

Б. Ф. Мельников, Е. А. Мельникова

**ПОДХОД К ПРОГРАММИРОВАНИЮ  
НЕДЕТЕРМИНИРОВАННЫХ ИГР  
(Часть I: Описание общих эвристик)**

**Аннотация.** *Актуальность и цели.* Рассмотрены некоторые подходы, используемые авторами при программировании интеллектуальных недетерминированных игр – те подходы, которые не были описаны в наших предыдущих публикациях. Методы алгоритмизации, разработанные и реализованные авторами, работают со специально модифицированным для недетерминированных игр деревом поиска и представляют собой дополнение (а иногда – альтернативу) нейросетевым методам. Важно отметить, что разработанные нами алгоритмы находят свое применение не только непосредственно в недетерминированных играх, но и в других задачах дискретной оптимизации. *Материалы и методы.* После реализации алгоритмов статической оценки позиции (выполняемых либо стандартными подходами программирования интеллектуальных игр, либо нейросетевыми методами) окончательная (динамическая) оценка позиции вычисляется по всем детерминированным оценкам, полученным для всех возможных реализаций случайного события. Эти оценки специальным образом усредняются – и результат этого усреднения рассматривается как динамическая оценка. С физической точки зрения применяемое усреднение дает положение центра тяжести одномерной системы тел, масса которых задается специальной функцией – так называемой функцией риска. Координаты тел соответствуют детерминированным оценкам, зависящим, как и в обычном методе минимакса, только от детерминированных факторов игры. Для определения последовательности обработки вершин недетерминированного дерева перебора мы строим специальные эвристические функции (функции предпочтения), на основе которых применяется сортировка в каждом из двух таких множеств вершин. Эти функции предпочтения зависят от следующих параметров: глубины текущей вершины в дереве игры; предшествующей оценки позиции; значения уже достигнутой глубины перебора. Одновременно с построением оценочной функции с помощью эксперта авторами были рассмотрены некоторые вопросы, касающиеся алгоритмов автоматизированного построения таких функций. В качестве аппроксимации статической оценки позиции использовалась трехслойная нейросеть с обратным распространением ошибки. *Результаты.* Результатами данной работы являются не только разработанные программы для интеллектуальных недетерминированных игр, но и описание применения рассмотренных нами «игровых» подходов в различных задачах дискретной оптимизации. *Выводы.* Практические результаты программ, созданных на основе рассмотренных нами алгоритмов, показывают преимущества нашего подхода к порядку обработки вершин недетерминированного дерева перебора – по сравнению с подходами, близкими к полному перебору.

**Ключевые слова:** алгоритмизация, недетерминированные игры, модифицированное дерево поиска.

B. F. Mel'nikov, E. A. Mel'nikova

**APPROACH TO PROGRAMMING  
OF NONDETERMINISTIC GAMES  
(Part I: Description of general heuristics)**

**Аннотация.** *Background.* The article considers several approaches used in programming intelligent nondeterministic games – the approaches that have not been described in authors’ previous publications. The methods of algorithmization, developed and realized by the authors, deal with the search tree, modified specially for the nondeterministic games, and represent an addition (sometimes an alternative) to the neural network methods. It is important to note that the developed algorithms may be applied not only directly in the nondeterministic games, but in other problems of discrete optimization. *Materials and methods.* After realization of the algorithms of statistical evaluation of the position (carried out either by standard approaches to intelligent games programming, or by neural network methods) the final (dynamic) evaluation of the position is calculated by all deterministic values obtained for all possible realizations of a random event. These values are averaged by a special method, and the results of the said averaging are considered as a dynamic value. From the physical point of view the applied averaging gives the center of gravity position of one-dimensional system of bodies, the mass of which is set by a special function – the so-called risk function. The coordinates of bodies correspond to the deterministic values depending, similarly to the regular method of minimax, only on the deterministic factors of the game. To determine the sequence of nondeterministic searching tree tops’ processing the authors build special heuristic functions (preference functions), on the basis of which the researchers apply sorting in each of two sets of tops. The said preference functions depend on the following parameters: depth of the current top of the game tree; previous evaluation of the position; value of the reached depth of searching. Simultaneously with the building of the evaluation function with the help of the expert the authors considered several problems relating to the algorithms of automated building of such functions. The three-layer neural network with reverse error distribution was used as the approximation of the static evaluation of the position. *Results.* The results of the study are not just the developed programs for intelligent nondeterministic games, but also the application description of the considered “game” approaches in various problems of discrete optimization. *Conclusions.* Practical results of the programs, created on the basis of the considered algorithms, show the advantages of the authors’ approach to the sequence of nondeterministic searching tree tops’ processing in comparison with the approaches similar to the exhaustive search.

**Key words:** algorithmics, nondeterministic games, search tree.

### **Введение**

Рассмотрены некоторые подходы, используемые авторами при программировании интеллектуальных недетерминированных игр<sup>1</sup> – те подходы, которые не были описаны в предыдущих публикациях авторов [1, 2]. В недетерминированных играх присутствуют случайные факторы, существенно ограничивающие использование методов и подходов, традиционно применяемых в программировании игр детерминированных. Методы алгоритмизации, разработанные и реализованные авторами, работают со специально модифицированным для недетерминированных игр деревом поиска и представляют собой дополнение (а иногда – альтернативу) нейросетевым методам.

Основным предметом изучения является описание общих подходов, используемых авторами при программировании недетерминированных игр – прежде всего описание некоторых общих эвристик. Эти эвристики реализованы и используются авторами на практике. Как обычно бывает в по-

---

<sup>1</sup> К таким играм относятся, например, большинство карточных игр, нарды и др.

добных случаях, большинство этих эвристик успешно применяется и при решении других задач – например, дискретной оптимизации [3–7]. Некоторые из таких задач кратко описаны в заключении (во второй части данной статьи).

Подход авторов к программированию недетерминированных игр [1, 2] можно считать обобщением классических подходов для различных таких игр ([8] и др.), а также так называемых алгоритмов ожидаемого максимума [9] и \* – минимакса [10]. Авторами была также представлена еще более сложная модель [2]. При этом стоит отметить, что далее мы *не* рассматриваем:

– подходы, связанные с методом Монте-Карло [11], хотя в реальных программах авторов эти методы применяются: например, в бэкгаммоне [1, 2] подобный метод очень удачно работает при обработке ситуаций, в которых велика вероятность двойного или тройного выигрыша одной из сторон – “gammon” и “backgammon”;

– а также подходы, ориентированные на правила конкретных игр (“rule-based approaches” [12] и др.).

Итак, в данной статье мы рассматриваем подходы, которые могут применяться *к любой* из недетерминированных игр.

Приведем краткое содержание обеих частей статьи по разделам (нумерация разделов и рисунков – сквозная). В разделе 1 содержится краткое описание дерева перебора для случая недетерминированных игр, т.е. очень краткое изложение предыдущих публикаций авторов [1, 2]. Раздел 2 содержит описание общих эвристических алгоритмов упорядочивания вершин дерева поиска в недетерминированных играх. Такое упорядочивание позволяет осуществлять обработку узлов дерева в реальном времени, что позволяет с большой вероятностью находить почти оптимальную оценку исследуемой игровой позиции. Для практического применения описываемых эвристик в игровых программах необходима оценочная функция.

Вторая часть статьи: в разделе 3 будут описаны вспомогательные эвристики для *динамического построения* дерева перебора. В разделе 4 будут описаны способы построения *и самообучения* оценочной функции, при этом рассматривается возможность *одновременного* применения недетерминированного дерева перебора и нейронной сети. В разделе 5 будут приведены примеры работы описанных алгоритмов для конкретных оценок позиций различных уровней, соответствующих вершинам дерева поиска недетерминированной игры. При этом всюду в разделе 5 значения оценок выбираются таким образом, чтобы примеры, несмотря на их малый объем, были бы интересными. Иногда эти значения выбираются случайно, но, конечно, в конкретных играх при построении оценочной функции должна использоваться помощь эксперта – сильного и опытного игрока.

Стоит также отметить, что случаи использования нами генетических алгоритмов не вынесены в отдельный раздел, а приводятся непосредственно в тексте статьи.

## 1. «Недетерминированное» дерево перебора

Как уже было сказано во введении, при программировании недетерминированных игр неприменимы стандартные методы и подходы, развитые для детерминированных игр. Наверное, именно поэтому практически все известные

авторам сильные программы для недетерминированных игр используют в качестве оценочной функции нейронные сети ([13] и др.). Напротив, наш подход использует авторскую модификацию классического минимаксного метода.<sup>1</sup>

Итак, что является главным различием между короткими нардами (и другими недетерминированными играми) и шахматами (и другими детерминированными)? Почему мы не можем применить классический минимаксный метод к первым? Различие в том, что дерево игры, построенное для коротких нардов, содержит не только вершины, в которых игрок выбирает ход, но также вершины, где оба игрока ждут реализации некоторого случайного события. Именно поэтому минимаксный метод должен быть обобщен на недетерминированные игры [1, 2] – здесь мы кратко повторим это обобщение.

В недетерминированных играх имеется следующая последовательность шагов: реализация некоторого случайного события, ход игрока, снова реализация случайного события, ход другого игрока и т.д. В результате в дереве игры между уровнями, на которых ход принадлежит одному из игроков, образуется дополнительный уровень, соответствующий реализациям случайного события. Пусть нам известна статическая оценка позиции, т.е., как и в детерминированных играх, пусть известны оценки позиций следующего уровня. Тогда «временным удалением недетерминированности» мы получаем предварительные оценки позиций дерева игры. Для этого мы предполагаем, что уже совершилась некоторая реализация случайного события – и можно вычислить динамическую оценку рассматриваемой позиции подобно тому, как это делается в классическом методе минимакса. Затем вычисляется динамическая оценка для следующей предполагаемой реализации случайного события и т.д. для всех возможных реализаций.

Окончательная *динамическая* оценка позиции вычисляется по всем детерминированным оценкам, полученным для всех возможных реализаций случайного события. Эти оценки специальным образом усредняются – и результат этого усреднения рассматривается как динамическая оценка. С физической точки зрения применяемое усреднение дает положение центра тяжести одномерной системы тел, масса которых задается специальной функцией – так называемой функцией риска. Координаты тел соответствуют детерминированным оценкам, зависящим, как и в методе минимакса, только от детерминированных факторов игры.

В качестве иллюстрации вышесказанного на рис. 1 приведен простой пример такого дерева недетерминированной игры. Случайные события взяты из коротких нардов (бэкгеммона); стрелки помечены возможными результатами бросков пары игральных костей. Для прочих недетерминированных игр стрелки помечаются соответствующей реализацией используемого в игре случайного события. Все остальные комментарии относятся к оценкам вершин.

Зная окончательные оценки и используя общий метод минимакса, мы получаем оценки малых вершин, появившихся в результате удаления недетерминизма (мы знаем все значения, выпадающие на игральные кости). Также мы используем некоторые дополнительные условия относительно оценок вершин, пропущенных под точками: мы полагаем, что эти значения лежат между границами, задаваемыми вершинами слева и справа от этих точек.

---

<sup>1</sup> Нейросетевые алгоритмы (часть II, раздел 5) нами используются только в качестве вспомогательного аппарата.

Оценки больших вершин (не листьев – отметим, что в нашем простом примере есть только одна большая вершина, корень) считаются как усредненное значение оценок малых вершин. На рис. 1 в качестве усредненного значения взято среднее арифметическое, но на практике эта часть вычислений часто значительно сложнее.

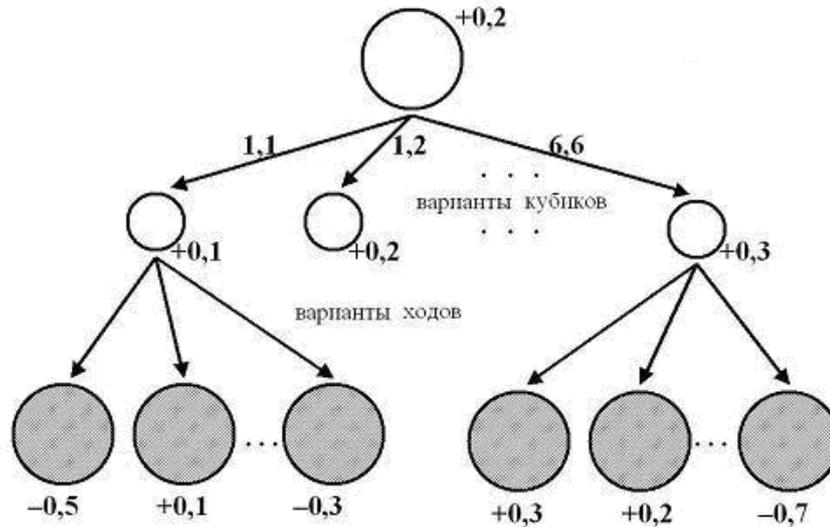


Рис. 1. Дерево недетерминированной игры

Среднее значение нами вычисляется как взвешенное среднее значение с использованием вышеупомянутой функции риска. Функции риска хорошо моделируются с помощью квадратичных полиномов. Следует также отметить, что авторами рассматриваются *динамические* функции риска, т.е. выбор конкретной функции для конечной оценки зависит от предшествующей оценки позиции [1, 2]. Избегая ряда технических деталей, кратко опишем эти функции и отметим при этом, что на практике их реализация представляет существенную часть работы программиста. Например, если оценка предшествующей позиции близка к +1 (ожидается «наша» победа), то, получая усреднение, мы приписываем более низкие веса к тем ожидаемым исходам случайного события, которые выгодны для нас, а более высокие веса приписываем к невыгодным исходам. Если же «мы» близки к поражению, то наоборот – мы уделяем больше внимания удачным исходам и меньше внимания уделяем неудачным. А если предшествующая оценка близка к нулю, то мы снова приписываем более низкие веса к выгодным исходам (но не в такой степени, как в случае ожидаемой победы); сказанное, по-видимому, соответствует повседневной привычке людей быть «немного пессимистами».

В следующем разделе мы переходим к некоторым специальным эвристикам, используемым для работы с деревом недетерминированных игр.

### 3. Общие эвристики в недетерминированных играх

В этом разделе мы рассматриваем некоторые общие эвристические алгоритмы упорядочивания вершин дерева поиска в недетерминированных играх; сортировка задает порядок обработки вершин дерева, позволяющий в ре-

альном времени и с высокой вероятностью находить почти оптимальную оценку исследуемой игровой позиции.

Как это было сказано ранее, дерево недетерминированной игры содержит вершины двух различных типов. Здесь мы имеем в виду не «черные» и «белые» вершины, а «детерминированные» и «недетерминированные» (один из вариантов английской терминологии – “simple nodes” и “chance nodes”). Отметим по этому поводу еще раз, что модель дерева, предложенная нами в 1998 г. [1], является более мощной, чем модель, рассмотренная С. Расселом и П. Норвигом *позднее* [8] – в 2002 г. (на языке оригинала). В недетерминированной вершине рассматриваются все возможные реализации случайного события. А в детерминированной вершине мы предполагаем, что уже произошла и известна некоторая конкретная реализация случайного события; эти вершины изображаются на рисунках малыми кругами. Важно отметить, что в случае большей (чем на рис. 1) глубины дерева поиска мы, кроме белой, должны рассматривать еще и черные недетерминированные («малые черные») вершины. Модель, описанная в наших предыдущих публикациях, предполагает отсутствие такой вершины в дереве поиска коротких нард из-за невозможности полного перебора для глубины, превосходящей 1. Однако в этом разделе мы описываем поиск для большей глубины. Малые черные вершины возможны и при программировании коротких нард, в которых относительно много вариантов перебора. Однако при этом мы не рассматриваем полный перебор, а выбор вершин для увеличения глубины перебора производится на основе не только статической ([14, 15] и др.), но и *динамической* оценки позиции.

Итак, в дереве перебора имеются вершины двух различных типов, и мы должны обрабатывать их двумя различными алгоритмами. Самый важный вопрос, на который мы должны ответить: «В какой последовательности должны быть обработаны вершины?». Отсутствие такой последовательности (т.е. если применяется простая обработка вершин сверху вниз) – это важная причина упомянутой невозможности (полного) перебора в реальных игровых программах – для глубины, превосходящей 1.

Для определения такой последовательности мы строим специальные эвристические функции (функции предпочтения), на основе которых применяется сортировка в каждом из двух таких множеств вершин. При этом мы заранее отметим некоторые обстоятельства. Во-первых, программная реализация таких множеств вершин не требует использования сортированных списков или массивов. Во-вторых, точный вид тех функций выбирается путем самообучения с использованием генетических алгоритмов. В-третьих, простой пример подобной обработки списков будет приведен в части II.

Обе вышеупомянутые эвристические функции предпочтения зависят от следующих параметров:

– первый параметр – глубина текущей вершины в дереве игры (для нее удобно вести нумерацию начиная с 0, причем отдельно для детерминированных и недетерминированных вершин);

– вторые параметры различны, но в обоих случаях связаны с предшествующей оценкой позиции;

– третий параметр требуется только для вычисления функции для недетерминированной вершины; в таком качестве используется значение уже достигнутой глубины перебора.

Прежде чем описать вторые параметры функций предпочтения, важно пояснить термин «предшествующая оценка позиции». Это понятие используется и в программировании детерминированных игр ([14, 15] и др.); оно связано с тем, что с увеличением глубины поиска в детерминированном дереве оценка некоторой вершины изменяется. Но еще важнее это понятие в случае недетерминированных игр. В них для детерминированных вершин оценка может быть найдена с помощью общего (обычного) метода минимакса. А оценка некоторой недетерминированной вершины (усредненная с помощью вышеупомянутых функций риска [2]) может измениться, если глубина перебора для какой-нибудь дочерней вершины увеличивается; это происходит, например, в случае обработки такой дочерней вершины. В случае недетерминированных игр хорошие практические результаты получаются в следующем случае. Мы запоминаем все предшествующие оценки (назовем их  $E_0, E_1, \dots, E_k$ , где  $k$  – глубина перебора) и считаем итоговую оценку  $E$  по формуле

$$E = (c_0 \cdot E_0 + c_1 \cdot E_1 + \dots + c_k \cdot E_k) / (c_0 + c_1 + \dots + c_k),$$

где  $c_0, c_1, \dots, c_k$  – некоторые *подбираемые* коэффициенты (еще раз мы отметим, что оценки  $E_0, E_1, \dots, E_k$  являются статическими, а динамические оценки получаются способом, кратко описанным в разд. 1 и более подробно рассмотренным в [5, 6].) На практике хорошие результаты достигаются, когда для коэффициентов  $c_i$  используются формулы

$$c_0 = 1, c_i = p \cdot r \cdot c_{i-1} \text{ для } 1 \leq i \leq k, r > 1,$$

где  $p$  – переменная, представляющая собой отношение оценки к уже полученной оценке предыдущего уровня  $E_{i-1}$ ;  $r$  – константа, настраиваемая генетическими алгоритмами<sup>1</sup>.

Следующая модификация этих формул дает хорошие практические результаты. Мы начинаем суммирование не с индекса 0 ( $E_0$  соответствует глубине поиска 0, т.е. статической оценке позиции), а с индекса – 1. При этом мы полагаем  $c_{-1} = 1$ , после чего рекуррентно строим формулы для  $c_i$  начиная с  $i = 0$ . Важно отметить, что  $E_{-1}$  является так называемой «оценкой ситуации», которая означает оценку, вычисленную на основании ранее принятого решения о принадлежности рассматриваемой позиции к одному из нескольких заранее определенных множеств ситуаций (позиций). Фактически в программах авторов количество таких рассматриваемых множеств ситуаций (для коротких нард и других недетерминированных игр) было между 5 и 30. Отметим также, что при рассмотрении таких множеств ситуаций способ динамического вычисления оценок позиций (разд. 2) остается неизменным. Кроме того, описанная модификация практически не нуждается в дополнительных вычислениях: в статьях, посвященных недетерминированным играм, а также другим задачам дискретной оптимизации [1–7], было показано, что, применяя генетические алгоритмы для настройки коэффициентов (при вычислении оценок позиций и в других подзадачах), нам желательно заранее связывать ситуацию с некоторым классом (классифицировать ситуации), т.е. производить вычисления, связанные с такой классификацией.

<sup>1</sup> Фактически все немного сложнее, но мы не будем рассматривать дополнительные технические детали.

Таким образом, окончательные оценки вершин (в данной формуле  $E$  без нижнего индекса) зависят от нескольких динамических оценок, соответствующих различным глубинам поиска. Имеется и такой пример использования самообучения и применения для него генетических алгоритмов: нам удастся устанавливать среднюю разность между любыми двумя оценками различных уровней, близкими к 0 (здесь имеются в виду оценки, начинающиеся со статической:  $E_0, E_1, \dots, E_k$ , но не  $E_{-1}$ ). По мнению авторов, это является некоторым аналогом несмещенной оценки в теории вероятностей.

Вернемся к рассмотрению вторых параметров функций предпочтения. В случае сортировки меньших вершин вторым параметром является абсолютное значение оценки позиции. Такой выбор (фактически – такая эвристика) отражает тот факт, что среди возможных вариантов случайных событий мы должны в первую очередь рассмотреть те, которые имеют «менее определенную» предшествующую оценку, для того чтобы сделать эту оценку «более уверенной» (если такое возможно). В случае сортировки недетерминированных («больших») вершин вторым параметром является оценка позиции. Ее знак для черных вершин удобно изменить на минус, чтобы сконструировать функцию предпочтения, монотонно возрастающую по обоим параметрам.

Рассмотрение эвристических алгоритмов для программирования интеллектуальных недетерминированных игр будет продолжено в части II настоящей статьи.

#### **Список литературы**

1. **Мельников, Б. Ф.** О выборе стратегии в недетерминированных антагонистических играх / Б. Ф. Мельников, А. Н. Радионов // Известия РАН. Программирование. – 1998. – № 5. – С. 55–62.
2. **Мельников, Б. Ф.** Эвристики в программировании недетерминированных игр / Б. Ф. Мельников // Известия РАН. Программирование. – 2001. – № 5. – С. 63–80.
3. **Мельников, Б. Ф.** Мультиэвристический подход к задачам дискретной оптимизации / Б. Ф. Мельников // Кибернетика и системный анализ. НАН Украины. – 2006. – № 3. – С. 32–42.
4. **Melnikov, V.** Discrete optimization problems – some new heuristic approaches / V. Melnikov // 8 th International Conference on High Performance Computing and Grid in Asia Pacific Region, IEEE Computer Society Press Ed. – Beijing, China (HPC-Asia), 2005. – P. 73–80.
5. **Melnikov, V.** Some special heuristics for discrete optimization problems / V. Melnikov, A. Radionov, V. Gumayunov // 8 th International Conference on Enterprise Information Systems, aphos. – Cyprus, ICEIS, 2006. – P. 360–364.
6. **Мельников, Б. Ф.** Кластеризация ситуаций в алгоритмах реального времени в некоторых задачах дискретной оптимизации / Б. Ф. Мельников, Е. А. Мельникова // Известия высших учебных заведений. Поволжский регион. Сер. Естественные науки. – 2007. – № 6 (33). – С. 3–11.
7. **Баумгертнер, С. В.** Мультиэвристический подход к проблеме звездно-высотной минимизации недетерминированных конечных автоматов / С. В. Баумгертнер, Б. Ф. Мельников // Вестник Воронежского государственного университета. Сер. Системный анализ и информационные технологии. – 2010. – № 1. – С. 5–7.
8. **Рассел, С.** Искусственный интеллект – современный подход / С. Рассел, П. Норвиг. – М. ; СПб. ; Киев : Вильямс, 2006. – 1410 с.
9. **Michie, D.** Game-playing and game-learning automata / D. Michie // Advanced in Programming and Non-Numerical Computation, Pergamon Ed. – Oxford, 1966. – P. 183–200.

10. **Hauk, T.** Rediscovering \*-minimax search / T. Hauk, M. Buro, J. Schaeffer. – Computers and Games, 2004. – С. 35–50.
11. **Люгер, Д.** Искусственный интеллект – стратегии и методы решения сложных проблем / Люгер Д. – М. ; СПб. ; Киев : Вильямс, 2003. – 864 с.
12. **Billings, D.** Algorithms and assessment in computer poker / D. Billings // University of Alberta (USA), Ph. D. thesis, 2006. – 211 с.
13. **Tesauro, G.** Neurogammon wins computer Olympiad / G. Tesauro // Neural Computation. – 1989. – Vol. 1, № 3. – P. 321–32.
14. **Адельсон-Вельский, Г. М.** Программирование игр / Г. М. Адельсон-Вельский, В. Л. Арлазаров, М. В. Донской. – М. : Наука, 1978. – 255 с.
15. **Адельсон-Вельский, Г. М.** Машина играет в шахматы / Г. М. Адельсон-Вельский, В. Л. Арлазаров, А. Битман, М. В. Донской. – М. : Наука, 1983. – 207 с.

### References

1. Mel'nikov B. F., Radionov A. N. *Izvestiya RAN. Programmirovaniye* [Proceedings of the Russian Academy of Sciences. Programming]. 1998, no. 5, pp. 55–62.
2. Mel'nikov B. F. *Izvestiya RAN. Programmirovaniye* [Proceedings of the Russian Academy of Sciences. Programming]. 2001, no. 5, pp. 63–80.
3. Mel'nikov B. F. *Kibernetika i sistemnyy analiz. NAN Ukrainy* [Cybernetics and system analysis. National Academy of Sciences of Ukraine]. 2006, no. 3, pp. 32–42.
4. Melnikov B. *8-th International Conference on High Performance Computing and Grid in Asia Pacific Region, IEEE Computer Society Press Ed.* Beijing, China (HPC-Asia), 2005, pp. 73–80.
5. Melnikov B, Radionov A., Gumayunov V. *8 th International Conference on Enterprise Information Systems, aphos.* Cyprus, ICEIS, 2006, pp. 360–364.
6. Mel'nikov B. F., Mel'nikova E. A. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Ser. Estestvennye nauki* [University proceedings. Volga region. Series: Natural sciences]. 2007, no. 6 (33), pp. 3–11.
7. Baumgertner S. V., Mel'nikov B. F. *Vestnik Voronezhskogo gosudarstvennogo universiteta. Ser. sistemnyy analiz i informatsionnye tekhnologii* [Bulletin of Voronezh State University. Ser. System analysis and information technologies]. 2010, no. 1, pp. 5–7.
8. Rassel S., Norvig P. *Iskusstvennyy intellekt – sovremennyy podkhod* [Artificial intelligence – modern approach]. Moscow; Saint Petersburg; Kiev: Vil'yams, 2006, 1410 p.
9. Michie D. *Advanced in Programming and Non-Numerical Computation, Pergamon Ed.* Oxford, 1966, pp. 183–200.
10. Hauk T., Buro M., Schaeffer J. *Rediscovering \*-minimax search.* Computers and Games, 2004, pp. 35–50.
11. Lyuger D. *Iskusstvennyy intellekt – strategii i metody resheniya slozhnykh problem* [Artificial intelligence – strategies and methods of complex problem solution]. Moscow; Saint Petersburg; Kiev: Vil'yams, 2003, 864 p.
12. Billings D. *University of Alberta (USA), Ph. D. thesis*, 2006, 211 p.
13. Tesauro G. *Neural Computation.* 1989, vol. 1, no. 3, pp. 321–32.
14. Adel'son-Vel'skiy G. M., Arlazarov V. L., Donskoy M. V. *Programmirovaniye igr* [Programming of games]. Moscow: Nauka, 1978, 255 p.
15. Adel'son-Vel'skiy G. M., Arlazarov V. L., Bitman A., Donskoy M. V. *Mashina igraet v shakhmaty* [Machine plays chess]. Moscow: Nauka, 1983, 207 p.

***Мельников Борис Феликсович***

доктор физико-математических наук,  
профессор, кафедра прикладной  
математики и информатики,  
Тольяттинский государственный  
университет (Россия, г. Тольятти,  
ул. Белорусская, 14)

E-mail: B.Melnikov@tltsu.ru

***Mel'nikov Boris Feliksovich***

Doctor of physical and mathematical  
sciences, professor, sub-department  
of applied mathematics and informatics,  
Togliatti State University (14 Belorusskaya  
street, Togliatti, Russia)

***Мельникова Елена Анатольевна***

кандидат физико-математических наук,  
доцент, кафедра прикладной математики  
и информатики, Тольяттинский  
государственный университет  
(Россия, г. Тольятти, ул. Белорусская, 14)

E-mail: E.Melnikova@tltsu.ru

***Mel'nikova Elena Anatol'evna***

Candidate of physical and mathematical  
sciences, associate professor, sub-department  
of applied mathematics and informatics,  
Togliatti State University (14 Belorusskaya  
street, Togliatti, Russia)

---

УДК 004.8.023, 004.83

**Мельников, Б. Ф.**

**Подход к программированию недетерминированных игр (Часть I. Описание общих эвристик) / Б. Ф. Мельников, Е. А. Мельникова // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2013. – № 4 (28). – С. 29–38.**