

ствующих методов балансировки загрузки вычислительной системы.

В настоящее время активно разрабатываются различные параллельные алгоритмы и соответствующее программное обеспечение, ориентированные на вычисления на графических картах. Представляется актуальной разработка такого же программного обеспечения для метода PSO.

#### Список литературы

1. Weise T. Global Optimization Algorithms Theory and Application: Ph. D. Thesis. University of Kassel, 2008.
2. Kennedy J., Eberhart R. Particle swarm optimization // Proceedings of IEEE International conference on Neural Networks. 1995. P. 1942–1948.
3. Li X., Engelbrecht A. P. Particle swarm optimization: an introduction and its recent developments // Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation. London: ACM. 2007. P. 3391–3414.
4. Mendes R., Kennedy J., Neves J. The fully informed particle swarm: Simpler, maybe better // IEEE Transactions on Evolutionary Computation. 2004. V. 8. P. 204–210.
5. Mendes R., Kennedy J., Neves J. Watch thy neighbor or how the swarm can learn from its environment // Proceedings of the Swarm Intelligence Symposium 2003. IEEE. 2003. P. 88–94.
6. Jordan J., Helwig S., Wanka R. Social interaction in particle swarm optimization, the ranked FPTPS, and adaptive multi-swarms // Proceedings of the 10<sup>th</sup> annual conference on Genetic and evolutionary computation. Atlanta, USA: ACM. 2008. P. 49–56.
7. Veeramachaneni K., Peram Th., Mohan Ch., Osadciw L. A. Optimization Using Particle Swarm with Near Neighbor Interactions // Lecture Notes Computer Science. Springer Verlag, 2003.
8. Kennedy J., Mendes R. Population structure and particle swarm performance // Proceedings of the 2002 Evolutionary Computation Congress. Washington: IEEE Computer Society. 2002. P. 1671–1676.
9. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002.
10. Kennedy J., Mendes R. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms // Systems, Man, and Cybernetics. 2006. V. 36. P. 515–519.
11. Elshamy W., Emara H. M., Bahgat A. Clubs-based Particle Swarm Optimization // Swarm Intelligence Symposium. 2007. P. 289–296.
12. Kennedy J. Stereotyping: improving particle swarm performance with cluster analysis // Proceedings of the 2000 Congress on Evolutionary Computation. 2002. V. 2. P. 1507–1512.
13. Suganthan P. N. Particle swarm optimizer with neighborhood operator // Proceedings of the 1999 Congress on Evolutionary Computation. 1999. V. 3. P. 1958–1962.
14. Richards M., Ventura D. Dynamic Sociometry in Particle Swarm Optimization // Proceedings of the Joint Conference on Information Sciences. 2003. P. 1557–1560.
15. Mohais A. S., Mendes R., Ward Ch., Posthoff Ch. Neighborhood Re-structuring in Particle Swarm Optimization // Proceedings of the Australian Conference on Artificial Intelligence. 2005. P. 776–785.
16. Dorigo M., Maniezzo V. Parallel Genetic Algorithms: Introduction and Overview of Current Research // Parallel Genetic Algorithms: Theory and Applications / Ed. by J. Stenders. Amsterdam: IOS Press, 1992.
17. Belal M., El-Ghazawi T. Parallel Models for Particle Swarm Optimizers // International Journal of Intelligent Computing and Information Sciences. 2004. V. 1. P. 100–111.
18. Shu-Chuan Chu, Jeng-Shyang Pan. Intelligent Parallel Particle Swarm Optimization Algorithms // Parallel Evolutionary Computations. Springer Berlin, Heidelberg. 2006. V. 22. P. 159–175.
19. Schutte J. F., Reinbolt J. A., Fregly B. J., Haftka R. T., George A. D. Parallel global optimization with particle swarm algorithm // Journal of Numerical Methods in Engineering. 2004. V. 61. P. 2296–2315.
20. Koh B., George A. D., Haftka R. T., Fregly B. J. Parallel asynchronous particle swarm optimization // International Journal for Numerical Methods in Engineering. 2006. V. 67 (4). P. 578–595.
21. Venter G., Sobiesczanski-sobieski J. A parallel particle swarm optimization algorithm accelerated by asynchronous evaluations // Journal of Aerospace Computing, Information, and Communication. 2006. V. 3. P. 123–137.

УДК 621.3

Н. И. Лиманова, д-р техн. наук, проф., Е. А. Мамзин, аспирант,  
Тольяттинский государственный университет, e-mail: N.Limanova@tltsu.ru

## Дискретная математическая модель детерминированного клеточного автомата и ее программная реализация

Предложены математическая модель детерминированного клеточного автомата на шестигранной пространственной сетке и ее программная реализация. Особенностью разработанного автомата является пошаговая смена конфигураций элементов, соседних с текущим, по состояниям которых определяется состояние текущего элемента. Программирование выполнено на языке Си. Визуализация результатов вычислений проводилась с помощью графической библиотеки OpenGL. Использование методов параллельного программирования и пошаговой смены конфигураций элементов позволили увеличить скорость вычислений более чем в 2 раза по сравнению с известными аналогами.

**Ключевые слова:** клеточный автомат, математическая модель, пространственная сетка, конфигурация соседних элементов, скорость вычислений

### Введение

Моделирование различных физических процессов и явлений в большинстве случаев основано на решении дифференциальных уравнений, используемых для описания этих процессов. Од-

нако аналитическое решение таких уравнений относится к довольно трудоемким задачам, к тому же его можно найти не всегда. Этим объясняется широта использования различных численных методов, одним из которых является метод клеточ-

ных автоматов [1]. В соответствии с [1] под клеточными автоматами понимаются сети элементов, изменяющих свое состояние в последовательные дискретные моменты времени по определенному закону в зависимости от того, каким было состояние самого элемента и его соседей в предыдущий дискретный момент времени. В данной статье рассматривается математическая модель детерминированного клеточного автомата, для которого определен закон перехода из предыдущего состояния в последующее. В основу реализации подавляющего большинства моделей клеточных автоматов положен шаблон, имеющий вид четырехгранной сетки. Однако в ряде случаев такой автомат не является оптимальным. Если рассматривать его с точки зрения взаимодействия соседних элементов сетки, то можно заметить, что диагональные элементы дальше удалены от центрального и должны оказывать меньшее воздействие на него, что необходимо учитывать при написании программного кода. Таким образом, получается, что для вычисления состояния текущего элемента клеточного автомата необходимо учитывать состояния восьми соседних, причем четыре диагональных клетки влияют на центральную слабее, чем клетки, соприкасающиеся с ней общими гранями. Целью данной статьи являлась разработка клеточного автомата на основе шестигранной сетки, не обладающего недостатками известных аналогов.

### Разработка дискретной математической модели детерминированного клеточного автомата

Рассмотрим задачу моделирования волновых колебаний поверхности вязкой жидкости без учета близости дна. Поставленная задача была решена на основе разработанного авторами шестигранного клеточного автомата [2] с правилами переходов, основанными на методе конечных разностей. Число ближайших соседей для любого элемента шаблона в данном случае равно шести, причем все клетки соприкасаются друг с другом только общими гранями. Данная модель позволяет получать достоверные картины волновых колебаний поверхности жидкости для различного типа волн, произвольного расположения источников колебаний, препятствий и барьеров без решения уравнения колебаний для каждого случая.

Для моделирования волновых колебаний детерминированным клеточным автоматом необходимо получить разностную схему решения широко известного уравнения колебаний [3]

$$\Delta u = \frac{1}{a^2} \frac{\partial^2 u}{\partial t^2}, \quad (1)$$

где  $\Delta$  — оператор Лапласа;  $u = u(t, r)$  — неизвестная функция;  $r \in R^n$  — пространственная переменная;  $t \in R$  — время;  $\frac{1}{a^2} = v^2$  — фазовая скорость.

Рассмотрим уравнение колебаний (1) для двумерного случая [3]:

$$\frac{\partial^2 u}{\partial t^2} = a^2 \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right). \quad (2)$$

Составим для данного уравнения разностную схему, проведя следующие замены [4]:

$$\frac{\partial^2 u_x}{\partial x^2} = \frac{u_{x+1} - 2u_x + u_{x-1}}{h^2} \quad (3)$$

и

$$\frac{\partial^2 u}{\partial t^2} = \frac{u^{t+1} - 2u^t + u^{t-1}}{\tau^2}, \quad (3^*)$$

где  $u_{i,j}^t$  — значение неизвестной функции в момент времени  $t$  в точке с координатами  $(i, j)$ . Здесь индексы вида  $(i+1, j)$  показывают, что рассматривается следующая точка по координате  $i$  относительно текущей.

Подставив замены (3) и (3\*) в волновое уравнение (2), получим:

$$\frac{u_{i,j}^{t+1} - 2u_{i,j}^t + u_{i,j}^{t-1}}{\tau^2} = a^2 \left( \frac{u_{i+1,j}^t - 2u_{i,j}^t + u_{i-1,j}^t}{h^2} + \frac{u_{i,j+1}^t - 2u_{i,j}^t + u_{i,j-1}^t}{h^2} \right), \quad (4)$$

где  $h$  — шаг пространственной сетки,  $\tau$  — шаг по времени.

Выразив из (4) следующий по времени шаг для функции  $u$  —  $u_{i,j}^{t+1}$ , получим разностную схему для решения волнового уравнения (2):

$$u_{i,j}^{t+1} = \left( \frac{a^2 \tau^2}{h^2} \right) (u_{i-1,j}^t + u_{i+1,j}^t + u_{i,j-1}^t + u_{i,j+1}^t - 4u_{i,j}^t) + 2u_{i,j}^t - u_{i,j}^{t-1}, \quad (5)$$

где  $a^2 = \frac{1}{v^2}$  — величина, обратная фазовой скорости.

Уравнение (5) представляет собой математическую модель детерминированного клеточного автомата в дискретные моменты времени. Из (5) видно, что сомножитель  $u_{i-1,j}^t + u_{i+1,j}^t + u_{i,j-1}^t + u_{i,j+1}^t - 4u_{i,j}^t$  определяет среднее значение амплитуды колебаний центральной клетки на основании известных значений амплитуд соседних, из которого вычитается текущая амплитуда центральной ячейки. Основываясь на этом, можно перейти к шестигранной сетке, переписав данное выражение в виде

$$\left( \sum_{k=1}^6 u_k^t - 6u_{i,j}^t \right), \text{ где } \sum_{k=1}^6 u_k^t \text{ — сумма значений}$$

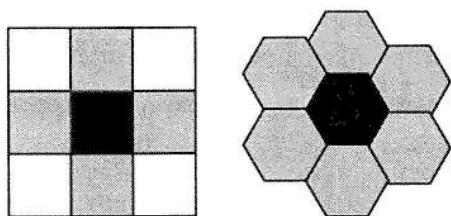


Рис. 1. Шаблоны для моделирования на основе стандартной четырехгранный сетки (слева) и предлагаемой шестиугранной (справа)

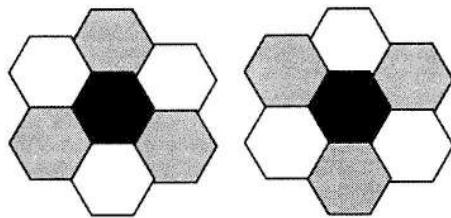


Рис. 2. Шаблон для моделирования на основе шестиугранной сетки с пошаговой сменой конфигурации учитываемых соседних элементов

амплитуд колебаний по всем соседям центральной ячейки на шестиугранной сетке.

На рис. 1 представлены рассматриваемые шаблоны для моделирования. Черным цветом показана центральная ячейка, расчет состояния которой проводится в текущий момент времени, серым цветом — соседние ячейки, по которым определяется состояние центральной.

Переход к шестиугранной сетке позволяет выровнять свойства дискретного пространства по всем направлениям, но дает ускорение только в том случае, когда для расчета требуется брать еще и диагональные элементы. Необходимо отметить, что диагональные элементы не учитываются только в случае очень грубых, приближенных расчетов. В целях повышения точности вычислений, а также для получения наглядной картины протекания процессов эти элементы необходимо учитывать. В этом случае скорость вычислений, которую дает шестиугранный клеточный автомат, увеличивается на 30–40 % по сравнению с известными аналогами на четырехгранных сетках. Для дальнейшего увеличения скорости работы программы на основе рассматриваемого клеточного автомата следует учитывать одновременно на каждом шаге вычислений параметры не шести соседних элементов, а трех:

$$u_{i,j}^{t+1} = \left( \frac{a^2 \tau^2}{h^2} \right) \left( \sum_{k=1}^3 u_k^t - 3u_{i,j}^t \right) + 2u_{i,j}^t - u_{i,j}^{t-1}. \quad (6)$$

Такой подход позволяет сократить число вычислений на одном шаге, что приводит к ускоре-

нию расчета смены состояний. Для повышения стабильности способа расчета (6) авторы предлагают использовать пошаговую смену конфигурации учитываемых в расчетах соседних элементов, т. е. на каждом шаге вычисления менять соседние ячейки так, как это показано на рис. 2.

Программа, использующая пошаговую реконфигурацию шаблона, показала увеличение скорости вычислений на 50 % по сравнению с программным кодом без ее использования.

### Применение параллельных методов вычислений в программной реализации модели

Эффективным способом продвижения в область исследования сложных математических моделей является применение параллельных методов программирования, которые позволяют разбить процесс вычислений на потоки, выполняющиеся одновременно на базе высокопроизводительных вычислительных комплексов или кластеров.

К неоспоримым достоинствам модели разработанного клеточного автомата можно отнести возможность распараллеливания процесса вычислений [5], что позволяет значительно увеличить скорость расчетов. Для данного случая наиболее оптимальным методом распараллеливания является метод декомпозиции данных. На рис. 3 представлена наглядная схема такой декомпозиции.

При таком способе распараллеливания весь массив, представляющий моделируемую плоскость, разбивается на равные элементы, которые рассылаются с сервера на прочие компьютеры в сети для обработки. После получения результатов обработки компьютеры отправляют их обратно серверу, который, в свою очередь, соединяет элементы и отображает на экране полученную кар-

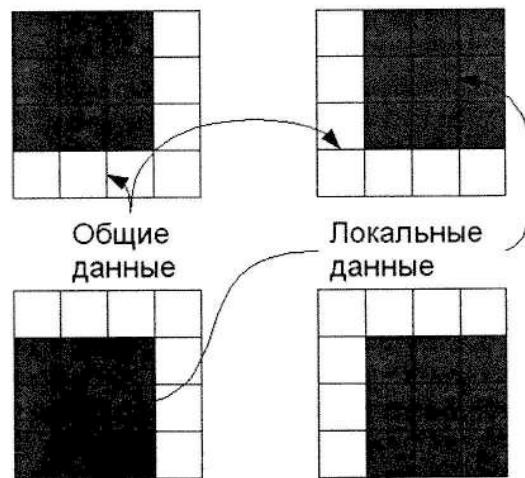


Рис. 3. Принцип декомпозиции данных

## Результаты моделирования

На основе разработанной модели детерминированного шестиугольного клеточного автомата и библиотеки распараллеливания MPI [6] была написана программа, наглядно демонстрирующая динамику распространения волн на поверхности вязкой жидкости без учета близости дна. Программирование выполнено на языке Си. Визуализация результатов вычисления осуществлялась с помощью графической библиотеки OpenGL.

На рис. 4, 5 и 6 представлены результаты работы программы. Рис. 4 наглядно демонстрирует интерференцию волн, распространяющихся от двух когерентных (а) и некогерентных (б) источников.

На рис. 5 представлена дифракция плоской волны для двух случаев: длина волны больше линейных размеров препятствия (а) и длина волны меньше линейных размеров препятствия (б).

Разработанный детерминированный шестиугольный клеточный автомат позволяет рассматривать и такие сложные для моделирования случаи, как взаимодействие сферической волны с препятствием (рис. 6, а) и распространение волны в пространстве, ограниченном с трех сторон барьерами, на выходе из которых образуется сферическая волна (рис. 6, б).

## Заключение

Переход к автомата на шестиугольной сетке позволяет увеличить скорость вычислений на 30—40 % по сравнению с известными аналогами на четырехугольных сетках.

Эффективность работы рассматриваемого клеточного автомата оценивалась по соотношению характерного времени работы его программной реализации и времени работы программы, реализующей метод конечных разностей, на 10 000 итераций.

Дальнейшему ускорению работы программы способствовало использование распараллеливания программного кода и пошаговой смены конфигураций,ываемых в расчетах соседних элементов, что в конечном итоге дало возможность наблюдать моделируемые процессы в динамике в реальном масштабе времени.

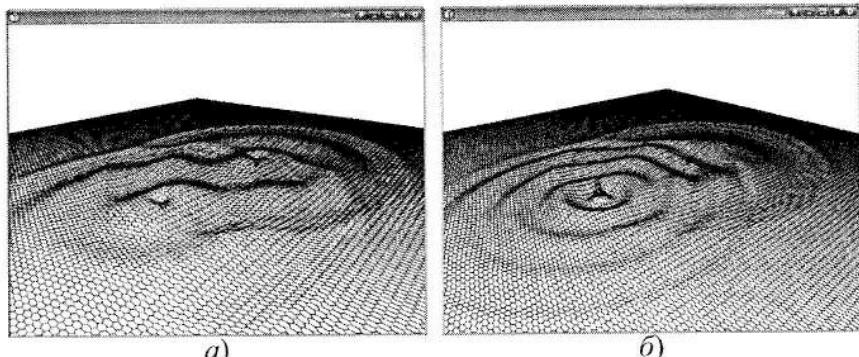


Рис. 4. Распространение волн от двух когерентных (а) и некогерентных (б) источников

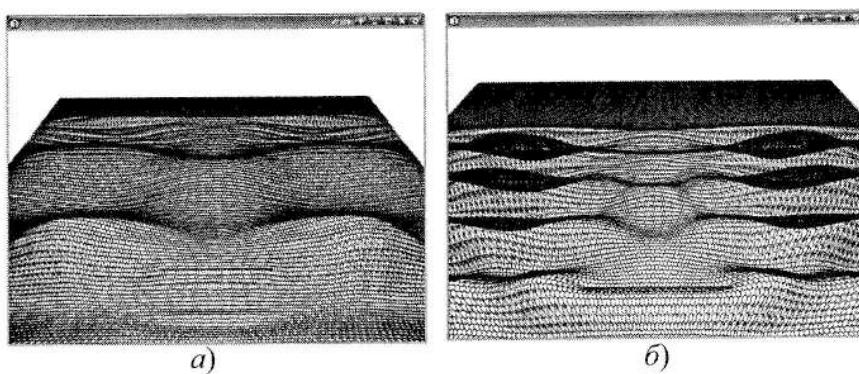


Рис. 5. Огибание препятствия плоской волной в случаях, когда длина волны больше линейных размеров препятствия (а) и когда меньше (б)

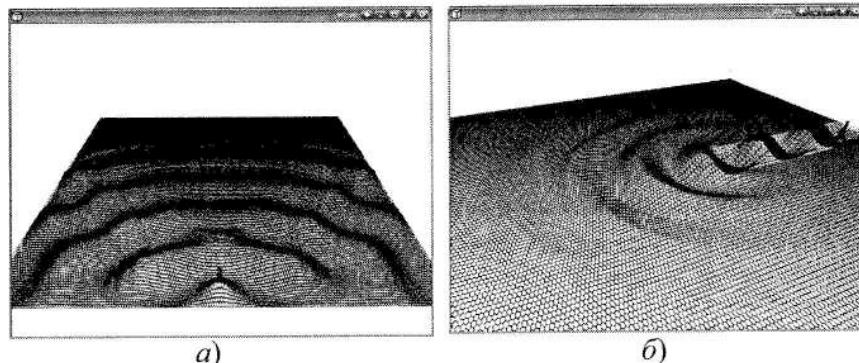


Рис. 6. Огибание сферической волной малого препятствия (а). Источник колебаний располагается в пространстве, ограниченном с трех сторон барьерами, на выходе из которых образуется сферическая волна (б)

тизу. Число и размер элементов, на которые разбивается исходный массив данных, зависит от размеров обрабатываемого массива. В процессе разработки программы учитывалось также, что пересылка данных занимает сравнительно большое время.

Распараллеливание вычислений выполнялось на компьютерном кластере Тольяттинского государственного университета. Кластер объединяет 168 компьютеров. Как показало тестирование, быстродействие системы на 60 узлах составило 30 Gflops.

Рассмотренный пример практического использования предложенной модели показывает, что детерминированный шестигранный клеточный автомат прекрасно описывает такие волновые явления, как дифракция и интерференция, позволяет в интерактивном режиме изменять параметры моделируемого пространства, т. е. задавать граничные и начальные условия, устанавливать центры источников колебания, преграды и т. д. Были реализованы вращение моделируемой области, приближение и возможности установки паузы для детального рассмотрения участка сетки. Необходимо отметить, что область практического применения предложенного автомата не ограничивается моделированием только волновых колебаний, а позволяет рассматривать широкий круг явлений и процессов в динамике. Программная реализация разработанного клеточного автомата

обладает скоростью вычислений, более чем в 2 раза превышающей известные аналоги.

#### Список литературы

1. Тoffoli Т., Марголус Н. Машины клеточных автоматов. М.: Мир, 1991. 280 с.
2. Лиманова Н. И., Мамзин Е. А. Моделирование физических явлений на основе использования клеточных автоматов // Всероссийский конгресс женщин-математиков: Матер. конф. 15–18 января 2008 г. Красноярск: РИО СФУ, 2008. С. 262–266.
3. Тихонов А. Н., Самарский А. А. Уравнения математической физики. М.: Наука, 1977. 404 с.
4. Гулин А. В., Самарский А. А. Численные методы. М.: Наука, 1989. 432 с.
5. Лиманова Н. И. Распределенные параллельные вычисления в задачах математического моделирования // Матер. Международной науч. конф. "Космос, астрономия и программирование (Лавровские чтения)" 20–22 мая 2008 г. СПб.: СПбГУ, 2008. С. 211–215.
6. Богачев К. Ю. Основы параллельного программирования. М.: БИНОМ. Лаборатория знаний, 2003. 342 с.

УДК 004.3.07

**И. В. Ковель**, аспирант,  
Кубанский государственный  
технологический университет,  
e-mail: vinitar@yandex.ru

## Вероятностный метод построения окрестности статистического глобального оптимума NPC-задач

*Рассматривается вероятностный метод получения ранжированных субоптимальных решений NPC-задач в окрестности статистического глобального оптимума. Суть метода заключается в построении решений с использованием генераторов псевдослучайных чисел, работающих по статистически обоснованной геометрической функции вероятности рангов локальных решений.*

**Ключевые слова:** NPC-задача, статистический глобальный оптимум, окрестность, ранг, локальный оптимум, задача коммивояжера, оптимизация раскроя-упаковки

### Постановка задачи

Круг рассматриваемых недетерминированных полиномиальных полных задач (NPC-задач) ограничен задачами потребления невозобновляемых ресурсов произвольной природы. К таким задачам можно отнести задачу коммивояжера, задачу оптимизации раскроя-упаковки и нестинга, задачу

чу оптимизации размещения обслуживающих центров и т. п.

Исходные данные задачи описываются в виде графов без петель и кратных ребер  $G = (V, E, d)$ , где  $V$  – множество вершин-ресурсов  $v_i$  произвольной природы,  $|V| = n$ , каждая вершина  $v_i$  характеризуется вектором параметров ресурсов  $P_i = (p_1, p_2, \dots, p_d)$ , определенных на множестве  $\mathbb{R}^+$ ,  $p_i \in \mathbb{R}^+$ ;  $d$  – размерность ресурсов, определенная на множестве  $\mathbb{Z}^+$ ;  $E$  – множество ребер  $e_{ij}$ , задающих отношение смежности произвольной природы между вершинами  $v_i$  и  $v_j$ ,  $e_{ij} = e_{ji}$ ,  $i \neq j$ ,  $|E| \leq \frac{n^2 - n}{2}$ .

Потребление ресурсов заключается в пошаговом исключении из графа  $G$  выбираемых по определенному рангу  $r_i$  ресурсов  $v_i$  и формировании ранжированной цепи  $\pi(n, k) = [r_1, r_2, \dots, r_{k-1}]$ ,  $k$  – длина цепи,  $k \leq n + 1$  [1]. Ранжированное дерево решений задается множеством ранжированных цепей  $\Pi_{n,k} = \{\pi(n, k) \mid 0 < k \leq n + 1\}$ .

Пусть вес вершины  $v$  графа  $G$  задается функцией  $q(v) = \prod_{i=1}^d p_i$ , а вес цепи  $h(\pi) = \sum_{i=1}^k q(v_i)$ . Задача состоит в нахождении подмножества субоптимальных решений  $\Pi_{n,k}^* \subset \Pi_{n,k}$ , которое содержит только те цепи  $\pi^* \in \Pi_{n,k}^*$ , у которых наи-