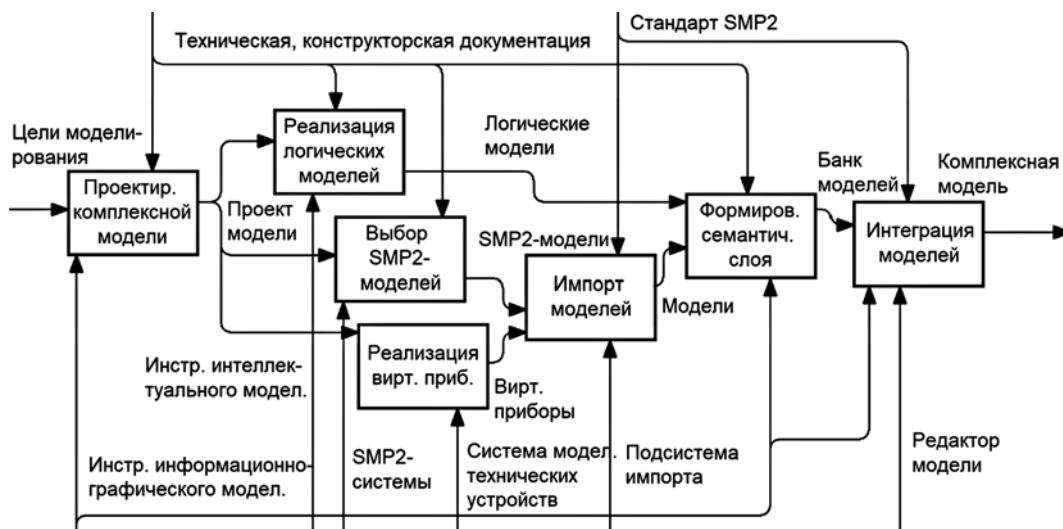


Рис. 2. Функциональные задачи построения комплексной модели

в телеметрии. Полученная бортовая команда передается в программный имитатор бортового комплекса управления, где она выполняется и по результатам формируется ответ в телеметрии. Телеметрия из бортовой аппаратуры телесигнализации передается в имитатор наземного комплекса управления, где выполняется ее анализ. На его основе делается вывод о прохождении и обработке команд, а также о состоянии бортовых систем.

Реализация элементов модели может быть выполнена в виде логических имитаторов бортовых систем и виртуальных проборов приемо-передающих устройств. Логические модели создаются на основе правил (condition-action rules) и описывают логику поведения оборудования для моделирования отработки команд и анализа их прохождения. Инструменты построения логических моделей описаны в работах [8, 9]. Виртуальные приборы создаются в среде Labview (Laboratory Virtual Instrumentation Engineering Workbench) [10] и предназначены для исследования приемо-передающего тракта моделируемой системы, оценки ошибок и потерь на тракте при различных заданных характеристиках устройств и внешних воздействиях.

Для построения комплексной модели требуется выполнить ее проектирование, реализацию отдельных моделей, импорт, формирование семантического слоя в терминах предметной области и интеграцию (рис. 2).

Проектирование комплексной модели выполняется на основе анализа целей и задач моделирования и включает графическое постро-

ение структуры модели, определение свойств ее элементов и интерфейсов информационного взаимодействия. Для каждой модели создаются семантические конструкции, позволяющие оперировать функциями и свойствами моделей в терминах предметной области. Интеграция моделей объединяет различные подмодели в комплексное решение. В основе технологии построения комплексной модели лежит унифицированное формальное представление, состоящее из структурно-параметрического и функционального описаний.

2. Формальное представление комплексной модели

Комплексная модель M представляется в виде: $M = \langle S, F, T \rangle$, где S — структурно-параметрическое описание, F — функциональное описание, T — множество элементов управления временем моделирования.

Структурно-параметрическое описание комплексной модели представлено в виде: $S = \langle B, C, D \rangle$, где B — элементы модели, описывающие свойства или функции отдельных физических устройств, C — типизированные информационные зависимости, описывающие коммутационные соединения между устройствами, D — множество структур данных, содержит стандартные типы данных и дополнительные конструкции — пакеты телекоманд и телеметрии. Структурно-параметрическое описание формирует семантический уровень модели, позволяя интерпретировать

понятия предметной области в графические и программные структуры. Элементы имитационной модели $B_i \in B$ представляются в виде кортежей $B_i = \langle N_i, K_i, I_i, P_i \rangle$, где N_i — наименование i -го элемента модели; K_i — тип элемента; P_i — множество параметров; I_i — множество коммутационных интерфейсов: $I_i = \{I_{i1}, \dots, I_{in}\}$, где n — число точек входов и выходов B_i . Каждый интерфейс I_{in} множества I_i имеет характеристики: тип интерфейса $Tr(I_{in}) \subseteq \{RS-232, RS-422, SpaceWire, \text{импульсный}, \text{релейный}, \text{вч}, \text{питание}\}$; направленность передачи $Rt(I_{in}) \subseteq \{\text{Вход}, \text{Выход}\}$; признак состояния $Onf(I_{in}) \subseteq \{\text{Включен}, \text{Выключен}\}$. Введение характеристик коммутационных интерфейсов позволяет выполнять типизацию информационных зависимостей между элементами комплексной модели и применять единые правила обработки данных, передаваемых по однотипным интерфейсам. Коммутационные соединения элементов моделей задаются в множестве C типизированных информационных зависимостей. Элемент $C_{ij} \in C$ описывает свойства коммутационного соединения пары моделей B_i и B_j : $C_{ij} = \langle I_{in}, I_{jm}, \tau_{ij} \rangle$, где I_{in} — интерфейс B_i , I_{jm} — интерфейс B_j , τ_{ij} — время прохождения сигнала между интерфейсами B_i и B_j .

Функциональное описание F представляет различные реализации элементов моделируемых устройств. Функциональное описание задается в виде: $F = \{R, CPP, VI\}$, где R — множество правил базы знаний, CPP — программные библиотеки SMP2-моделей и VI — виртуальные приборы. $R = \{A \rightarrow Z\}$ — правила, описывающие логику работы модели; A — условие выполнения правила; Z — действия, изменяющие состояние модели, значения параметров или таймеров. Для формирования правил используются множества, заданные в структурно-параметрическом описании.

Множество элементов управления временем T предназначено для задания точек перехода между событиями модели: $T = \{Tt, Tf\}$, где Tt — таймеры, Tf — методы работы со службой единого времени. Способы реализации таймеров Tt определяются в каждой модели в зависимости от ее назначения. В моделях, реализованных в виде правил, таймеры задают время начала выполнения правила, длительность ожидания, время его повторения или окончания. Таймеры, как и другие виды переменных, участвуют в формировании правил. Для моделей, реализованных в виртуальных при-

борах, элементы времени предназначены для имитации задержек преобразования или передачи данных. Синхронизация таймеров выполняется через единую службу времени *Time Keeper*. Описание службы времени *Time Keeper* приведено в стандарте SMP2 [2]. Множество Tf определяет методы работы со службой синхронизации времени: "Get_Time" и "Set_Time". *Time Keeper* использует два основных типа времени: *Mission Time* (определяет время моделей, выполняющих одну задачу моделирования) и *Simulation Time* (время моделирования, которое прогрессирует только в имитационном окружении в состоянии выполнения). Для управления временем *Time Keeper* отсчитывает время, единое для всех моделей, и при его изменении создает события, на которые подписываются все модели. В некоторых случаях модель может сама установить значение времени в зависимости от того, сколько ей требуется для выполнения моделирования. В этом случае *Time Keeper* выполняет синхронизацию времени между остальными процессами. Если при достижении модельного времени не происходит планового окончания выполнения отдельного процесса, то время моделирования останавливается до его завершения.

Time Keeper поддерживает несколько режимов моделирования: *Real-Time* (реального времени), *Accelerated* (ускоренный), *Free Running* (режим выполнения так быстро, как это возможно), *Debugging* (пошаговый). Для повышения наглядности процесса моделирования используется режим *Accelerated*, который позволяет быстрые и медленные события размещать на одной временной шкале и выполнять их с удобной для восприятия скоростью. Шаг времени в этом случае рассчитывается с использованием коэффициента ускорения k , т. е. $t_i = t_{i-1} + k\Delta t$. Если $k < 1$, то выполнение модели ускоряется, если $k > 1$, то замедляется (рис. 3).

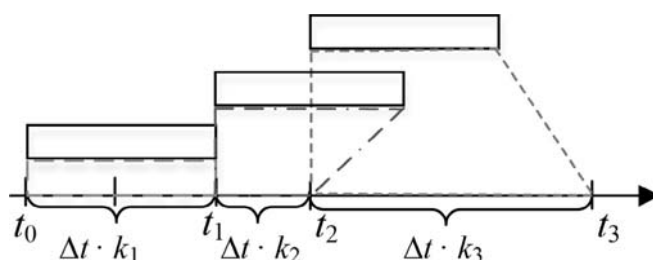


Рис. 3. Режим моделирования *Accelerated*, $k_1 = 1$, $k_2 < 1$, $k_3 > 1$

Не существует универсального подхода к организации взаимодействия моделей, для которых существенным является время протекания отдельных процессов. *Time Keeper* не всегда может решить задачу их синхронизации. Подготовка к моделированию в этом случае требует выявления критических циклов моделирования, возникающих из-за разницы во времени выполнения отдельных моделей. Для исключения ситуаций, когда одна модель может потребовать параметры, которые еще не были обновлены на предыдущем временном шаге, требуется уделить дополнительное внимание построению плана выполнения моделей.

3. Построение комплексных моделей

Реализация технологии построения комплексных моделей выполнена в системе имитационного моделирования, разработанной в ИВМ СО РАН [11]. Разработан алгоритм подготовки и интеграции разнородных имитационных моделей (рис. 4). Модели, вне зависимости от способа их реализации, импортируются в имитационную систему, в состав которой входит программный конструктор с инструментами информационно-графического моделирования. На этапе импорта моделей программное обеспечение автоматически определяет все элементы, которые может построить из представления модели.

При импорте автоматически строится множество S — структурно-параметрическое описание модели. В него входят B — блоки модели (элементы), параметры и таймеры. Инструмен-

ты информационно-графического моделирования на основе структурно-параметрического описания строят графические представления. Для каждого блока модели из импортируемых данных выделяют методы работы модели и формируют множества F функциональных представлений.

Далее формируется семантический слой модели. Он основан на формальном описании и позволяет привести различные реализации моделей к унифицированному виду, что, в свою очередь, обеспечивает возможность применения общих подходов к обработке и анализу однотипных элементов. Для моделей, реализованных в базе знаний $F = \{R\}$, семантический слой создан при построении модели и в имитационную среду загружается автоматически при импорте. Для остальных моделей $F = \{CPP, VI\}$ семантический слой формируется специалистом предметной области самостоятельно (полуавтоматически). Программное обеспечение анализирует множество S — структурно-параметрическое описание и формирует список входных и выходных параметров. Эти параметры загружаются в графическую модель формирования коммутационных интерфейсов I . Специалист выбирает в графическом редакторе характеристики интерфейсов: тип Tr , направленность передачи Rt , признак состояния Onf . После типизации интерфейсов программное обеспечение предоставляет дополнительные методы обработки данных по однотипным интерфейсам, которые не зависят от способа реализации модели. На рис. 5 (см. вторую сторону обложки) показан Конструктор модели, содержащий графические элементы импорта и формирования семантического слоя.

Если модель состоит из нескольких блоков, то специалист может сформировать множество C коммутационных соединений блоков моделей.

Логические модели в своем составе содержат структурно-параметрическое и функциональное описание. Методы функционирования логических моделей отображаются в редакторе (рис. 6, см. вторую сторону обложки). Для интеграции моделей и их переносимости выполняется генерация

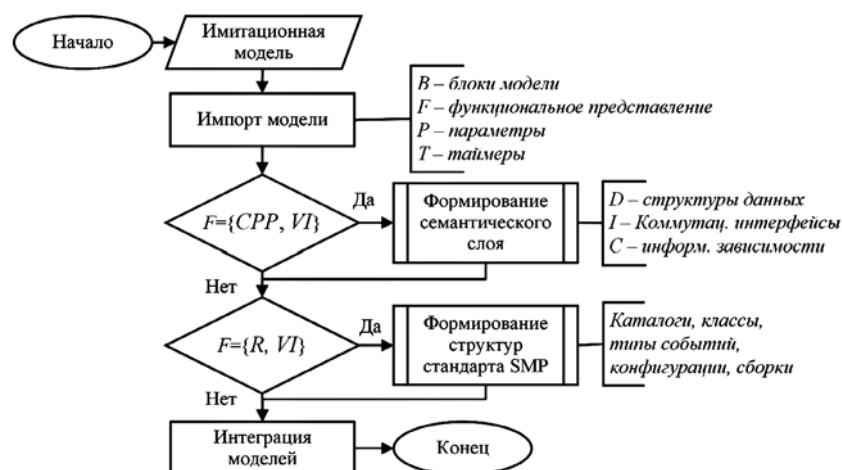


Рис. 4. Алгоритм интеграции разнородных имитационных моделей

структур, определенных в стандарте SMP2. Требования к структурам заданы в разделе 2а: Meta-model [12]. Создается набор каталогов, определяются классы, типы событий, задается конфигурация экземпляров, формируются сборки. Пользователь оперирует с ними с помощью заданных семантических конструкций.

Созданные структуры служат промежуточным звеном и обеспечивают вызов декларированных в модели функций путем передачи управления в библиотеку реализаций. После выполнения преобразований модели размещаются в банке моделей. В результате работы была построена комплексная модель командно-программного управления бортовой аппаратурой, фрагмент модели показан на рис. 7 (см. вторую сторону обложки).

Построение комплексной модели командно-программного управления позволило исследовать особенности функционирования бортовой аппаратуры, используя различные реализации отдельных подсистем и оперируя терминами и понятиями предметной области. Результаты имитационного моделирования отображаются в протоколе, который содержит источники и приемники данных, передаваемые пакеты, значения параметров и др. Фрагмент протокола имитационного моделирования приведен на рис. 8.

Заключение

Технология построения комплексных моделей обеспечивает взаимодействие разнородных имитационных моделей в единой среде моделирования. Расширение стандарта SMP2 семантическими конструкциями предметной области позволяет получать разнообразные и качественные решения, используя знания групп инженеров, имеющих большой опыт проектирования бортовой аппаратуры. Такой подход позволяет легко разбираться в устройстве как создаваемых конструктором собственных моделей, так и моделей, созданных другими специалистами, а также представлять моделируемую систему в целом.

Предложенная технология рассматривает три типа моделей: модели на основе правил,

Время начала передачи	Передающий интерфейс	Принимающий блок	Принимающий инт	Продолжительность пере...
Передающий блок: БУ БКУ				
00:00:00.1000000	КУ1 (Рел.)	МИ КИС	КУ1 (Рел.)	100
Длительность (мс): 100				
Напряжение (В): 5				
00:00:01.6600000	Выход (RS-232)	МИ КИС	От БУ БКУ (RS-232)	1
Передающий блок: МИ КИС				
00:00:01.6500000	На БУ БКУ (RS-232)	БУ БКУ	Вход (RS-232)	1
Поле данных сегмента - 64 07 F8				
Служебный байт - 64				
Прямой код РК - 07				
Инверсный код РК - F8				
Передающий блок: Приемник				
00:00:01.6400000	Выход (RS-422)	МИ КИС	От ПРМ (RS-422)	1
Данные:				
00 01 04 10 00 C1 64 07 F8 F3 FD				

Рис. 8. Фрагмент протокола имитационного моделирования

виртуальные приборы и модели, реализованные по стандарту SMP2. Применение других реализаций моделей требует их доработки в соответствии со стандартом или использования существующих инструментов преобразования, например, таких как программный комплекс MOSAIC, выполняющий автоматическую передачу моделей MATLAB/Simulink в SMP2-стандарт. Для использования моделей, не имеющих собственных инструментов построения SMP2-моделей, в дальнейшем может быть обобщен опыт интеграции виртуальных приборов. Переход моделей к стандарту SMP2 обеспечит их переносимость между всеми средами моделирования в космической отрасли, поддерживающими данный стандарт.

Создание программного обеспечения на основе международных стандартов обеспечит независимость российских разработок от зарубежных коммерческих систем. Внедрение технологии переносимости имитационных моделей на этапе конструирования бортового оборудования позволит российским производителям интегрироваться в международные проекты.

Список литературы

1. **Space engineering.** System engineering general requirements (ECSS-E-ST-10C) // European space agency (ESA): ESTEC. 2009. P. 100.
2. **Simulation modelling platform.** ECSS E-40-07 // ESA Requirements and Standards Division ESTEC. 2011. Vol. 1a—5a. P. 869.
3. **Sarkarati M., Merri M., Spada M.** Cloud based architectures in ground systems of space missions // GSW 2013. ESA. 2013. P. 14.
4. **Cazenave C., Arrouy W.** Implementing SMP2 standard within SimTG simulation infrastructure // SESP 2012: Simulation and EGSE for space programmes. ESTEC. Noordwijk. 2012. P. 13.

5. **Fritzen P., Segneri D., Pignede M.** SWARMSIM — The first fully SMP2 based Simulator for ESOC // Proceedings of the 11th International Workshop on Simulation & EGSE facilities for Space Programmes. ESTEC. 2010. P. 7.

6. **Strzepek A., Esteve F., Salas S., Millet B., Darnes H.** A training, operations and maintenance simulator made to serve the MERLIN mission // 14th International Conference on Space Operations. 2016. P. 11. DOI: 10.2514/6.2016-2410.

7. **Nozhenkova L., Isaeva O., Evsyukov A.** Software tools for modeling space systems' equipment command-and-software control // DEStech Transactions on Computer Science and Engineering (CECE2017). 2017. P. 87–92. DOI: 10.12783/dtscse/cece2017/14379.

8. **Исаева О. С., Грузенко Е. А., Вогоровский Р. В., Колдырев А. Ю.** Моделирование и анализ функционирования бортовой аппаратуры командно-измерительной системы космического аппарата // Информатизация и связь. № 1. 2015. С. 58–64.

9. **Ноженкова Л. Ф., Исаева О. С., Грузенко Е. А., Вогоровский Р. В., Колдырев А. Ю., Евсюков А. А.** Комплексная поддержка конструирования бортовых систем контроля и управления космических аппаратов на основе интеллектуальной имитационной модели // Информационные технологии. № 9. 2015. С. 706–714.

10. **Bress T.** Effective LabVIEW Programming. NTS Press, 2013. P. 720.

11. **Nozhenkova L., Isaeva O., Markov A., Koldyrev A., Vovorovskiy R., Evsyukov A.** Simulation infrastructure design on the basis of the space industry's international standards // Advances in Intelligent Systems Research (CAAI2017). 2017. Vol. 134. P. 138–141. DOI: 10.2991/caai-17.2017.28.

12. **Simulation modelling platform** — Volume 2a: Metamodel. ECSS E-40-07 // ESA Requirements and Standards Division ESTEC. The Netherlands. 2011. P. 169.

O. S. Isaeva, Ph. D., Senior Researcher, isaeva@icm.krasn.ru,

Institute of Computational Modelling of the Siberian Branch of the Russian Academy of Sciences,
Russia, Krasnoyarsk

The Technology of Building Complex Models in the Simulation Modeling Infrastructure

In the paper the technology for constructing complex simulation models of spacecraft onboard equipment based on the European Space Agency standard — Simulation model portability (SMP2) are proposed. The standard provides the principles of portability and reuse of simulation models. In addition to SMP2-models, virtual instruments and logical models the rules of which are set in knowledge bases are used in the work. The goal of the technology is to ensure the interaction of heterogeneous simulation models, their integration based on the principles of portability and universality of the SMP2 standard. The technology is based on the formalization of models. The model contains structural-parametric and functional description of the elements determining its composition and features, as well as the method of control of transitions between the conditions under modeling. The principles of unification allowing to build a model in the terms of the subject area, with their following translation to the program elements of the model working in accordance with the SMP2 standard are suggested. This approach allows the onboard equipment designer to understand both created by own models, and models created by other specialists, and also to represent the simulated system as a whole. The technology is tested in software designed for simulation of satellite systems. The construction of complex simulation models will make to provide accumulation and usage of expert knowledge about spacecraft onboard equipment functioning.

Keywords: spacecraft, onboard equipment, simulation modeling, simulation model portability standard (SMP2), simulation modeling infrastructure.

DOI: 10.17587/it.24.474-480

References

1. **Space engineering.** System engineering general requirements (ECSS-E-ST-10C), European space agency (ESA), ESTEC, 2009, pp. 100.

2. **Simulation modelling platform.** ECSS E-40-07. ESA Requirements and Standards Division ESTEC, The Netherlands, 2011, pp. 49.

3. **Sarkarati M., Merri M., Spada M.** Cloud based architectures in ground systems of space missions, *GSW 2013*, ESA, 2013, pp. 14.

4. **Cazenave C., Arrouy W.** Implementing SMP2 standard within SimTG simulation infrastructure, *SESP 2012: Simulation and EGSE for space programmes*, ESTEC, Noordwijk, 2012, pp. 13.

5. **Fritzen P., Segneri D., Pignede M.** SWARMSIM — The first fully SMP2 based Simulator for ESOC, *Proceedings of the 11th International Workshop on Simulation & EGSE facilities for Space Programmes*, ESTEC, 2010, pp. 7.

6. **Strzepek A., Esteve F., Salas S., Millet B., Darnes H.** A training, operations and maintenance simulator made to serve the MERLIN mission, *14th International Conference on Space Operations*, 2016, pp. 11, DOI: 10.2514/6.2016-2410.

7. **Nozhenkova L., Isaeva O., Evsyukov A.** Software tools for modeling space systems' equipment command-and-software control. *DEStech Transactions on Computer Science and Engineering (CECE2017)*, 2017, pp. 87–92, DOI: 10.12783/dtscse/cece2017/14379.

8. **Isaeva O. S., Gruzenko E. A., Vovorovskiy R. V., Koldyrev A. Yu.** Modelirovaniye i analiz funkcionirovaniya bortovoy apparatury komandno-izmeritel'noy sistemy kosmicheskogo apparata (Modeling and analysis of functioning of the spacecraft command and measuring system), *Informatsiya i Svjaz'*, 2015, no. 1, pp. 58–64 (in Russian).

9. **Nozhenkova L. F., Isaeva O. S., Gruzenko E. A., Vovorovskiy R. V., Koldyrev A. Yu., Evsyukov A. A.** Kompleksnaya podderzhka konstruirovaniya bortovykh sistem kontrolja i upravlenija kosmicheskikh apparatov na osnove intellektual'noj imitacionnoj modeli (Complex support of designing onboard control and management systems of spacecrafts on the basis of intellectual simulation model). *Informacionnye Tehnologii*, 2015, no. 9, pp. 706–714 (in Russian).

10. **Bress T.** Effective LabVIEW Programming. NTS Press. 2013. P. 720.

11. **Nozhenkova L., Isaeva O., Markov A., Koldyrev A., Vovorovskiy R., Evsyukov A.** Simulation infrastructure design on the basis of the space industry's international standards. *Advances in Intelligent Systems Research (CAAI2017)* 2017, vol. 134, pp. 138–141, DOI: 10.2991/caai-17.2017.28.

12. **Simulation modelling platform** — Volume 2a: Metamodel. ECSS E-40-07. ESA Requirements and Standards Division ESTEC, The Netherlands, 2011, pp. 169.

Н. О. Ефимов, аспирант, e-mail: n.o.efimov@gmail.com,
Е. Г. Жилияков, д-р техн. наук, проф., zhilyakov@bsu.edu.ru,
 Белгородский государственный национальный исследовательский университет

Распознавание фрагментов изображений рукописного текста. Характерные частотные интервалы

Исследование существующих методов распознавания изображений текста позволяет говорить о том, что в общем случае задача решена только для изображений печатного текста. В случае рукописного текста каждое решение является индивидуальным и нередко требует многочасового обучения либо вмешательства оператора. В статье предложена новая решающая процедура обнаружения идентичных фрагментов текста по заданному образцу, реагирующая на различия нормированных трансформант Фурье в заданных частотных интервалах. Исследована возможность выделения характерных для заданного символа частотных интервалов, использование которых может существенно уменьшить число признаков, необходимых для описания фрагмента.

Ключевые слова: обработка изображений, распознавание текстовых и рукописных изображений, субполосный метод, характерные частотные интервалы, решающая функция, мера близости, трансформанты Фурье

К настоящему времени накопилось множество цифровых изображений текста: сканированные книги, статьи, журналы, документы, как офисные, так и персональные (например, изображения паспорта, ИНН и др.). На многих интернет-ресурсах во избежание копирования и плагиата различного рода документов (например, дипломных и диссертационных работ) интересующая пользователя информация представляется в формате изображений. Однако обработка информации, представленной в подобной форме, затруднена невозможностью осуществить такие операции, как поиск по тексту, обнаружение и выделение интересующих фрагментов.

Разработанный подход позволяет осуществлять поиск интересующих фрагментов на изображениях текста (в том числе и рукописного). Так, можно выделить одно слово на изображении и осуществить поиск этого слова по всему документу, не изменяя формат файла. В случае цифрового документооборота можно найти все документы, подписанные одним и тем же человеком. Также представленный подход позволяет искать на изображениях заранее заданные слова для установления его содержания (например, для обнаружения пропаганды терроризма, наркотиков и т.д.). Предлагаемый метод прецедентного распознавания фрагментов изображений основан на использовании субинтервальных методов в частотной области [1, 2].

В общем случае системы распознавания имеют следующую структуру. Входные данные, подлежащие распознаванию, подаются на вход системы и подвергаются предобработке в целях их преобразования в необходимый для следующего этапа вид или для выделения из них необходимых характерных признаков. Далее на этапе принятия решения над обработанным массивом данных проводится ряд вычислений, и на основе их результатов формируется ответ, содержащий ожидаемые от системы сведения о входных данных. Содержание входных и выходных данных определяется назначением системы.

Цифровое изображение можно определить как двумерную функцию f_{ik} , где $i = 1, 2, \dots, M$; $k = 1, 2, \dots, N$ — координаты в пространстве (конкретно на плоскости), а значение f в любой точке, задаваемой парой координат, называется интенсивностью изображения в этой точке.

Частотным представлением функции f_{ik} называется следующее выражение [3]:

$$f_{ik} = \frac{1}{4\pi^2} \sum_{u=1}^M \sum_{v=1}^N F(u, v) e^{ju(i-1)} e^{jv(k-1)}, \quad (1)$$

$$i = 1, 2, \dots, M; k = 1, 2, \dots, N,$$

где j — мнимая единица ($j^2 = -1$), аргументы u и v — пространственные частоты, отражающие периодичность (цикличность) изменений исходной функции f_{ik} с изменением аргументов i и k .

Частотной областью называется координатная система, задающая соответствие между аргументами $F(u, v)$ и частотными переменными u и v [4, 5].

Прямоугольную область размера $M \times N$, задаваемую при $u = 0, 1, 2, \dots, M - 1$ и $v = 0, 1, 2, \dots, N - 1$, принято называть частотным прямоугольником. Частотный прямоугольник имеет те же размеры, что и исходное изображение.

Так как изображение является функцией не времени, а координаты, используется термин "пространственные частоты" (волновые числа), который подразумевается под словом "частота" далее по тексту.

В качестве весовой функции $F(u, v)$ можно использовать трансформанту Фурье:

$$F(u, v) = \sum_{i=1}^M \sum_{k=1}^N f_{ik} e^{-ju(i-1)} e^{-jv(k-1)}, \quad (2)$$

где $i = 1, 2, \dots, M$; $k = 1, 2, \dots, N$.

На основе равенства Парсевала [5] энергию изображения можно представить в виде суммы:

$$\sum_{i=1}^M \sum_{k=1}^N f_{ik}^2 = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} |F(u, v)|^2 dudv = \sum_{r=1}^{R_1} \sum_{m=1}^{R_2} P_{\Omega_r, m}, \quad (3)$$

где значение доли энергии $P_{\Omega_r, m}$ двумерного сигнала $f_{i, k}$ в двумерной частотной области Ω равно

$$P_{\Omega_r, m} = \frac{1}{4\pi^2} \iint_{(u, v) \in \Omega} |F(u, v)|^2 dudv, \quad (4)$$

а интервалы $\Omega_{r_1 r_2}$ определяют разбиение частотной области на интервалы:

$$\begin{aligned} \Omega_{r_1 r_2} : \{ & \Omega(u, v) | (u \in [\sigma_1, \sigma_r], v \in [\gamma_1, \gamma_r]) \cup \\ & \cup (u \in [\sigma_1, \sigma_r], v \in [-\gamma_r, -\gamma_1]) \cup \\ & \cup (u \in [-\sigma_r, -\sigma_1], v \in [-\gamma_r, -\gamma_1]) \cup \\ & \cup (u \in [-\sigma_r, -\sigma_1], v \in [\gamma_1, \gamma_r]) \}. \end{aligned}$$

Для нахождения точных значений долей энергии изображений в заданных частотных интервалах в работах [1, 6—8] был разработан и исследован следующий субинтервальный метод.

Если в правую часть представления (4) подставить определение (2), то после преобразований можно получить соотношение [6]

$$P_{\Omega} = \sum_{i_2=1}^M \left(\sum_{k_1=1}^N \left(\sum_{i_1=1}^M a_{i_1 i_2} f_{i_1 k_1} \right) \left(\sum_{k_2=1}^N b_{k_1 k_2} f_{i_2 k_2} \right) \right), \quad (5)$$

где

$$a_{i_1 i_2} = \begin{cases} \frac{\sin(\alpha_2(i_1 - i_2)) - \sin(\alpha_1(i_1 - i_2))}{\pi(i_1 - i_2)}, & i_1 \neq i_2, \\ \frac{\alpha_2 - \alpha_1}{\pi}, & i_1 = i_2; \end{cases} \quad (6)$$

$$b_{k_1 k_2} = \begin{cases} \frac{\sin(\beta_2(k_1 - k_2)) - \sin(\beta_1(k_1 - k_2))}{\pi(k_1 - k_2)}, & k_1 \neq k_2, \\ \frac{\beta_2 - \beta_1}{\pi}, & k_1 = k_2. \end{cases} \quad (7)$$

Введем матрицы $A = (a_{i_1 i_2})$ и $B = (b_{k_1 k_2})$ размерностей $M \times M$ и $N \times N$ соответственно, в соответствии с выражениями (5), (7). Матрицы A и B называются субполосными матрицами [1, 2, 6, 7].

Тогда выражение (5) позволяет записать формулу вычисления точных значений энергии P_{Ω} дискретного двумерного сигнала Φ в частотной двумерной области Ω , используя матричные обозначения, в следующем виде:

$$P_{\Omega} = \text{tr}(A\Phi B\Phi^T), \quad (8)$$

где Φ — исходное изображение; tr — след матрицы; A и B — субполосные матрицы.

Соотношение (8) определяет метод субинтервальной обработки изображений на основе частотных представлений и позволяет для нахождения точных значений энергии двумерного сигнала в любой частотной двумерной области построить вычислительную процедуру, не находя при этом трансформанту Фурье.

Концептуальные основы процедуры идентификации фрагментов изображений текста

Исходные условия: на изображении текста Φ_0 размерностью $M \times N$ выделяется прямоугольная область Φ_p размерностью $M_p \times N_p$, которая в дальнейшем именуется эталонным фрагментом (индекс p здесь означает прецедент).

Далее из изображения Φ_0 определенным образом формируется множество фрагментов Φ_k размерностью $M_p \times N_p$, где k — число полученных фрагментов, каждый из которых необходимо сопоставить с прецедентом. В случае изображений с текстом процедура поиска фрагментов заключается в перемещении маски по строкам текста с шагом в один пиксель.

На исходном изображении Φ_0 необходимо найти идентичные фрагменты. Для сопоставления фрагментов Φ_k с эталоном Φ_p нужно иметь решающую функцию, которая задает меру идентичности, значения которой вычисляются по значениям пикселей в сравниваемых фрагментах. Область значений решающей функции, используемой в данной работе, — положительные вещественные числа.

Исходная гипотеза H_0 : сравниваемые фрагменты идентичны. Для проверки гипотезы используется решающая функция. Гипотеза отвергается, когда значение решающей функции больше порога, значение которого выбирается исходя из требования обеспечения заданного уровня вероятности ошибок первого рода (ложных тревог). Решающая функция должна отвечать принципу максимизации вероятности правильного принятия противоположной гипотезы, когда она верна (минимизации вероятности ошибок второго рода) (рис. 1, см. третью сторону обложки).

Для определения порога необходимо провести предварительное обучение системы, превышение порога свидетельствует о неидентичности сравниваемых объектов. Обучение проводится на основе единственного образца, который искусственно трансформируется в соответствии с возможными реальными трансформациями.

Решающая функция на основе субполосной меры близости спектров двумерных дискретных сигналов

Предлагается использовать расстояние между нормированными спектрами в заданном частотном интервале:

$$S_{rm} = \frac{1}{2\pi} \iint_{\omega_1, \omega_2 \in \Omega} \left| \frac{F_1(\omega_1, \omega_2)}{\sqrt{P_{rm}^{\Phi_p}}} - \frac{F_2(\omega_1, \omega_2)}{\sqrt{P_{rm}^{\Phi_k}}} \right|^2 d\omega_1 d\omega_2 =$$

$$= \frac{1}{2\pi} \iint_{\omega_1, \omega_2 \in \Omega} \left(\left(\frac{F_1(\omega_1, \omega_2)}{\sqrt{P_{rm}^{\Phi_p}}} \right)^2 - \frac{2F_1(\omega_1, \omega_2)F_2(\omega_1, \omega_2)}{\sqrt{P_{rm}^{\Phi_p} P_{rm}^{\Phi_k}}} + \left(\frac{F_2(\omega_1, \omega_2)}{\sqrt{P_{rm}^{\Phi_k}}} \right)^2 \right) d\omega_1 d\omega_2, \quad (9)$$

где F_1, F_2 — трансформанты Фурье эталонного и сравниваемого объектов соответственно;

$$P_{rm}^{\Phi_p} = \text{tr}(A_r \Phi_p B_m \Phi_p^T); \quad (10)$$

$$P_{rm}^{\Phi_k} = \text{tr}(A_r \Phi_k B_m \Phi_k^T), \quad (11)$$

где A_r, B_m — субполосные матрицы, элементы которых вычисляются согласно выражениям (6), (7).

В основе этой меры используются нормированные субполосные коэффициенты корреляции в частотном интервале:

$$p_{rm}^{pk} = \frac{1}{2\pi} \iint_{\omega_1, \omega_2 \in \Omega} \frac{F_1(\omega_1, \omega_2)F_2(\omega_1, \omega_2)}{\sqrt{P_{rm}^{\Phi_p} P_{rm}^{\Phi_k}}} d\omega_1 d\omega_2. \quad (12)$$

Для сопоставления фрагментов цифровых изображений в данной работе используется мера близости следующего вида:

$$S_{pk} = 2 \left(\frac{1}{R_1 R_2} \sum_{j=1}^{R_2} \sum_{i=1}^{R_1} (1 - p_{rm}^{pk}) \right), \quad (13)$$

где

$$p_{rm}^{pk} = \frac{\text{tr}(A_r \Phi_p B_m \Phi_k^T)}{\sqrt{P_{rm}^{\Phi_p} P_{rm}^{\Phi_k}}}. \quad (14)$$

Выражение (13) является используемой в данной работе мерой идентичности фрагментов; R_1, R_2 — число информационных частотных интервалов для каждой координаты.

Информационным частотным интервалом будем считать такой интервал, что

$$P_{11} \geq h_1; \quad (15)$$

$$P_{1m} \geq h_2, \quad m = 2, \dots, R_b; \quad (16)$$

$$P_{r1} \geq h_3, \quad r = 2, \dots, R_a; \quad (17)$$

$$P_{rm} \geq h_4, \quad r = 2, \dots, R_a; \quad m = 2, \dots, R_b, \quad (18)$$

где R_a, R_b — число интервалов по заданным осям.

Порог h представляет собой среднее значение энергии изображения на единицу частотной полосы.

Рассчитываются четыре значения порога, так как частотные интервалы имеют отличные друг от друга площади:

$$h_1 = \frac{\|F\|^2}{\pi^2} \cdot \frac{\pi}{2Ra + 1} \cdot \frac{\pi}{2Rb + 1}; \quad (19)$$

$$h_2 = \frac{\|F\|^2}{\pi^2} \cdot \frac{2\pi}{2Ra + 1} \cdot \frac{\pi}{2Rb + 1}; \quad (20)$$

$$h_3 = \frac{\|F\|^2}{\pi^2} \cdot \frac{\pi}{2Ra+1} \cdot \frac{\pi}{2Rb+1}; \quad (21)$$

$$h_4 = \frac{\|F\|^2}{\pi^2} \cdot \frac{2\pi}{2Ra+1} \cdot \frac{\pi}{2Rb+1}. \quad (22)$$

Основная гипотеза (H_0 — сравниваемые фрагменты идентичны) отвергается при выполнении неравенства

$$S_{pk} > h_p, \quad (23)$$

где h_p — граница критической области, которая определяется на этапе обучения.

Представленный выше метод позволяет получить следующие результаты: вероятность

ошибки первого рода равна 0 (обнаружены все искомые фрагменты); вероятность ошибки второго рода равна 0,0052 (к классу искомого неверно отнесены 6788 фрагментов из 1 308 150). Подробно исследование метода представлено в работе [9].

Рис. 2—4 иллюстрируют тот факт, что для разных символов информационными интервалами являются различные частотные интервалы.

Представляет интерес обнаружение таких частотных интервалов, которые будут являться информационными только для одного символа или небольшой группы символов. Присутствие подобной закономерности потенциально позволит повысить качество распознавания с точки зрения уменьшения вероятности ошибки первого рода при фиксированном значении вероятности ошибки первого рода.

Цель исследования — выделить характерные для данного символа частотные интервалы.

Условия эксперимента

1. Набор объектов — заглавные буквы русского алфавита. Их общее число составило 29 символов, так как нет слов, начинающихся с Ъ, Ь, Ы. Размерность каждого символа 145×145 . Число субполосных интервалов $37 \times 37 = 1369$.

2. Поочередно вычисляются информационные частотные интервалы для каждого из объектов.

3. Определяются такие частотные интервалы, которые являются информационными не более чем для 15 объектов. Назовем их характерными.

На рис. 5, 6 (см. третью сторону обложки) представлены информационные частотные интервалы для символов У, Э соответственно.

Значение по оси аппликат означает число объектов, для которых интервал является информационным. Интервалы, являющиеся информационными для всех объектов (низкочастотная область), для наглядности приравнены к нулю. Для отдельного символа наиболее характерными являются те интервалы, значение которых по оси аппликат менее 5.

Обозначение $[r, m]$ задает координаты отдельного интервала в двумерной частотной области Ω . Так, $[4, 18]$ — это частотный интервал с координатами $r = 4, m = 18$.

Уникальные частотные интервалы — такие частотные интервалы, которые являются информационными только для одного символа (табл. 1).

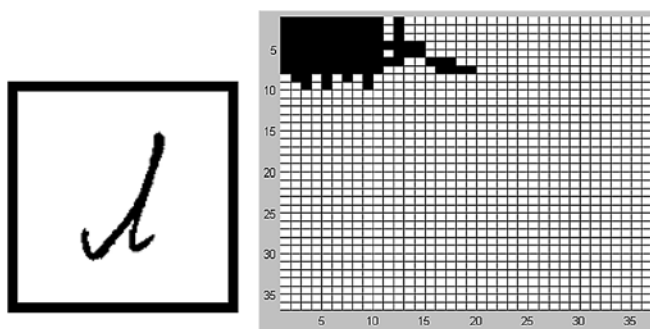


Рис. 2. Изображение символа Л размерностью 145×145 и его информационные частотные интервалы

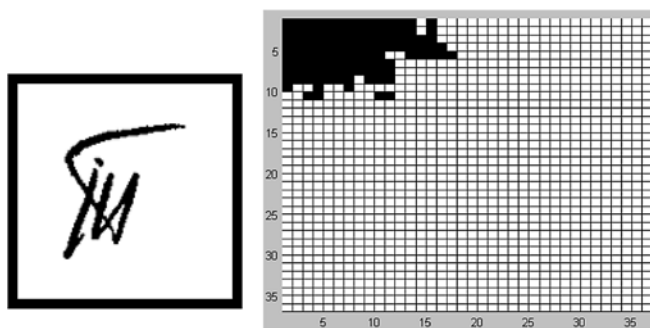


Рис. 3. Изображение символа Т размерностью 145×145 и его информационные частотные интервалы

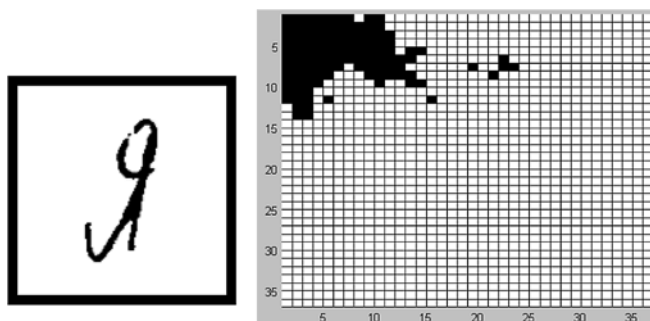


Рис. 4. Изображение символа Я размерностью 145×145 и его информационные частотные интервалы

Таблица 1

Уникальные частотные интервалы

Символ	Число частотных интервалов	Координаты частотных интервалов
А	1	[4,18]
Б	8	[12,1], [12,5], [12,6], [13,4], [13,5], [13,6], [14,5], [15,6]
В	3	[3,19], [3,21], [5,22]
Г	1	[12,3]
Е	1	[3,11]
Ж	3	[3,10], [9,12], [10,17]
И	3	[3,17], [4,17], [6,19]
К	4	[7,5], [10,15], [10,16], [11,18]
Л	3	[6,16], [7,17], [7,18]
М	3	[7,2], [7,13], [9,11]
Н	9	[4,7], [5,8], [4,21], [5,21], [5,22], [5,23], [6,7], [6,24], [6,25]
О	1	[7,16]
Т	8	[1,8], [6,22], [7,22], [7,23], [8,21], [9,13], [11,15], [13,2]
Ф	2	[11,7], [11,9]
Х	1	[1,9]
Ц	1	[6,2]
Ч	1	[7,1]
Щ	1	[7,6]
Э	2	[12,1], [12,2]
Ю	2	[7,24], [10,8]
Я	4	[2,15], [3,15], [4,16], [10,11]

Таким образом, 21 символ из 29 имеет хотя бы один уникальный частотный интервал (табл. 2).

Из 29 символов 27 имеют частотные интервалы, характерные для четырех и менее объектов.

Таким образом, можно сделать вывод, что большее число символов возможно охарактеризовать небольшим числом определенных частотных интервалов (порядка 5...20, при общем числе интервалов 1359). Также необходимо учитывать, что для каждого почерка характерными могут быть различные частотные интервалы. Дальнейшие исследования направлены на вычисление вероятностей ошибок при распознавании фрагментов рукописного текста с использованием характерных частотных интервалов.

Таблица 2

Частотные интервалы, характерные для четырех и менее объектов

Символ	Число частотных интервалов	Координаты частотных интервалов
А	5	[1,19], [2,14], [4,20], [5,15], [7,7]
Б	12	[2,14], [1,14], [6,13], [9,10], [9,14], [10,6], [10,7], [11,1], [11,5], [11,6], [12,4], [13,1]
В	3	[2,14], [4,20], [10,6]
Г	4	[11,5], [12,4], [1,11], [13,3]
Д	3	[1,14], [6,13], [1,11]
Е	3	[9,10], [10,6], [1,10]
Ж	5	[1,10], [2,11], [6,18], [8,4], [8,13]
З	3	[11,5], [13,3], [12,3]
И	8	[5,15], [2,11], [4,15], [5,10], [5,16], [5,17], [5,18], [5,19]
К	3	[9,14], [8,4], [5,10]
Л	4	[2,11], [6,15], [6,17], [7,19]
М	5	[6,13], [6,18], [6,17], [5,13], [6,14]
Н	9	[7,7], [5,10], [5,18], [5,19], [6,9], [5,9], [5,20], [6,20], [6,23]
П	1	[10,7]
Р	3	[1,11], [6,15]
С	4	[6,13], [9,10], [10,6], [7,21]
Т	9	[7,7], [11,5], [13,3], [2,11], [8,13], [12,3], [12,2], [7,19], [8,12]
У	4	[5,18], [5,19], [5,20], [6,23]
Й	3	[12,3], [6,9], [12,2]
Ф	4	[10,7], [1,10], [8,12], [10,9]
Х	1	[5,13]
Ц	7	[1,19], [6,18], [5,10], [5,16], [5,18], [6,9], [6,20]
Ш	6	[4,15], [6,15], [5,13], [6,14], [5,9], [7,21]
Щ	7	[5,15], [13,1], [8,4], [6,15], [6,17], [6,14], [7,24]
Э	2	[11,1], [13,1]
Ю	10	[9,10], [10,7], [11,6], [5,17], [5,13], [6,20], [10,9], [7,24], [1,15], [10,10]
Я	6	[5,15], [4,15], [5,16], [5,17], [1,15], [10,10]

Список литературы

1. **Жиляков Е. Г., Черноморец А. А.** Вариационные алгоритмы анализа и обработки изображений на основе частотных представлений. М.: ГиК, 2009. 146 с.
2. **Жиляков Е. Г., Черноморец А. А., Лысенко И. В.** Метод определения точных значений долей энергии изображений в заданных частотных интервалах // Вопросы радиоэлектроники. 2007. Вып. 4. С. 115–123.
3. **Семенов Ю. А.** Алгоритмы телекоммуникационных сетей. Часть 1. Алгоритмы и протоколы каналов и сетей передачи данных. М.: Бинوم. Лаборатория знаний, 2007. 640 с.
4. **Рабинер Л., Голд Б.** Теория и применение цифровой обработки сигналов. М.: Мир, 1978.
5. **Сергиенко А. Б.** Цифровая обработка сигналов. СПб.: Питер, 2002. 603 с.
6. **Жиляков Е. Г., Прохоренко Е. И.** Частотный анализ речевых сигналов // Научные ведомости Белгородского государственного университета. 2006. № 2 (3.118), Вып. 3. С. 201–208.
7. **Черноморец А. А., Иванов О. Н.** Метод анализа распределения энергий изображений по заданным частотным интервалам // Научные ведомости БелГУ. Сер. История. Политология. Экономика. Информатика. 2010. № 19 (90). Вып. 16/1. С. 161–166.
8. **Черноморец А. А., Жиляков Е. Г., Голощапова В. А., Болгова Е. В.** Оценка эффективности субполосного внедрения данных в изображение // Научные ведомости БелГУ. Сер.: История. Политология. Экономика. Информатика. 2014. № 8–1 (179). С. 200–206.
9. **Ефимов Н. О., Жиляков Е. Г.** О субполосном анализе изображений // Научные ведомости БелГУ. Сер. История. Политология. Экономика. Информатика. 2015. № 1 (198), вып. 33/1. С. 118–124.

N. O. Efimov, Graduate Student, e-mail: n.o.efimov@gmail.com,

E. G. Zhiljakov, D. Sc., Professor, department of Information and telecommunication systems and technologies, Belgorod National Research University, Belgorod, 308000, Russian Federation

Handwritten Text Fragment Recognition. Typical Subband Intervals

The research of existing methods of identification scanning text images allow to draw a conclusion, what in general case problem solve only for printed text images. In case of handwritten text each solution is individual, and frequently need a long time education or operator inspection. In the article introduce new decision procedure of precedential identical fragment detection, which react on normalized Fourier transform differences in defined frequency intervals. Possibility of detection typical subband intervals for specified symbol is analyzed. Using typical subband intervals can appreciably decrease number of features, necessary for fragment description.

Keywords: image processing, image recognition, HWR, OCR, substrip method, subband method, typical subband intervals, decision procedure, measure of proximity, Fourier transform

DOI: 10.17587/it.24.481-486

References

1. **Zhiljakov E. G., Chernomorec A. A.** *Variacionnye algoritmy analiza i obrabotki izobrazhenij na osnove chastotnyh predstavlenij* (Variational algorithms for analysis and image processing based on frequency representations), Moscow, GiK, 2009, 146 p. (in Russian).
2. **Zhiljakov E. G., Chernomorec A. A., Lysenko I. V.** *Metod opredelenija tochnyh znachenij dolej jenerгии izobrazhenij v zadannyh chastotnyh intervalah* (The method of determination the exact values of the image energy in frequency intervals), *Voprosy Radiojelektroniki*, 2007, iss. 4, pp. 115–123 (in Russian).
3. **Semenov Ju. A.** *Algoritmy telekommunikacionnyh setej. Chast' 1. Algoritmy i protokoly kanalov i setej peredachi dannyh* (Algorithms of telecommunication networks. Part 1. Algorithms and protocols of channels and data networks), Moscow, Binom. Laboratorija znaniy, 2007, 640 p. (in Russian).
4. **Rabiner L., Gold B.** *Teoriya i primenenie tsifrovoi obrabotki signalov* (Theory and application of digital signal processing), Moscow, Mir, 1978 (in Russian).
5. **Sergienko A. B.** *Cifrovaja obrabotka signalov* (Digital signal processing), SPb, Piter, 2002, 603 p. (in Russian).
6. **Zhiljakov E. G., Prohorenko E. I.** Chastotnyj analiz rechevyh signalov, *Nauchnye vedomosti Belgorodskogo gosudarstvennogo universiteta*. 2006, no. 2 (3.118), iss. 3. pp. 201–208 (in Russian).
7. **Chernomorec A. A., Ivanov O. N.** Metod analiza raspredelenija jenerгии izobrazhenij po zadannyh chastotnym intervalam, *Nauchnye vedomosti BelGU*. Ser. Istorija. Politologija. Jekonomika. Informatika, 2010, no. 19 (90), iss. 16/1, pp. 161–166 (in Russian).
8. **Chernomorec A. A., Zhiljakov E. G., Goloshhapova V. A., Bolgova E. V.** Ocenka jeffektivnosti subpolosnogo vnedrenija dannyh v izobrazhenie, *Nauchnye vedomosti BelGU*. Ser.: Istorija. Politologija. Jekonomika. Informatika, 2014, no. 8–1 (179), pp. 200–206 (in Russian).
9. **Efimov N. O., Zhiljakov E. G.** O subpolosnom analize izobrazhenij, *Nauchnye vedomosti BelGU*. Ser. Istorija. Politologija. Jekonomika. Informatika, 2015, no. 1 (198), iss. 33/1, pp. 118–124 (in Russian).

А. С. Наумов, инженер-программист, e-mail: alex.naumov53@mail.ru
ООО "Электра-Н", Великий Новгород, 173003

Разработка программного обеспечения для реконструкции фрагментированных изображений

Рассмотрены технические аспекты разработки программного комплекса автоматизированной сборки фрагментированных изображений, предназначенного для восстановления разрушенных объектов с плоской поверхностью в криминалистике, реставрации, археологии. Описана общая архитектура комплекса, логика работы и механизмы взаимодействия отдельных компонентов. Разработаны форматы хранения данных, рассмотрены особенности практической реализации используемых методов, моделей и алгоритмов.

Ключевые слова: программный комплекс, реконструкция изображений, стыковка фрагментов, композиция, контур, атрибут, визуализация, формат XML

Введение

Автоматизация и виртуализация процесса восстановления фрагментированных объектов являются актуальными задачами в криминалистике, археологии, реставрации и архивном деле [1]. В большинстве случаев подобное восстановление предполагает подбор стыкующихся между собой фрагментов с последующей сборкой композиции из найденных пар. Выполнение таких операций вручную требует значительных временных и человеческих ресурсов, приводит к физическому разрушению материалов.

К настоящему времени опубликовано значительное число работ, посвященных автоматизированной реконструкции керамических изделий [2], бумажных документов [3], фотографий [4], [5], фресковых композиций [6], а также общим теоретическим аспектам сравнения формы объектов [7]. Значимым примером является проект Института Фраунгофера по восстановлению документов архива Госбезопасности Восточной Германии [8].

В российской практике известны примеры виртуальной реконструкции фресок путем подбора положения фрагментов по фотографиям, сделанным до разрушения [9], однако подобная работа не предусматривала непосредственно автоматизированного поиска стыков.

Реконструкция относительно плоских объектов может быть сведена к задаче синтеза двухмерного цифрового изображения из фрагментов произвольной формы. Для этого

используют изображения поверхности фрагментов, дополненные информацией об их характеристиках, формируемой вручную оператором-экспертом.

Реконструкция фрагментированного изображения состоит из следующих этапов.

1. Оцифровка материалов путем сканирования или фотографирования.

2. Формирование дескрипторов фрагментов, в частности:

- предварительная обработка изображений (сегментация, выделение отдельных фрагментов, их маркировка и сохранение);
- анализ изображений для описания отдельных характеристик;
- формирование и сохранение дескрипторов.

3. Поиск и локализация стыков, осуществляемые путем перебора потенциальных пар фрагментов и сравнения их характеристик по выбранным критериям.

4. Синтез и визуализация изображения композиции на основе найденных стыков.

Ранее автором был получен ряд теоретических результатов в рамках данной проблемы, в том числе предложены новые математические методы и алгоритмы сравнения характеристик фрагментов и синтеза композиции. Для их практической реализации и экспериментальных исследований автор разрабатывает соответствующий программный комплекс. В работе рассмотрены технические особенности подобного комплекса, основные аспекты его разработки и применения.

1. Постановка задачи

По результатам исследования проблемы автоматизированной реконструкции изображений из фрагментов [10] автором были разработаны методы описания и сравнения геометрических и цветовых характеристик контурной линии фрагмента с использованием многооткликовых статистических моделей [11], динамической модификации контура [12], а также метод сравнения цветовой палитры фрагментов. Эти методы ранее были реализованы автором в виде отдельных программ, на которые получены свидетельства о государственной регистрации. Предложен комплексный подход к поиску стыков двух фрагментов на основе совокупности их характеристик, а также алгоритмы синтеза и визуализации композиции [13].

Полученные результаты представляют собой основу технологии реконструкции, которая может быть использована для сборки таких объектов, как разорванные и разрезанные бумажные документы, керамические изделия, фрески и другие материалы с относительно плоской поверхностью. В дальнейшем технология может быть расширена и дополнена новыми методами.

Упрощенная схема технологии приведена на рис. 1. Она включает в себя основной каркас, обобщенно реализующий функции обработки исходных данных, поиска стыков, синтеза и визуализации композиции, а также дополнительные модули, отвечающие за описание и сравнение отдельных характеристик фрагментов.

К разрабатываемому программному обеспечению предъявлены следующие требования.

1. Реализация всех этапов реконструкции от ввода фрагментов до сохранения результирующих композитных изображений.

2. Реализация ранее разработанных автором методов и алгоритмов.

3. Возможность анализа тестовых выборок фрагментов для уточнения критериев сравнения их характеристик.

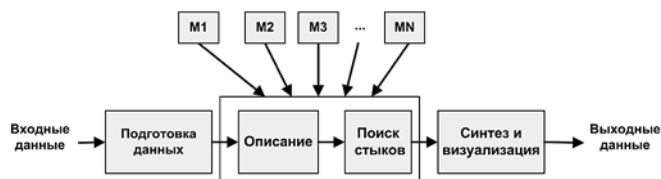


Рис. 1. Обобщенная схема технологии реконструкции: M1...MN — модули описания и сравнения характеристик фрагментов

4. Разделение функциональности в соответствии с этапами реконструкции.

5. Учет количественных параметров качества сборки.

6. Возможность последующего расширения функциональности за счет включения модулей работы с дополнительными характеристиками фрагментов.

7. Доступность аппаратных и программных средств.

Разрабатываемый комплекс предполагает обработку ограниченных объемов исходных данных (не более 1,5–2 тысяч фрагментов) и предназначен прежде всего для исследовательских целей. Предполагается сочетание автоматических вычислительных процедур и работы оператора-эксперта на всех этапах реконструкции.

2. Состав программного комплекса

С учетом указанных выше требований предложена архитектура программного обеспечения, предполагающая разделение на три отдельные программы (рис. 2), взаимодействующие между собой через хранилище данных и использующие общую библиотеку модулей.

Программа описания фрагментов (ПОФ) предназначена для ввода и обработки исходных данных с последующим формированием библиотеки фрагментов и их дескрипторов в соответствии с заданным набором описываемых характеристик.

Программа поиска стыков (ППС) осуществляет перебор потенциально возможных пар фрагментов, сравнение их характеристик, выявление и локализацию области стыка, а также расчет взаимного расположения фрагментов. В программу включены функции анализа из-



Рис. 2. Структура программного комплекса

вестных стыков для подбора параметров критериев сравнения (анализ тестовых выборок).

Программа визуализации композиции (ПВК) осуществляет синтез, визуализацию и сохранение результирующего изображения реконструированной композиции на основе найденных пар стыкующихся между собой фрагментов, а также реализует процедуры оценки качества проведенной реконструкции.

Хранилище данных осуществляет функцию долговременного хранения информации, формируемой в процессе работы указанных программ, и в простейшем случае представляет собой файловую систему с соответствующей структурой каталогов, форматами файлов и соглашениями по их именованию. К хранимым данным относятся:

- исходные изображения совокупностей фрагментов;
- выделенные изображения отдельных фрагментов;
- дескрипторы фрагментов;
- дескрипторы стыков;
- результирующие изображения.

Модули описания и сравнения отдельных характеристик фрагментов реализованы в виде библиотек, совместно используемых ПОФ и ППС.

Данная архитектура обеспечивает оптимальное разделение функциональности, аппаратных и людских ресурсов и позволяет выполнять отдельные этапы реконструкции в разные временные промежутки. Изменения, вносимые в одну из программ, в минимальной степени затрагивают остальные.

3. Логика работы и функциональные возможности программ

Реализуемая технология предполагает использование следующих характеристик фрагментов для поиска и локализации стыка:

- внешние атрибуты, задаваемые оператором;
- геометрический и цветовой контур, представленный в нескольких вариантах с различными детализацией, способом выбора узловых точек и числом описываемых параметров;
- векторы, описывающие такие характеристики, как направление строк текста, мазков кисти и иные геометрические особенности изображений на поверхности;
- цветовая палитра изображения;
- текстурные характеристики.

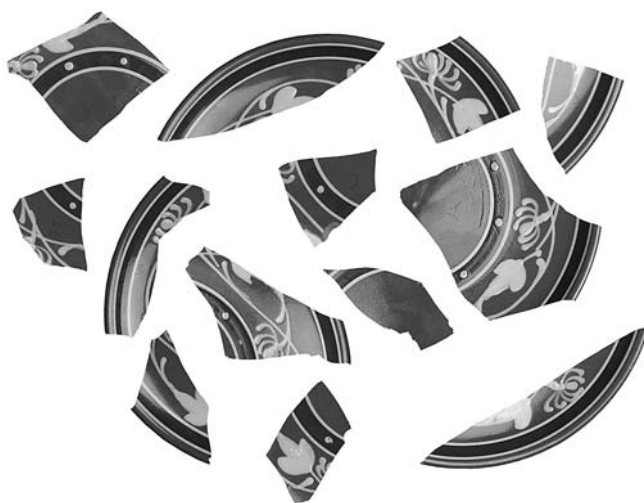


Рис. 3. Исходное изображение группы фрагментов

На вход программы описания подаются сканированные изображения групп фрагментов на контрастном фоне (рис. 3), после чего выполняется их автоматическая сегментация с выделением контура. Она включает в себя:

- построение бинарной маски изображения, на которой пикселям фона соответствует значение 0, а пикселям фрагментов — значение 1;
- обход контура с одновременным определением границ фрагмента;
- копирование области фрагмента с сохранением его изображения в отдельный файл;
- создание и сохранение заготовки дескриптора.

После формирования заготовок по команде пользователя выполняется дальнейшая обработка:

- удаление ошибочных фрагментов;
- описание контура [10, 11] с возможностью ручного задания границ участков (сегментов);
- задание внешних атрибутов (тип краски, тип материала, толщина фрагмента и т.п.);
- задание векторных характеристик;
- описание цвета и текстуры;
- сохранение данных в файл дескриптора.

Результатом работы программы описания является набор изображений фрагментов и их дескрипторов, сохраняемых в отдельные файлы, структура которых описана далее.

Предусмотрена возможность автоматической классификации фрагментов на основе внешних атрибутов [13], что позволяет кратно снизить размерность задачи поиска стыков за счет исключения заведомо невозможных вариантов.

Поскольку в реальных задачах (особенно связанных с реконструкцией фресковых композиций) качественное выделение границ объекта затруднено вследствие разрушений материала, в программе предусмотрен режим ручного выделения контура посредством его отрисовки с использованием графического планшета.

На вход программы поиска стыков поступают следующие данные:

- список фрагментов, в рамках которого осуществляется поиск;
- дескрипторы фрагментов;
- протокол сравнения.

Поиск стыков осуществляется путем полного перебора доступных пар фрагментов из всего множества или отдельных классов с последующим сравнением их характеристик и локализацией области стыка, для чего используется трехэтапный метод сравнения характеристик, предложенный автором в работе [13]. Он позволяет повысить достоверность идентификации стыков за счет использования совокупности признаков и является дополнительным фактором ограничения перебора для оптимизации производительности. Суть метода заключается в последовательном отсеивании неподходящих вариантов, при этом сравнение наиболее вычислительно сложных характеристик по возможности выполняется на более поздних стадиях конвейера. Метод включает в себя классификацию фрагментов, поиск и подтверждение стыков.

Порядок сравнения характеристик, их значимость и критерии определяются протоколом сравнения — структурой данных, описываемой в виде XML-документа. Локализация об-

ластей стыков проводится путем выявления совмещающихся сегментов контура [12].

Для каждого найденного стыка формируется дескриптор, содержащий параметры взаимной привязки фрагментов, который в виде XML-документа сохраняется в заданный каталог.

Предусмотрена процедура ручного подтверждения стыков, в ходе которой эксперту предлагается просмотреть найденные варианты стыков между двумя фрагментами, выбрав наиболее подходящий из них. Для уменьшения трудоемкости оператору отображается не более пяти ранжированных по весу вариантов. Результатом работы программы является набор дескрипторов стыков.

В зависимости от требуемого уровня автоматизации и особенностей конкретной задачи возможны различные пути реализации синтеза и визуализации композиции. В данном комплексе реализован метод, основанный на обходе графа связности в ширину с учетом факторов приоритета. Модель композиции представляется взвешенным неориентированным графом, в котором каждая вершина соответствует фрагменту, а ребро — стыку между фрагментами. Вес ребра является интегральной характеристикой, учитывающей длину стыка и значения критериев сравнения. Алгоритм синтеза описан в работе [13].

Программа визуализации композиции может быть использована как для ручной, так и для автоматизированной реконструкции объектов и предусматривает следующий набор функций:

- загрузка дескрипторов стыков;
- просмотр одиночных фрагментов;
- просмотр стыкующихся пар;
- сборка участка композиции с обходом графа от выбранного фрагмента и заданной глубиной;
- коррекция параметров стыка как в числовой форме, так и за счет перемещения изображений в графическом интерфейсе;
- сохранение и загрузка дескриптора композиции;
- оценка качества сборки;
- сохранение результирующих изображений.

Для оценки качества сборки предусмотрена возможность построения карты перекрытий (рис. 4), отражающей отношение площадей зон

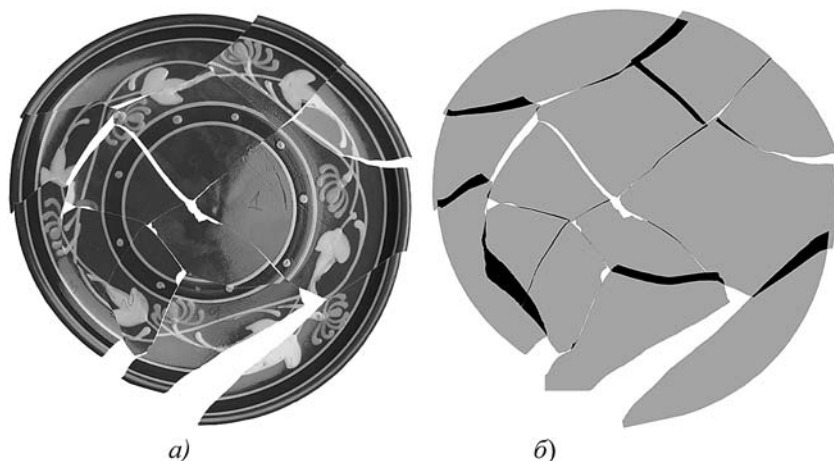


Рис. 4. Изображение собранной композиции (а) и соответствующей ей карты перекрытий (б)

взаимного наложения фрагментов к их общей площади. Идеальная композиция не должна содержать перекрытий, однако в реальных задачах можно говорить лишь о минимизации их количества. Технология предусматривает проверку глобальной совместимости фрагментов (размеры и форма композиции, приоритет стыков, образующих общий угол трех и более фрагментов) [13], реализация данного функционала будет выполнена в рамках дальнейших этапов разработки.

4. Организация хранения данных

Исходные материалы предоставляются в виде цветных изображений формата TIFF или JPEG, после обработки которых программа описания фрагментов формирует BMP-файлы формата ARGB глубиной цвета 8 бит на канал. Альфа-канал изображения представляет собой маску выделения, в которой пикселям фона соответствует значение 0, пикселям фрагмента — значение 255.

Дескрипторы фрагментов хранятся в XML-формате, обеспечивающем иерархическую структуру данных, пригодном для ручного редактирования и обеспечивающем расширяемость описания по мере развития функциональности комплекса. Документ имеет следующую структуру:

```
<fragment id="" description="">
  <image></image>
  <features>
    <feature1>
      ...
    </feature1>
    ...
    <featureN>
      ...
    </featureN>
  </features/>
</fragment>.
```

Для корневого элемента `<fragment>` определяются уникальный идентификатор (`id`) и описание (`description`) в виде атрибутов. Элемент `<image>` содержит путь к файлу с изображением. Далее следуют элементы `<feature>`, содержащие описания отдельных характеристик. Внутренняя структура каждого из этих элементов определяется и формируется соответствующим модулем описания.

Дескрипторы стыков и композиций хранятся в XML-документах следующей структуры:

```
<composition id="" description="">
  <fragments>
    <fragment id="" description="">
      <image></image>
    </fragment>
  <matches>
    <match id1="" id2="" weight="">
      <point1 x="" y=""/>
      <point2 x="" y=""/>
      <vector1 x="" y=""/>
      <vector2 x="" y=""/>
    </match>
  </matches>
</composition>.
```

Корневой элемент `<composition>` также содержит атрибуты идентификатора и описания. Элемент `<fragments>` задает перечисление фрагментов, входящих в данную пару или композицию. Элементы `<match>` описывают стык двух фрагментов с идентификаторами `id1` и `id2` соответственно и содержат информацию об их взаимном расположении — координаты точки привязки (элементы `<point1>` и `<point2>`) и вектора, соответствующего области стыка (`<vector1>` и `<vector2>`) на первом и втором фрагментах соответственно. На основе этой информации программа визуализации композиции выполняет геометрическую трансформацию изображений фрагментов при синтезе конечного изображения так, чтобы области стыка оказались совмещены.

Протокол сравнения описывает последовательность и критерии сравнения характеристик фрагментов, а также параметры их применения. Файл имеет следующую структуру:

```
<compare name="" description="">
  <feature1 opt="false">
    ...
  </feature1>
  <alt>
    <feature2 opt="false">
      ...
    </feature2>
    <feature3 opt="false">
      ...
    </feature3>
  </alt>
</compare>.
```

Корневой элемент `<compare>` содержит элементы описания критериев сравнения ха-

рактических (<feature>) в порядке их выполнения, при этом атрибут `opt` определяет опциональность или обязательность выполнения данного критерия при переходе к дальнейшим этапам обработки. Элементы <alt> определяют группу альтернативного выбора, в рамках которой для перехода к последующим этапам требуется выполнение любого из входящих в нее критериев [13]. Внутренняя структура элементов <feature> определяется соответствующими модулями сравнения.

Изображения результирующих композиций сохраняются в форматах BMP, TIFF или JPG.

5. Особенности реализации

Компоненты комплекса разработаны на языке C# с использованием платформы .NET Framework 4.0 в среде разработки Microsoft Visual Studio 2010. Программы выполнены в виде приложений архитектуры x86 и x64 с графическим интерфейсом пользователя на основе технологии Windows Forms.

Все программы используют многопоточную организацию вычислений. Взаимодействие между программами осуществляется только через файловую систему.

Масштабирование комплекса достигается за счет одновременной работы нескольких экземпляров приложений на различных компьютерах, объединенных в локальную сеть. При работе с программой описания фрагментов каждый пользователь выполняет описание своей порции фрагментов, сохраняя их в общий сетевой каталог. Программа поиска стыков также может быть запущена на разных компьютерах, при этом для каждого экземпляра программы имеется возможность задать свой диапазон перебора фрагментов. В дальнейшем предполагается разделить ППС на два отдельных приложения так, чтобы с помощью единого графического интерфейса можно было управлять несколькими модулями вычислений.

Алгоритмы описания и сравнения характеристик фрагментов реализованы в виде отдельных проектов, которые после компиляции формируют DLL-библиотеки со стандартизированным интерфейсом. Это позволяет упростить поддержку и обновление программного обеспечения, поскольку модули описания и сравнения характеристик являются независимыми от общей логики работы приложений.

Загрузка данных модулей реализована с помощью механизма позднего связывания (рефлексии), что позволяет легко вносить изменения в конфигурацию программы за счет их замены или добавления. Требуется лишь внести соответствующие изменения в настройки программы и протокол сравнения.

Предусмотрено ведение текстовых логов работы программ, в которые заносится информация о выполняемых операциях.

6. Достигнутые результаты

Разработка комплекса предполагает итеративный подход, в ходе которого проводится постепенная доработка и расширение его функциональности по результатам нескольких этапов тестирования и практических экспериментов.

В настоящее время завершен первый этап данного процесса, предполагающий реализацию базового функционала программ, проверку общей работоспособности комплекса и основных элементов технологии реконструкции. Отладку и тестирование проводили на фрагментах керамических изделий (см. рис. 3), фресок (рис. 5) и бумажных документов (рис. 6) общим числом около 250 штук. Изображения керамических изделий были получены путем фотографирования, а фресок и документов — с помощью планшетного сканера. Для всех материалов была использована единая последовательность сравнения характеристик, различия состояли в параметрах настройки конкретных методов описания, критериев сравнения и разрешении исходных изображений.

Вначале была проведена классификация фрагментов по их толщине, типу материала

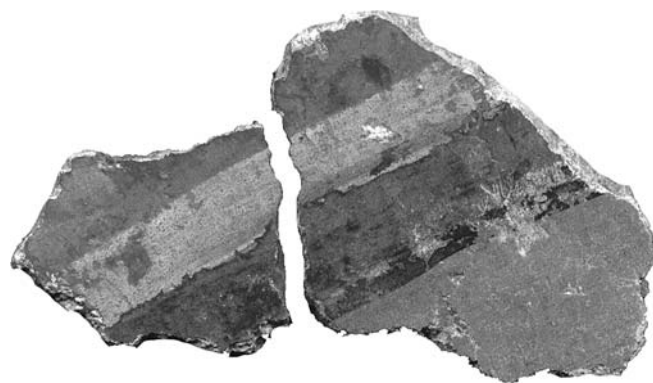


Рис. 5. Фрагменты фресок с ярко выраженным сочетанием цветовых и геометрических характеристик контура в месте разлома

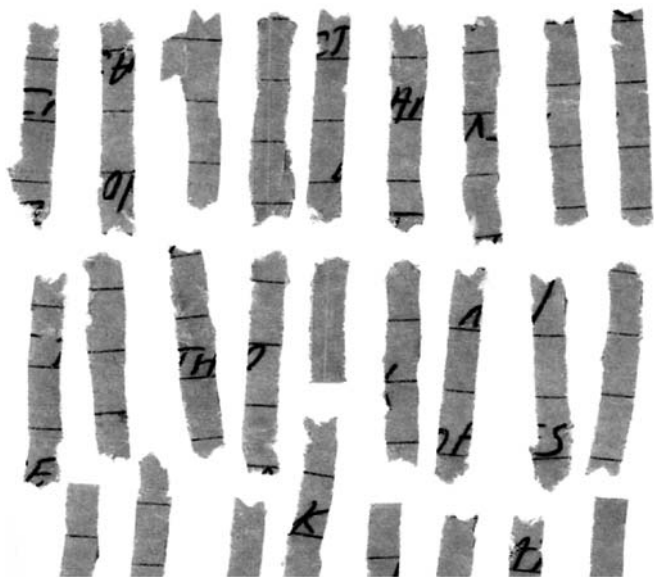


Рис. 6. Фрагменты шредерованных бумажных документов представляют сложность для реконструкции ввиду малой степени уникальности формы

и цвету, после чего выполнено грубое сравнение контуров и локализация стыков методом динамической модификации контура [12]. Найденные стыки с более высокой степенью детализации подтверждались методом многооткликовой статистической модели [11], после чего выполнялось сравнение векторов (направлений строк текста, мазков кисти, расположения геометрических линий). Сравнение цветовой палитры фрагментов использовалось как опциональный критерий для ранжирования идентифицированных стыков в процессе

последующего ручного утверждения. В завершение выполнялись синтез композиции и редактирование результатов.

В ходе отладки и пробных экспериментов работоспособность программ и элементов технологии была успешно подтверждена, что позволяет приступить к их дальнейшей доработке в целях создания полнофункциональной версии продукта и ее последующей апробации с участием специалистов в области реставрации и криминалистики. Достигнуты приемлемые показатели качества реконструкции отдельных примеров, в частности, для объекта, представленного на рис. 7 (керамическая тарелка), из 25 стыков было обнаружено 17 (из них один ложный), что в конечном итоге позволило корректно совместить 12 фрагментов из 13 и восстановить 91 % общей площади объекта. Доля перекрытий при этом не превысила 6 %. Для фресковых фрагментов, как правило, удается обнаружить не более 20 % стыков, что объясняется сильными повреждениями их лицевой поверхности. По мере дальнейших экспериментов будет проводиться доработка программного обеспечения и общей методики реконструкции для повышения ее эффективности и результативности.

Практические особенности сборки различных материалов, включая настройки алгоритмов описания и сравнения характеристик, а также количественные и качественные оценки результатов являются темой отдельных исследований и будут рассмотрены автором в следующих публикациях.



Рис. 7. Общий вид главного окна программы визуализации композиции

В настоящее время создана версия программного комплекса, реализующая предложенную архитектуру, логику работы и механизмы взаимодействия компонентов с учетом выбранных аппаратных и программных средств. Комплекс позволяет выполнять описание фрагментов, поиск стыков и визуализацию результирующей композиции с учетом базового набора обрабатываемых характеристик (контуры, атрибуты, цветовая палитра и векторы) на ограниченных наборах исходных данных.

Проведенные пробные эксперименты по реконструкции керамических фрагментов, фресковых композиций и бумажных документов показали принципиальную работоспособность как самого программного обеспечения, так и используемых в нем теоретических разработок, ранее предложенных автором. Сформулированы задачи по дальнейшему развитию продукта.

Ведется доработка комплекса в части включения методов работы с текстурными характеристиками, проверки совместимости фрагментов на глобальном уровне композиции, оптимизации рабочего процесса. В рамках создания полнофункциональной версии продукта будет проведена доработка пользовательского интерфейса, вычислительная оптимизация компонентов комплекса и уточнение отдельных алгоритмов.

Достигнуты договоренности об апробации программного обеспечения на базе Центра реставрации монументальной живописи Новгородского музея-заповедника. Подана заявка на государственную регистрацию программы визуализации композиции под названием "Программа визуализации фрагментированного изображения" и ведется подготовка документации для регистрации остальных компонентов комплекса.

Работа выполнена при поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере (Фонд содействия инновациям).

1. Willis A. R., Cooper D. B. Computational reconstruction of ancient artifacts // IEEE Signal Processing Magazine. 2008. Vol. 25, N. 4. P. 65–83.
2. Leitao H. C., Stolfi J. Multiscale method for the reassembly of two-dimensional fragmented objects // IEEE Trans. on Pattern Analysis and Machine Intelligence. 2002. Vol. 24, N. 9. P. 1239–1251.
3. Deever A., Gallagher A. Semi-automatic assembly of real cross-cut shredded documents // Proceedings 19th IEEE International Conference on Image Processing, ICIP 2012. 2012. P. 233–236.
4. Shengjiao C., Hairong L., Shuicheng Y. Automated assembly of shredded pieces from multiple photos // Proceedings of IEEE International Conference on Multimedia & Expo (ICME). 2010.
5. Amigoni F., Gazzani S., Podico S. A method for reassembling fragments in image // Proceedings of International conference on image processing ICIP 2003. 2003. Vol. 2, N. 3. P. 581–584.
6. Papaodysseus C., Panagopoulos T. Contour-shape based reconstruction of fragmented 1600 B. C. wall paintings // IEEE Transactions on Signal Processing. 2002. Vol. 50, N. 6. P. 1277–1288.
7. Kong W., Kimia B. On solving 2D and 3D puzzles under curve matching // Conference on Computer Vision and Pattern Recognition. 2001. Vol. 2, N. 10. P. 583–590.
8. Nickolay B., Schneider J. Automated Virtual Reconstruction // Fraunhofer IPK. URL: <http://www.ipk.fraunhofer.de/en/divisions/automation-technology/departments/security-technology/core-rd-activities-of-the-security-technology-department/virtual-reconstruction/automated-virtual-reconstruction-of-ripped-stasi-files/> (дата обращения: 02.02.2018).
9. Анисимова Т. И., Зотов А. В., Поневаж А. В. Использование компьютерной технологии при реставрации живописи XIV в. церкви Успения на Волотовом поле // URL: <http://art-con.ru/node/477> (дата обращения: 02.02.2018).
10. Наумов А. С., Луций С. А. Формирование дескриптора контурной линии в задаче синтеза изображения из фрагментов произвольной формы // Вестник НовГУ, Сер.: Естеств. и техн. науки. 2012. № 68. С. 68–74.
11. Наумов А. С. Стыковка фрагментов изображения по многооткликовой модели контурной линии // Вестник НовГУ, Сер.: Техн. Науки. 2015. № 7 (90). С. 11–14.
12. Наумов А. С. Стыковка фрагментов изображения по геометрическому и цветовому контурам // Вестник компьютерных и информационных технологий. 2013. № 9. С. 16–21.
13. Наумов А. С. Метод автоматизированной реконструкции фрагментированных изображений на основе совокупности признаков фрагментов // Информационные технологии. 2016. Том 22, № 1. С. 48–53.

Fragmented Image Reconstruction Software Development

Article deals with the technical aspects of the software complex development, which is intended for an automated fragmented image reassembling. This software could be used for a flat objects reconstruction with two-dimensional digital images in forensics, archeology, restoration and archive science. Complex is based on previous author's research and theoretical results. Software architecture features, work logic and program interaction mechanisms of the software are described in the article.

Software complex consist of three separate programs, developed with C# language for the .NET 4.0 platform. Fragment description program performs image preparation and feature description. Match searching program performs matching pairs searching and joint localization. Composition visualizer constructs final reassembled images of the composition.

Software architecture provides separation of the basic logic and algorithms for fragment features description and comparing, so that particular feature set could be changed, expanded or reduced for the current application without basic logic change. Data storage formats for a fragment and match descriptors are based on the XML documents. Intermediate images are stored in the ARGB BMP format files.

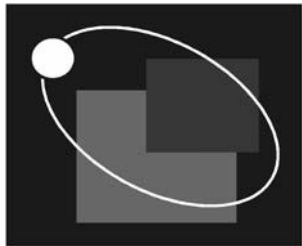
Match searching is based on the geometrical and color contours, color palette, texture, other features of the surface and user-defined attributes. Software implements some commonly known feature description methods as well as previously developed by the author, such as contour description method, based on multi-response statistical models, and 3-stage multi-feature match searching algorithm. Examples of ceramic fragments reassembling and some other practical results are presented.

Keywords: software complex, image reconstruction, fragment matching, joint searching, composition, contour, attribute, visualization, XML format

DOI: 10.17587/it.24.487-495

References

1. Willis A. R., Cooper D. B. Computational reconstruction of ancient artifacts, *IEEE Signal Processing Magazine*, 2008. vol. 25, no. 4, pp. 65–83.
2. Leitao H. C., Stolfi J. Multiscale method for the reassembly of two-dimensional fragmented objects, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002, vol. 24, no. 9, pp. 1239–1251.
3. Deever A., Gallagher A. Semi-automatic assembly of real cross-cut shredded documents, *Proceedings 19th IEEE International Conference on Image Processing, ICIP 2012*, 2012, pp. 233–236.
4. Shengjiao C., Hairong L., Shuicheng Y. Automated assembly of shredded pieces from multiple photos, *IEEE International Conference on Multimedia & Expo (ICME)*, 2010. Proceedings.
5. Amigoni F. Gazzani S., Podico S. A method for reassembling fragments in image, *International conference on image processing ICIP 2003. Proceedings*, 2003, vol. 2, no. 3, pp. 581–584.
6. Papaodysseus C., Panagopoulos T. Contour-shape based reconstruction of fragmented 1600 B. C. wall paintings, *IEEE Transactions on Signal Processing*, 2002, vol. 50, no. 6, pp. 1277–1288.
7. Kong W., Kimia B. On solving 2D and 3D puzzles under curve matching, *Conference on Computer Vision and Pattern Recognition*, 2001, vol. 2, no. 10, pp. 583–590.
8. Nickolay B., Schneider J. Automated Virtual Reconstruction, *Fraunhofer IPK*. URL: <http://www.ipk.fraunhofer.de/en/divisions/automation-technology/departments/security-technology/core-rd-activities-of-the-security-technology-department/virtual-reconstruction/automated-virtual-reconstruction-of-ripped-stasi-files/> (date of access: 02.02.2018).
9. Anisimova T. I., Zotov A. V., Ponevazh A. V. *Ispol'zovanie komp'yuternoy tehnologii pri restavracii zhivopisi XIV v. cerkvi Uspenija na Volotovom pole* (Using computer technology for XIV century painting restoration of the "Uspenija na Volotovom pole" church), URL: <http://art-con.ru/node/477> (date of access: 02.02.2018) (in Russian).
10. Naumov A. S., Lucij S. A. Formirovanie deskriptora konturnoj linii v zadache sinteza izobrazhenija iz fragmentov proizvol'noj formy (Contour descriptor forming for the arbitrary shaped fragments image synthesis problem), *Vestnik NovGU, ser.: Estestv. i tehn. nauki*, 2012, no. 68, pp. 68–74 (in Russian)
11. Naumov A. C. Stykovka fragmentov izobrazhenija po mnogootklikovoj modeli konturnoj linii (Image fragment matching using multiresponse contour line model), *Vestnik NovGU, Ser.: Tehn. Nauki*, 2015, no. 7 (90), pp. 11–14 (in Russian)
12. Naumov A. S. Stykovka fragmentov izobrazhenija po geometricheskomu i cvetovomu konturam (Image fragment matching using geometrical and color contour), *Vestnik komp'yuternyh i informacionnyh tehnologij*, 2013, no. 9, pp. 16–21 (in Russian)
13. Naumov A. S. Metod avtomatizirovannoj rekonstrukcii fragmentirovannyh izobrazhenij na osnove sovokupnosti priznakov fragmentov. (Method for automated reconstruction of the fragmented images based on multiple fragment features), *Informacionnye tehnologii*, 2016, vol. 22, no. 1, pp. 48–53 (in Russian)



2—4 октября 2018 г. в Санкт-Петербурге
в ГНЦ РФ АО "Концерн "ЦНИИ "Электроприбор"
состоится



**11-я Российская
мультиконференция
ПО ПРОБЛЕМАМ УПРАВЛЕНИЯ
(РМКПУ—2018)**

Мультиконференция включает в себя три конференции, объединенные общей идеей:

- XXXI конференция памяти выдающегося конструктора гироскопических приборов Н. Н. Острякова (<http://www.elektropribor.spb.ru/ostr2018>)
- Конференция «Информационные технологии в управлении» (ИТУ-2018) (<http://www.elektropribor.spb.ru/itu2018>)
- Конференция «Управление в морских и аэрокосмических системах» (УМАС-2018) (<http://www.elektropribor.spb.ru/umas2018>)

**Конференция
«ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНИИ» (ИТУ-2018)**

НАУЧНЫЕ НАПРАВЛЕНИЯ КОНФЕРЕНЦИИ

- Интеллектуальные информационные технологии в управлении
- Системный анализ и комплексное моделирование автоматизированных информационных систем
- Мультиагентные системы, искусственный интеллект, Big Data и нейротехнологии в задачах управления
- Геоинформационные технологии в управлении
- Информационные технологии и системы в управлении производственными процессами и предприятиями
- Прикладные задачи обработки информации в управлении
- Обработка сигналов в системах управления и связи (секция памяти А. Е. Барабанова)
- Безопасность информационных технологий в управлении
- Информационные технологии человеко-машинного взаимодействия
- Проблемы инженерного образования в области управления и информационных технологий: современное содержание и технологии обучения

*Подробную информацию о конференции см. сайте:
<http://www.elektropribor.spb.ru/itu2018>*

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова.*

Корректор *З. В. Наумова.*

Сдано в набор 26.04.2018. Подписано в печать 26.06.2018. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ718. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.