

ложения” и внедрить автоматизированное компьютерное обучение и тестирование в “самое сердце” учебного процесса.

#### Заключение

Вообще, с появлением компьютеров в вузах, почти сразу, появился и интерес к автоматизированному обучению и тестированию. Это было еще во времена ЕС ЭВМ и СМ. С появлением ПЭВМ этот интерес возрос еще более. Однако компьютеров не хватало, и они направлялись для другого, более приоритетного, целевого использования. Теперь же, когда практически любой вуз имеет десятки компьютерных классов, многие (не все, конечно) по привычке по-прежнему озабочены дальнейшим наращиванием количества компьютерных классов.

Представляется более актуальным следующее:

- 1) придание идее компьютерного обучения и тестирования как бы “второго дыхания” и сосредоточения приоритетного внимания не на количественной, а на качественной составляющей – на вопросах эффективного использования ПЭВМ в учебном процессе;
- 2) в этом же плане можно утверждать, что погоня за количеством ПЭВМ, компьютерных учебных классов менее приоритетна, чем оснащение вузов компьютерами с видеокартами для высокопроизводительных расчетов – настольными суперкомпьютерами.

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Башмаков, И.А. Разработка компьютерных учебников и обучающих систем / А.И. Башмаков, И.А. Башмаков. – М.: Информационно-издательский дом «Филинь», 2003.
2. Рыбина, Г.В. Основы построения интеллектуальных систем / Г.В. Рыбина – М.: Финансы и статистика; ИНФРА-М, 2010.

Материал поступил в редакцию 24.10.12

#### POJTA P.S., KHVEDCHUK V.I., NADEINA N.G., KUZMICKIJ N.I. The use of interactive learning system of education and the control of knowledge in the university educational process

Paper is devoted the means of building a learning environment in the university-based interactive training system and the control of knowledge. The questions of the organization of the preparatory period, the development of supervisory material, proper testing. Describes the elements of a software implementation of the control of knowledge. The results of the practical test within the university.

УДК 681.3

Муравьев Г.Л., Никонюк А.Н., Хвещук В.И.

### АЛГОРИТМЫ ПОРОЖДЕНИЯ АРХИТЕКТУР СТОХАСТИЧЕСКИХ СЕТЕЙ С ЗАДАНЫМИ ХАРАКТЕРИСТИКАМИ

**Введение.** Для автоматизации тестирования моделей систем, оценки их характеристик, включая степень адекватности, для целей обучения аналитическому и имитационному моделированию систем (обеспечения стадии концептуального моделирования с имитацией систем и возможностью их мониторинга, контроля результатов разработки моделей) необходимо решать задачу автоматического формирования параметров систем заданной сложности и режима функционирования, соответствующих тестовых описаний [1–4].

Сложность “ручного” получения уникальных вариантов архитектур, потребность в формировании описаний параметров большого числа систем, учебных систем, трудности обучения на реальных системах, трудоемкость проверки корректности моделей, получения эталонных характеристик делают указанную задачу актуальной.

**Постановка задачи.** Типовая система состоит из узлов, обеспечивающих обслуживание и перемещение потоков запросов. Они поступают извне, образуя входные потоки. Система организует процессы обслуживания в соответствии со своим законом функционирования. Процессы в общем случае носят вероятностный характер. Соответственно система описывается набором множеств  $\langle X, H, C \rangle$ , где  $X$  задает параметры окружения,  $H = \langle S, F \rangle$  – параметры архитектуры системы, включая параметры структуры  $S$  и процессов  $F$ ,  $C$  – характеристики функционирования системы [1, 2].

В качестве математических моделей систем здесь будут использоваться сети массового обслуживания (СМО) [1]. Тогда параметры структуры задаются как: состав узлов  $B = \{b_i \mid i = \overline{1, N}\}$ ; матрица связности  $D = [d_{ij}]$ , где  $d_{ij} = \{0; 1\}$  в зависимости от наличия

связи  $i$ -го узла с  $j$ -м; параметры узлов  $Z = \{\bar{z}_i \mid i = \overline{1, N}\}$ , включая канальность узла  $K_i$ , быстродействие канала  $V_i$  и другие параметры. Параметры  $Q$  процессов задаются как: матрицы переходов  $\{P^{(q)} = [p^{(q)}_{ij} \mid q = \overline{1, Q}]\}$ , где значение  $p^{(q)}_{ij} = [0, 1]$  – вероятность движения запроса с выхода узла  $i$  в узел  $j$ ; параметры  $\{\bar{h}_i^{(q)} \mid q = \overline{1, Q}; i = \overline{1, N}\}$ , задающие соответственно законы поступления запросов  $\{\bar{f}_t^{(q)} \mid q = \overline{1, Q}\}$  и законы обслуживания запросов  $\{\bar{f}_{0,i}^{(q)} \mid q = \overline{1, Q}; i = \overline{1, N}\}$  в узлах сети.

Сложность сетей  $N_C$  может быть ограничена заданием требований к таким параметрам сети, как количество процессов (число типов потоков запросов на обслуживание)  $Q$ , количество обслуживающих узлов  $N$ , сложность маршрутов запросов – количество прямых и обратных связей в сети, типы распределений значений параметров и т.д. Требуемый режим функционирования может быть задан ограничениями на допустимые значения  $\{p_{ij}\}$  коэффициентов загрузки узлов сети. Дополнительный параметр – требуемое количество уникальных сетевых архитектур.

Тогда задача сводится к получению значений параметров  $X, H$  архитектур стохастических сетей, далее – сетевых архитектур (СА) при соблюдении ограничений на  $N_C$ . При этом на этапе “комбинаторного” порождения каркасов структур сетей (КС) определяются параметры  $B, D, P$ , отвечающие  $N_C$ ; на этапе аналитического

**Муравьев Геннадий Леонидович**, кандидат технических наук, доцент, профессор кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

**Никонюк Александр Николаевич**, студент пятого курса факультета электронно-информационных систем Брестского государственного технического университета.

**Хвещук Владимир Иванович**, кандидат технических наук, доцент, профессор кафедры интеллектуальных информационных технологий Брестского государственного технического университета. Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

расчета определяются параметры узлов, обеспечивающие заданный режим функционирования сети; на этапе вероятностного доопределения уточняются недостающие параметры сети.

Таким образом, получаемые спецификации сетевых архитектур должны быть корректными и уникальными, что обеспечивается следующим набором требований: управляемой сложностью; полнотой спецификаций; контролируемостью; документированностью спецификаций. Управляемая сложность СА обеспечивается использованием эмпирически и математически обоснованных правил порождения спецификаций, правилами их хранения, учета, контроля сигнатуры. Полнота спецификаций обеспечивается генерацией наборов как параметров, так и контролируемых характеристик функционирования сетей. Контролируемость описаний может быть обеспечена построением, в том числе автоматически, соответствующих результативных моделей, их имитацией и анализом получаемых характеристик. Документированность обеспечивается процедурами генерации соответствующих отчетов, содержащих спецификации СА.

**Общее описание подхода.** Генерация СА сводится к "комбинаторному" получению каркасов сетей [5], снабжаемых далее всеми необходимыми параметрами [3, 4]. Стратегия применения алгоритмов генерации СА базируется на дереве сложности архитектур, описывающем иерархию важности параметров сети по критерию достижения оптимальных результатов генерации. Здесь это структура потоков запросов (число обслуживаемых узлов, число и характер размещения вероятностных узлов), затем параметры потоков (характер распределения вероятностей - матрица вероятностей переходов запросов, коэффициенты загрузок узлов, типы законов распределения числовых параметров).

При этом для получения наибольшего числа существенно отличающихся по архитектуре сетей следует двигаться от максимально значимых уровней сложности и переходить на нижние уровни лишь при отсутствии желаемого числа "уникальных" архитектур текущего уровня. Соответственно при генерации только уникальных структур "разброс" по значениям параметров потоков не учитывается.

В основу генерации положен смешанный комбинаторно-аналитический подход с вероятностным доопределением параметров, который обеспечивает: получение на базе рекурсивного алгоритма перебора КС заданной сложности в одномерном, "сжатом" формате и отсеивание некорректных КС; получение матричного представления КС и его "оснащение" ветвлениями (прямыми и обратными связями); расчет параметров матрицы Р переходов запросов в сети; аналитический расчет интенсивностей потоков запросов, параметров узлов, декомпозицию параметров по потокам запросов, вероятностное определение недостающих параметров.

Если требуемое число вариантов сетевых архитектур не ограничивается, то подход обеспечивает получение всех возможных СА с уникальными каркасами, то есть каждой СА соответствует свой КС. Их общее число определяется количеством сочетаний  $n$  уникальных каркасов по  $Q$  потокам и составляет  $C_n^Q = n! / (Q!(n-Q)!)$ .

**Порядок порождения спецификаций.** Первоначально генерируются матрицы  $D$ , описывающие структуру сети и удовлетворяющие входным ограничениям. Для этого для фиксированного числа узлов выполняется перебор возможных вариантов их расстановок, обеспечивающих однократный проход из источника запросов сети в приемник через каждый узел. Каждый вариант расстановки дает один вариант КС, который хранится в сжатом формате  $Dc$ , отображающем матрицу  $D$  с помощью одномерных структур данных. При этом КС представляется последовательностью номеров всех однократно проходимых узлов сети  $X_i$ , что отображается размещением

$Dc = \{x_i\}_0^{N+1}$ . Для получения матричного представления КС -  $D$  используются данные о связях узлов  $d_{ij}$ .

Таким образом, генерируемые каркасы сетей должны удовлетворять условиям, сформулированным ниже, где символами  $X_i, X_{i+1}$  обозначены номера смежных узлов матрицы  $Dc$ :

$$\begin{cases} Dc = \{x_i\}_0^{N+1}, \forall i, j (i = 1, N+1; j = 1, N+1) | x_i \neq x_j; \\ x_0 = 0, x_{N+1} = N+1; \\ p_{x_i, x_{i+1}} = \text{число\_из\_диапазона}(0, 1]. \end{cases}$$

Соответственно в каждом элементе  $Dc$  запоминается номер смежного узла, а в структуре данных  $P_L$  формируется список последовательно расположенных узлов каркаса. Построение КС реализуется рекурсивным алгоритмом, приведенным ниже, где на каждом этапе рекурсивного вызова для узла каркаса  $u$  ищется номер смежного узла  $x$

ГЕНЕРИРОВАТЬ-КС ( $Dc, u$ )

ЕСЛИ не получено необходимое число КС ТО

ЦИКЛ-ПОКА не просмотрены все узлы сети ( $x = 1, N+1$ )

ЕСЛИ (СМЕЖНОСТЬ-УЗЛОВ-ПОДТВЕРЖДЕНА ( $x, u, Dc$ )) ТО

Корректировать  $Dc [u] := x$

ЕСЛИ текущий КС построен ( $y = N$ ) ТО

ЕСЛИ (КАРКАС-ВАЛИДЕН ( $Dc, 0, 0, P_L$ )) ТО

Запомнить описание  $Dc, P_L$

КОНЕЦ-ЕСЛИ

КОНЕЦ-ЕСЛИ

ГЕНЕРИРОВАТЬ-КС ( $Dc, u+1$ )

КОНЕЦ-ЕСЛИ

КОНЕЦ-ЦИКЛА

КОНЕЦ-ЕСЛИ

КОНЕЦ.

При этом проверяется корректность смежности узлов КС, которая нарушается, если узел  $x$  предшествует узлу  $u$ , если образуется петля или переход из текущей вершины в конец КС

bool СМЕЖНОСТЬ-УЗЛОВ-ПОДТВЕРЖДЕНА ( $x, u, Dc$ )

ЕСЛИ ( $(x < u)$  И ( $Dc[x] = u$ ) ИЛИ ( $(u = 0)$  И ( $x = N+1$ ))) ТО

Вернуть результат - ложь

ИНАЧЕ

Вернуть результат - истина

КОНЕЦ-ЕСЛИ

КОНЕЦ.

Каждый каркас проверяется на корректность, которая нарушается, если не выполняется условие соответствия КС, приведенное выше, что на соответствующем шаге генерации приводит к рекурсивному откату назад на предыдущий шаг формирования КС. Или, если превышено максимальное количество прямых либо обратных связей,  $(N+1) * N / 2$ .

Проверка КС производится алгоритмом обхода графа (представленного матрицей  $D$ ), где  $k$  - номер шага,  $x$  - номер вершины,  $Nv$  - список посещенных вершин

bool КАРКАС-ВАЛИДЕН ( $Dc, x, k, Nv$ )

Добавить узел  $x$  в список  $Nv$

ЕСЛИ пройдено необходимое количество шагов ( $k = N+1$ ) ТО

ЕСЛИ узлы пройдены по разу

( $Nv$  содержит  $N+1$  уникальный узел) ТО

Вернуть результат - истина

ИНАЧЕ

Вернуть результат - ложь

КОНЕЦ-ЕСЛИ

КОНЕЦ-ЕСЛИ

КАРКАС-ВАЛИДЕН ( $Dc, Dc[x], k+1, Nv$ )

КОНЕЦ.

Каркасы оснащаются заданным числом прямых и обратных связей, что реализуется добавлением к сети вероятностных узлов. Для этого:

- описание КС переводится в матричный формат  $D$ ;
- в матрице  $D$  случайным образом выбирается номер строки  $i = \text{ЧИСЛО-ИЗ-ДИАПАЗОНА}(0, N-1)$ ;
- производится поиск узла  $i$  в ранее полученном списке вершин  $P_L$ .

Соответственно номера вершин до найденной включительно дают конечные точки обратных связей, а после найденной - конечные точки прямых связей. На пересечении найденного столбца и строки проставляется единица.

Полученная матрица  $D$  является прообразом  $P$  и может содержать в строке от одной и более единиц, что указывает на ветвлевые маршруты. На ее основе генерируются значения элементов матрицы  $P$ .

Так, для переходов, помеченных в  $D$  нулем, в соответствующий элемент  $P$  записывается нулевая вероятность; остальные значения формируются с учетом нормирующего условия  $1 = \sum_{j=0}^N p_{ij}$  (где

$i = \overline{1, N}$ ) случайным образом как

$$p_{ij} = \begin{cases} \text{rand}(0, 1], & \text{если } s_{ji} = 1; \\ 0, & \text{если } s_{ji} = 0. \end{cases}$$

Аналитически определяются параметры узлов КС. В том числе:

- по среднему значению времени  $m_i$  между запросами в потоке, взятому из заданного диапазона, определяется интенсивность поступления запросов  $\lambda_0 = 1 / m_i$ ;
- в соответствии со свойством линейности сети [1] строится система уравнений  $\lambda_i = \sum_{j=0}^N \lambda_j \cdot p_{ji}$  (где  $i = \overline{1, N}$ ), решение которой дает значения интенсивностей поступления запросов в

узлы сети  $\{\lambda_i\}$ ;

- значения интенсивностей используются для определения быстроедействий каналов, для расчета значений времени  $t_j = \rho_j \kappa_j / \lambda_j = \rho_j \kappa_j / \lambda_0 \alpha_j$  обработки запросов в каналах устройств (здесь  $\alpha_j$  - коэффициент передачи соответствующего узла [1]).

Далее производится распределение потоков по узлам сети. При этом, если количество вариантов СА задано строго, то выполняется случайное распределение потоков. В противном случае перебираются все возможные случаи размещения потоков по устройствам.

Значения остальных параметров КС генерируются случайным образом из заданных диапазонов или вариантов.

При генерации СА неоднородных сетей необходимо обеспечить значение коэффициента загрузки каждого узла всеми потоками запросов в заданном диапазоне. Для этого в схему применения приведенных алгоритмов вносятся коррективы. КС генерируются для всех потоков. Для каждого узла и потока определяются конкретные (случайные) значения коэффициентов загрузки рандомизированным выбором из заданного диапазона. Для каждого потока выполняется: а) дооснащение КС прямыми и обратными связями; б) построение матрицы  $P$ , расчет интенсивностей узловых потоков; в) расчет задержек каналов.

Полученные СА на этапе генерации проверяются на корректность и неповторяемость. Дополнительная проверка производится имитационным моделированием полученных описаний и оценкой их характеристик.

**Реализация.** Подход макетировался применительно к генерации параметров сетевых архитектур (СА) для систем, описываемых в терминах сетей массового обслуживания [1-4], и может быть распространен на произвольные стохастические сетевые модели.

Для поддержки функциональности системы реализации алгоритмов в объектно-ориентированных технологиях разработаны классы: иерархия классов для автоматического построения сетевых архитектур (упрощенный фрагмент представлен в нотации UML на рис. 1) и система классов для организации хранения результатов и

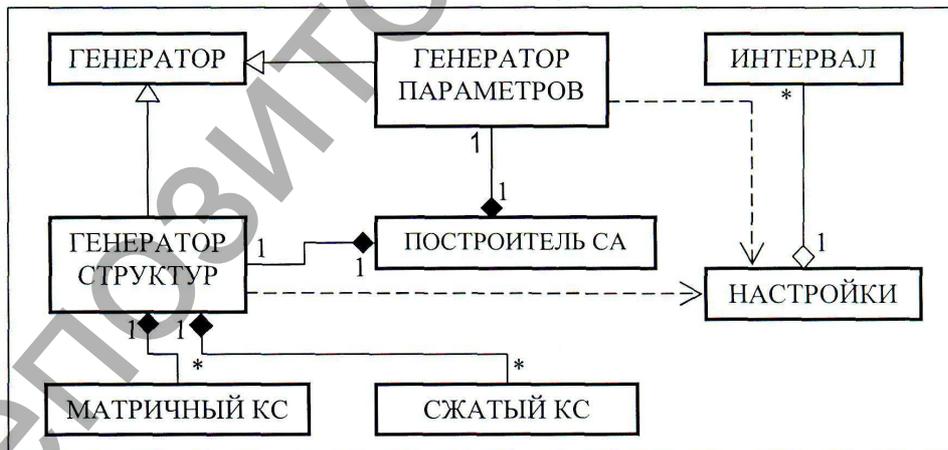


Рис. 1. Иерархия классов для поддержки генерации сетевых архитектур

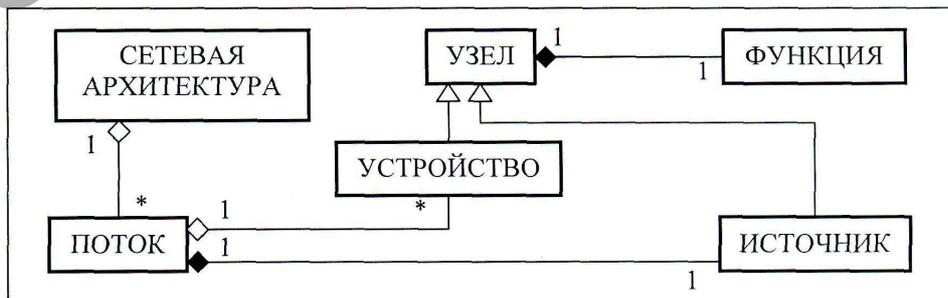


Рис. 2. Классы, обеспечивающие хранение и документирование результатов

генерации соответствующих отчетов, документов (упрощенный фрагмент представлен на рис. 2).

Из диаграммы (рис. 1) видно, что построитель СА представлен в виде двух основных частей и соответственно агрегирует классы генератора структур и генератора параметров структур. Оба класса получены специализацией наследованием общего базового класса – генератора, содержат реализации интерфейсных методов генерации КС и описаний потоков соответственно и используют члены класса настроек для инициализации и функционирования своих объектов.

Последний класс обеспечивает корректность, хранение, доступ к характеристикам (ограничениям) сетевых спецификаций. При этом большая их часть имеет интервальный тип, поддерживаемый соответствующим классом, методы которого позволяют вероятно "разыгрывать" целые и вещественные значения из заданного диапазона, выполнять пересечение диапазонов.

Класс генератора структур включает скрытые методы получения одномерных КС (сжатый формат), трансформации КС в описания с заданным числом вероятностных узлов, трансформации КС в двумерный (матричный) формат, валидации каркасов, отсеивания некорректных решений. Класс генератора параметров сети включает скрытые методы построения матриц Р и расчета недостающих параметров, включая интенсивности потоков запросов.

Вторая система классов обеспечивает хранение данных в виде описаний потоков, включающих маршруты перемещения запросов, сведения о характере потребления ресурсов, устройств, законы распределения параметров, поддерживаемые классом функция.

Система реализована на языке С++ с использованием библиотеки линейной алгебры uBLAS из собрания библиотек Boost. Пользовательский интерфейс, экспорт-загрузка XML, генерация html-отчетов реализованы на языке С++ в двух версиях: с использованием кросс-платформенного инструментария QT; в среде Microsoft Visual Studio 2008 с использованием каркаса и классов библиотеки MFC.

Система поддерживается большинством ОС типа Windows, Linux, MacOS путём перекомпиляции без изменения исходных кодов, требует около 600 Кб памяти.

Хранение результатов, описаний сетевых архитектур производится в структурированном виде в html формате в XML базе данных, что в совокупности с каскадными таблицами стилей CSS обеспечивает генерацию и форматирование отчетов любой сложности и совместимость при передаче данных другим системам обработки информации.

**Заключение.** Таким образом, сформулированы требования к спецификациям сетевых архитектур, согласованные с задачами организации имитационного моделирования. Рассмотрен подход к автоматизации их генерации с учетом требуемой сложности сетей. Приведены алгоритмы получения сетевых архитектур: каркасов сетей и расчета их параметров. Приведены результаты макетирования.

Алгоритмы обеспечивают: генерацию спецификаций сетевых архитектур сетей массового обслуживания по заданным ограничениям с сохранением результатов в XML базе данных; генерацию html-отчетов с описаниями СА, с проверочной, тестовой информацией по данным из XML базы данных.

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Ивницкий, В.А. Теория сетей массового обслуживания / В.А. Ивницкий. – М.: Физико-математическая литература, 2004. – 772 с.
2. Рыжиков, Ю.И. Имитационное моделирование. Теория и технологии / Ю.И. Рыжиков. – СПб.: КОРОНА, 2004. – 320 с.
3. Муравьев, Г.Л. Автоматизация получения тестовых описаний систем для обучения моделированию / Г.Л. Муравьев, А.Н. Никонюк // Инновационные технологии обучения физико-математическим дисциплинам: материалы III Междунар. научно-практ. конф., Мозырь, 2011. – С. 85–86.
4. Муравьев, Г.Л. Компьютерная генерация спецификаций сетевых архитектур заданной сложности / Г.Л. Муравьев, А.Н. Никонюк, В.И. Хвещук // Технологии информатизации и управления: сб. науч. ст. 2-й Междунар. научно-практ. конф. (ТИМ-2011), Минск, 2011. – С. 50–53.
5. Андерсон, Д. Дискретная математика и комбинаторика / Д. Андерсон. – СПб.: Вильямс, 2004. – 960 с.

Материал поступил в редакцию 19.10.12

#### MURAVJOV G.L., NIKONYUK A.N., KHVASHCHUK V.I. Algorithms of generation of architecture of stochastic networks with the given characteristics

Requirements are provided to specifications of network architectures taking into account tasks of the organization of simulation modeling. Algorithms of receiving frames of networks and determination of their parameters of required complexity are considered.

УДК 656.13.08

Капский Д.В., Рябчинский А.И.

### ПОВЫШЕНИЕ БЕЗОПАСНОСТИ ДОРОЖНОГО ДВИЖЕНИЯ НА ОСНОВЕ КОМПЛЕКСНОГО УЧЕТА ПОТЕРЬ

**Введение.** Проблема роста аварийности в городах характерна не только для Республики Беларусь. Так, за шесть лет в Республике Беларусь произошла 554 521 авария, в т.ч. с пострадавшими 41 914 аварий, в которых ранены 44 621 и погибли 8509 человек (рис. 1). По данным статистики, в авариях ежедневно гибнут около 4 человек и около 18 получают ранения [1, 2, 3, 4].

Не установлены зависимости аварийности от различных факторов в условиях Республики Беларусь, что вынуждает пользоваться западноевропейскими, американскими или японскими данными, которые не учитывают особенности нашей страны. В Российской Федерации ежегодно происходит около 200 тысяч аварий с пострадавшими, в которых гибнут более 26 тысяч человек и более 250 тысяч человек получают ранения. Снижения количества и тяжести последствий аварий можно достичь четким контролем и учетом скорости движения в типичных местах – очагах аварий: в зоне

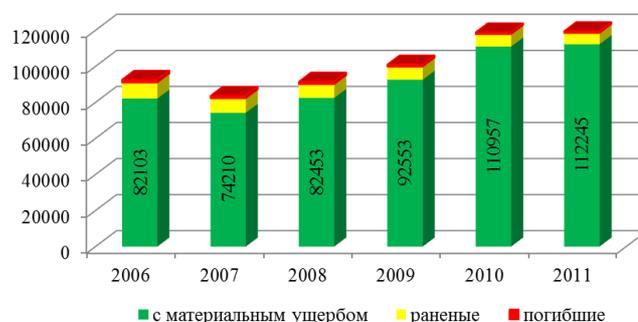


Рис. 1. Динамика аварийности в Республике Беларусь (2006–2011 гг.)

Капский Д.В., кандидат технических наук, доцент Белорусского национального технического университета.

Беларусь, БНТУ, 220013, г. Минск, пр. Независимости, 65.

Рябчинский А.И., доктор технических наук, профессор Московского автомобильно-дорожного государственного технического университета, г. Москва.