

УДК 519.833+519.837

ББК 22.18

О ПРОБЛЕМЕ РЕШЕНИЯ МНОГОШАГОВЫХ ИГР В УСЛОВИЯХ ДЕФИЦИТА ВРЕМЕНИ

АНДРЕЙ В. ЧЕРНОВ

Нижегородский государственный
университет им. Н.И. Лобачевского
603950, Нижний Новгород, пр. Гагарина, 23,
Нижегородский государственный технический
университет им. Р.Е. Алексеева
603950, Нижний Новгород, ул. Минина, 24
e-mail: chavnn@mail.ru

На примере антагонистической многошаговой игры рассматривается проблема выбора первым игроком по возможности оптимальной стратегии поведения в условиях дефицита времени, которым он располагает для осуществления этого выбора. Предполагается, что игроки делают ходы поочередно и перед началом каждого хода времени, которое имеется в их распоряжении для его выбора, недостаточно для построения всего дерева развития игры, а достаточно лишь для построения поддерева, состоящего из ограниченного количества дуг. Поэтому встает проблема наилучшего выбора стратегии построения этого поддерева и соответствующего сужения дальнейшей игры. Предполагается, что после построения указанного поддерева каждый игрок производит выбор своего хода на основании расчетов по алгоритму Куна для игры на этом поддереве. Рассматривается два способа построения поддерева. Первый – наивный способ, основанный на построении полного дерева

для «равномерного» сужения игры на меньшее количество ходов. Второй способ основан на вероятностном отборе для дальнейшего ветвления наиболее «перспективных» дуг, исходящих из текущей вершины. Доказывается, что первый способ может приводить к сколь угодно большим просчетам игрока. Для второго способа доказывается, что игрок, хотя и сужает свои априорные возможности, вынуждает другого игрока действовать в рамках выбранной первым игроком подыгры, для которой им произведен полный расчет на существенно большее количество ходов вперед и при этом выбор поддерева произведен как ожидаемо наилучший (в вероятностном смысле). Теоретические (довольно прозрачные) рассуждения подкрепляются конкретным примером антагонистической игры на древовидном графе, а также результатами работы авторской программы, реализующей игру в шашки.

Ключевые слова: многошаговая игра в условиях дефицита времени, алгоритм Куна, ожидаемо наилучший выбор поддерева игры.

Поступила в редакцию: 16.01.19 *После доработки:* 25.04.19 *Принята к публикации:* 10.06.19

1. Введение

Как отмечено в обзорной статье [10], «Во многих приложениях динамической теории игр требуется, по крайней мере, одна, а подчас и много моделей поведения агентов с ограничениями по восприятию, интеллекту, памяти, способности к расчетам и дееспособности». В частности, на взгляд автора, одним из ограничений, с которым приходится считаться игрокам на практике, является нехватка времени, доступного им для анализа ситуации, либо бессмысленность построения дерева игры дальше, чем на определенное количество ходов в связи с существенным влиянием на последующее развитие игры различных неучтенных факторов. В связи с этим, в [10] указывается, что «Требуется больше исследований многошаговых игр $\langle \dots \rangle$, в которых игроки не рассматривают прошлое дальше, чем на T_1 периодов $\langle \text{шагов по времени} \rangle$, и не строят планов дальше, чем на T_2 периодов $\langle \dots \rangle$ Эта работа сейчас далека от формальной теории игр». А между тем, «Теория может влиять на практику, но практика должна влиять на разработку подходящих моделей, если мы хотим, чтобы

их предмет не был всего лишь упражнением в маловажной области чистой математики».

В условиях дефицита времени перед игроками встает проблема рационального сужения множеств возможных альтернатив. Отметим, что в главах 2 и 5 книги [6], посвященной проблеме устойчивости принципа оптимальности в неантагонистических многошаговых играх с полной информацией, рассмотрены возможные способы сужения множеств ситуаций абсолютного равновесия. К сожалению, в условиях дефицита времени нельзя считать, что игроки обладают всей полнотой информации. Объективно говоря, они могут успеть получить лишь часть информации по своему выбору.

Проблема рационального поведения игроков в условиях дефицита времени в некотором смысле родственна проблеме рационального поведения в условиях неопределенности. Учет фактора неопределенности в рамках коллективного принятия решений изучался, например, в [4, 5], где соответствующая формализация строилась как игра с неполной информацией. При этом неполнота информации игрока может касаться как множеств допустимых альтернатив, так и функций выигрыша других игроков. В литературе проблема неполноты информации изучалась особенно интенсивно для моделей иерархических систем управления, где Ведущий может быть информирован лучше или хуже Ведомого (Ведомых). Здесь наряду с объективным описанием игры приходится рассматривать множество ее субъективных описаний с позиций всех игроков [3]; см. также обзор в [9, глава 1, §1.2].

Отметим, что многошаговые игры с учетом фактора времени впервые, видимо, были рассмотрены в работе [7]. Там предполагалось, что обоим игрокам известно полное дерево игры, а время расходуется лишь на реализацию сделанного выбора на каждом шаге. Однако построение дерева игры тоже требует расхода времени.

Как известно, для решения конечных многошаговых игр (в смысле отыскания положения равновесия по Нэшу), и в частности, для игр, заданных на древовидном графе, широко применяется алгоритм Куна (см., например, [1, теорема 13.1, с.150]), что объясняется достаточно высокой его эффективностью. Однако для того, чтобы можно было им воспользоваться, игроки должны обладать ресурсом вре-

мени, достаточным для построения дерева игры. Вместе с тем, для практики характерна ситуация, когда игроки вынуждены действовать в условиях дефицита времени, когда доступного ресурса достаточно лишь для построения некоторого поддерева, состоящего из весьма ограниченного количества дуг по сравнению со всем деревом. Более того, для некоторых игр (классический пример: шашки, шахматы) полное дерево столь велико, что построить его просто бывает невозможно. В таких случаях, как правило, производят расчет лишь на ограниченное количество ходов вперед (в компьютерных играх, реализующих шашки и шахматы, насколько известно автору, рассчитываемый горизонт событий составляет, как правило, порядка шести ходов или меньше; при большем количестве ходов время расчетов повышается настолько, что «пользователю становится скучно»). Далее мы покажем, что это не самый лучший способ выбора поддерева, поскольку он может приводить к весьма существенным просчетам соответствующего игрока. Кроме того, исходя из достаточно простых соображений, мы предложим другой способ, который, как будет показано, является ожидаемо наилучшим в вероятностном смысле.

В целях простоты изложения будем рассматривать антагонистическую игру на древовидном графе. При рассмотрении таких игр обычно указывают лишь выигрыш первого игрока на каждой из висячих (финальных) вершин. Для игры в условиях дефицита времени такая модель не является информативной, а следовательно, не может быть адекватной. Здесь для адекватной информативности на каждой дуге нужно указывать вес, характеризующий тот выигрыш, который игрок, производящий выбор, получит, выбрав эту дугу. Финальный выигрыш по каждому пути, ведущему к той или иной висячей вершине, формируется как сумма весов составляющих его дуг, взятых со знаком плюс, если дуга относится к первому игроку, и со знаком минус в противном случае.

2. Два способа построения поддерева игры и их анализ

Будем считать, что игрок уже находится в некоторой позиции, которая является корнем полного дерева всех возможных путей дальнейшего развития игры.

Одноточечное множество, состоящее из корневой вершины, называем *первым уровнем*. Множество всех вершин, достижимых за одну

дугу из корневой вершины, называем *вторым уровнем*. Множество всех вершин, достижимых за одну дугу из какой-либо вершины k -го уровня, называем $(k + 1)$ -м *уровнем*. Всякую дугу, исходящую из какой-либо вершины k -го уровня, называем *дугой k -го уровня*. Будем считать, что для каждой дуги вполне определенным образом назначено некоторое число, которое будем называть *весом дуги*. Пусть указанное полное дерево имеет ℓ уровней, соответствующих ℓ ходам, совершаемых игроками поочередно. Ходом является выбор одной из дуг, выходящих из данной вершины. При этом вес дуги понимается как та сиюминутная плата, которую получает игрок, делающий ход, от своего противника за выбор этой дуги. Финальный выигрыш игрока для реализовавшегося пути к висячей вершине понимается как суммарный его выигрыш по всем дугам пути. Однако указанное полное дерево игроку неизвестно. Но для каждой вершины (до которой он дошел в своих расчетах) он может, потратив на это некоторое время, построить все исходящие из нее дуги и определить их веса. Это построение исходящих дуг и определение их весов будем называть *ветвлением*. Для корня (*вершины первого уровня*) все исходящие из него дуги (*дуги первого уровня*) и их веса игроку известны. Вопрос состоит в том, по какому принципу отбирать вершины для дальнейшего ветвления. Дугу, для которой вес уже определен, будем называть *прозондированной*. Вершину, которая является началом прозондированной дуги, также будем называть *прозондированной*. Вершину, которая, находясь на конце прозондированной дуги, еще не подвергнута ветвлению, будем называть *вершиной переходного статуса*, или коротко *переходной*. Если переходная вершина является финальной, то ее считаем зондированной по определению (поскольку для нее ветвление не требуется). Все остальные вершины называем *незондированными*. Переходную вершину называем *частично зондированной*, если для нее известно количество исходящих из нее дуг (но не известны их веса). Частично зондированную вершину будем называть также *частично зондированной на один уровень*. Если же для переходной вершины известно количество исходящих из нее дуг, а также для конца каждой из этих дуг – количество исходящих из него дуг, то такую вершину будем называть *частично зондированной на два уровня*, и т.д. При этом будем предполагать, что

для каждой переходной вершины определена метка, понимаемая как некоторое число, характеризующее перспективность этой вершины для дальнейшего анализа (то есть ветвления). Отметим, что сама по себе идея использования меток вершин достаточно характерна для алгоритмов сетевой оптимизации (в частности, алгоритма Дейкстры, алгоритма решения задачи о максимальном потоке и т.д.), см., например, [8, глава 6]. Для переходной вершины, которая не является частично зондированной, будем вычислять эту метку как сумму весов всех дуг, составляющих путь из корня в эту вершину, взятых со знаком плюс, если дуга относится к первому игроку, и со знаком минус в противном случае. Для финальной зондированной вершины метку в любом случае определяем именно таким образом. Что касается частично зондированных переходных вершин, см. раздел 4. В этом разделе считаем, что ни одна из переходных вершин не является частично зондированной (то есть мы каждый раз имеем минимум априорной информации). В начале работы переходными вершинами являются все вершины второго уровня. Таким образом, вопрос состоит в том, как именно отбирать вершины для дальнейшего ветвления. Рассмотрим следующие два принципиально различных способа.

Способ I. Сначала производим ветвление всех вершин второго уровня, затем – ветвление всех вершин третьего уровня, и т.д., до тех пор, пока хватает отведенного для расчетов времени.

Способ II. Из всех переходных вершин выбираем вершину с наибольшей меткой; затем выбранную вершину подвергаем ветвлению, и т.д., до тех пор, пока хватает отведенного для расчетов времени.

В [10, §4.3] приводится простой учебный пример многошаговой игры, наглядно демонстрирующий, что более высокий уровень информированности дает преимущества в выигрыше. Далее мы докажем, по сути дела, некоторое родственное утверждение.

Теорема 2.1. *Если оба игрока используют способ I, то оценка своего выигрыша первым игроком в корне дерева, может отличаться от его выигрыша в подыгре, анализируемой по алгоритму Куна на следующем ходе вторым игроком, на сколь угодно большую величину.*

Доказательство. Во избежание громоздких выкладок и никак

не ограничивая общности рассуждений, будем считать, что ресурс времени позволяет каждому из игроков производить расчеты на два хода вперед. Пусть k – количество дуг, исходящих из корня дерева, и при заданном $i \in \overline{1, k}$, k_i – количество дуг, исходящих из вершины на конце i -й дуги первого уровня; a_i – вес i -й дуги первого уровня, $i \in \overline{1, k}$; b_{ij} – вес j -й дуги второго уровня, исходящей из вершины на конце i -й дуги первого уровня, $i \in \overline{1, k}$, $j \in \overline{1, k_i}$; k_{ij} – количество дуг, исходящих из вершины на конце j -й дуги второго уровня, исходящей из вершины на конце i -й дуги первого уровня, $i \in \overline{1, k}$, $j \in \overline{1, k_i}$; a_{ijs} – вес s -й дуги третьего уровня, исходящей из вершины на конце j -й дуги второго уровня, исходящей из вершины на конце i -й дуги первого уровня, $i \in \overline{1, k}$, $j \in \overline{1, k_i}$, $s \in \overline{1, k_{ij}}$; и т.д. Тогда в рамках способа I оценка своего выигрыша первым игроком будет равна

$$v_1 = \max_{i \in \overline{1, k}} \min_{j \in \overline{1, k_i}} (a_i - b_{ij}).$$

Исходя из этой оценки, первый игрок, находясь в корне дерева, выберет дугу номер $i \in \overline{1, k}$, на котором реализуется максимум. Теперь, находясь в вершине на конце этой дуги, второй игрок, действуя аналогичным образом, будет анализировать двухходовую подыгру со значением выигрыша первого игрока, определяемого как

$$v_2 = \min_{j \in \overline{1, k_i}} \max_{s \in \overline{1, k_{ij}}} (a_i - b_{ij} + a_{ijs}), \quad \text{при } i \in \overline{1, k}.$$

Если бы все элементы a_{ijs} были равны нулю, то оценки v_1 и v_2 совпали бы. Если бы все элементы a_{ijs} были равны одной и той же константе, то значения v_1 и v_2 отличались бы на эту константу, но выбор второго игрока соответствовал бы его выбору, предсказанному первым игроком. Однако на самом деле величины a_{ijs} могут быть, вне ведения первого игрока, отрицательны и сколь угодно велики по модулю. В итоге оказывается, что опираясь на оценку v_1 , первый игрок может совершить сколь угодно большую ошибку. Теорема доказана.

Обращаясь к доказательству теоремы 2.1, нетрудно заметить, что до того, как мы прозондировали дуги второго уровня, их веса b_{ij} нам неизвестны. Однако нам вполне может быть известно, что эти величины принимают значения в некоторых известных границах, скажем, $b_{ij} \in [\sigma_1; \sigma_2]$. Поскольку при отборе вершин для дальнейшего ветвле-

ния никакая другая информация нам в принципе неизвестна, то естественно рассматривать b_{ij} (и их дальнейшие аналоги) как случайные величины. Идея использования случайных величин при недостатке информации, вообще говоря, не нова, см., например, [11]. Новым в нашем подходе является то, что математические ожидания «локальных» (разной степени локальности) выигрышей мы используем в качестве предварительной информации для выделения подыгры, в которой собираемся проводить все расчеты точно (по алгоритму Куна). Распределение случайных величин b_{ij} при отсутствии дополнительной информации естественно считать равномерным. Тогда при своих расчетах, связанных с отбором вершин для ветвления (но не при расчетах по алгоритму Куна после построения поддерева) разумно заменить b_{ij} (и их дальнейшие аналоги) их математическими ожиданиями $\sigma = (\sigma_1 + \sigma_2)/2$. При таком подходе вместо v_1 мы получим

$$\tilde{v}_1 = -\sigma + \max_{i=1,k}(a_i).$$

Соответствующий индекс $i = \bar{i}$, на котором достигается максимум, можно найти из условия $a_i \rightarrow \max_{i=1,k}$, что как раз и соответствует способу II. Конечно, оценка \tilde{v}_1 более грубая по сравнению с v_1 . Но она нам ничего не стоила в смысле расхода времени. Поэтому мы можем произвести ветвление вершины на конце дуги номер $i = \bar{i}$ и повторить эту процедуру (выбирая для ветвления очередную переходную вершину с максимальной меткой независимо от ее уровня) еще несколько раз, дойдя, возможно, до гораздо более высоких уровней, чем в рамках способа I. В итоге (если в большинстве случаев наш выбор оказался удачным, а исходя из вероятностного способа выбора, так, скорее всего, и будет) мы получим сужение исходного дерева не «по высоте» (типа I), а «по ширине» (типа II). Поскольку это сужение будет иметь существенно большую высоту, чем сужение типа I, и возможно, до высшего уровня ℓ , то ясно, что оно будет более адекватно для применения алгоритма Куна. Что же касается тех дуг, которые мы не успели провести ветвлением до приемлемого уровня, то их мы можем просто отсечь (запретить для себя их использование), если в полученной таким образом подыгре наш выигрыш не меньше некоторого приемлемого значения; в противном случае – наоборот, их оставить, а все остальное отсечь (поскольку отсекаемые в этом случае ветви нам

почти достоверно ничего хорошего не дадут). Кроме того, поскольку при отборе вершин для ветвления мы исходили из математических ожиданий выигрыша противника, то в большинстве случаев ветви, концы которых остались незондированными, будут не слишком перспективными, и в этом смысле, отказываясь от их использования, мы не сильно теряем.

3. Сравнение расчетов по способам I и II на конкретном примере

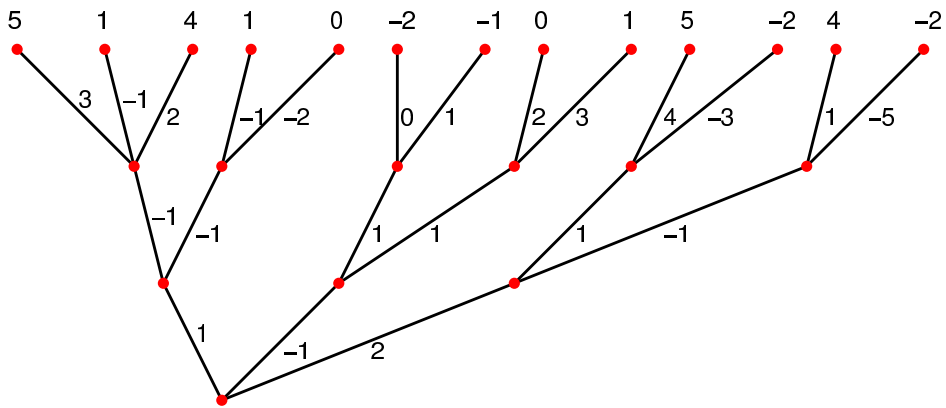


Рисунок 1. Полное дерево

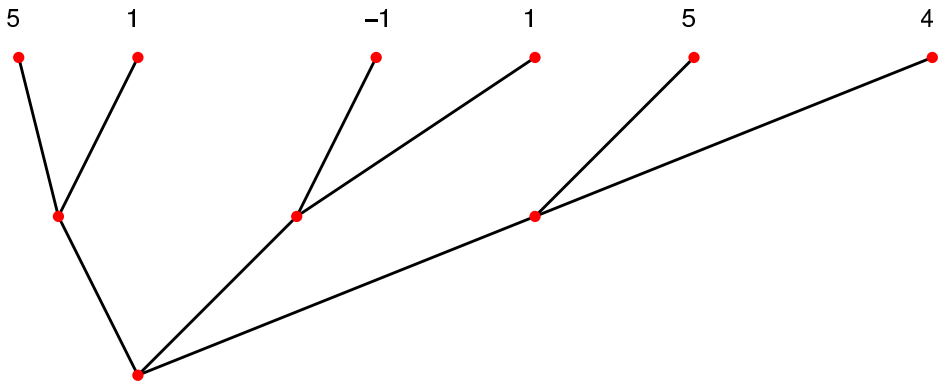


Рисунок 2. Редукция полного дерева № 1

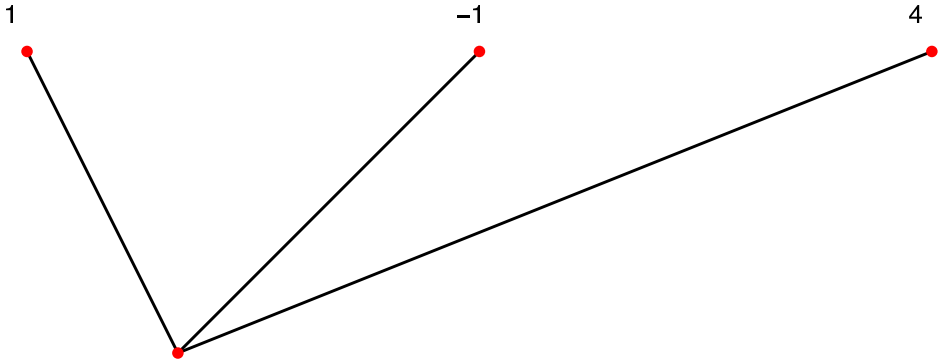


Рисунок 3. Редукция полного дерева № 2

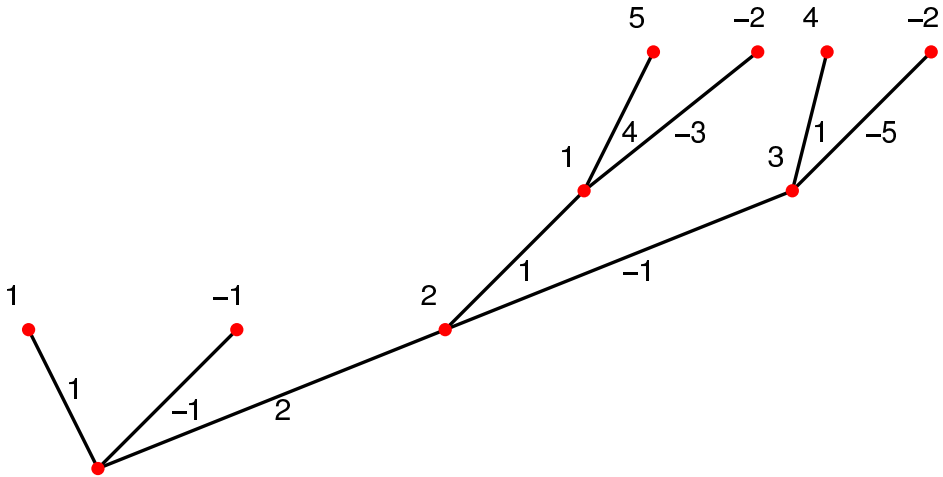


Рисунок 4. Способ II

Пусть дерево всех возможных путей развития игры изображено на рис. 1. Расчеты по алгоритму Куна по этому полному дереву дают первому игроку выигрыш 4, с выбором крайней правой дуги в качестве наилучшего первого хода – см. рис. 2, 3. Однако предположим, что из-за нехватки времени каждый из игроков может прозондировать лишь 9 дуг (включая те, которые выходят из корня). При использовании способа II получаем подыгру с тем же выигрышем первого игрока 4, и с тем же оптимальным выбором первого хода, см. рис. 4. При использовании же способа I получаем подыгру с выигрышем 2 и крайней левой дугой в качестве наилучшего первого хода, см. рис. 5, 6. Нетрудно заметить, что если, находясь в вершине

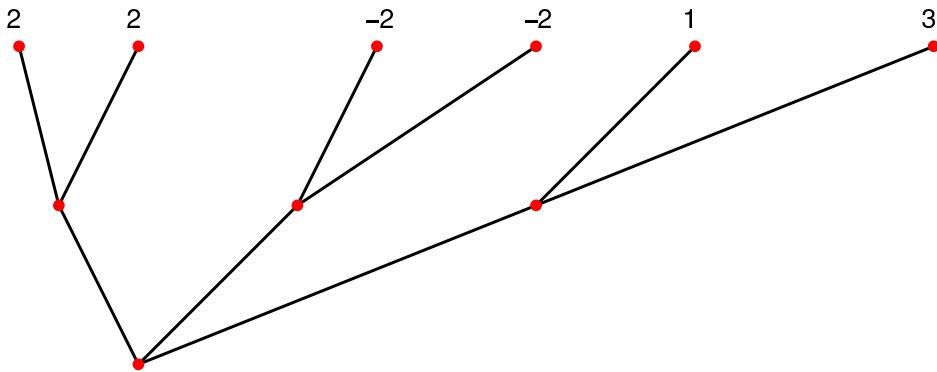


Рисунок 5. Способ I – поддереву

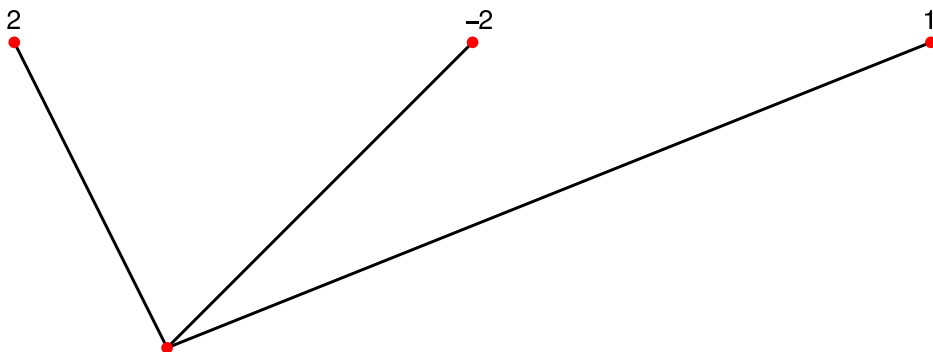


Рисунок 6. Способ I – редукция

на ее конце, второй игрок будет производить расчеты по способу I, то он будет находиться в рамках подыгры с выигрышем первого игрока 1. Таким образом, пользуясь способом I, первый игрок делает неверный выбор первого хода и вместо выигрыша 4 получает даже не 2 (на что рассчитывает), а 1. Причем можно подобрать цифры и таким образом, что мнимый выигрыш 2 обратится в проигрыш.

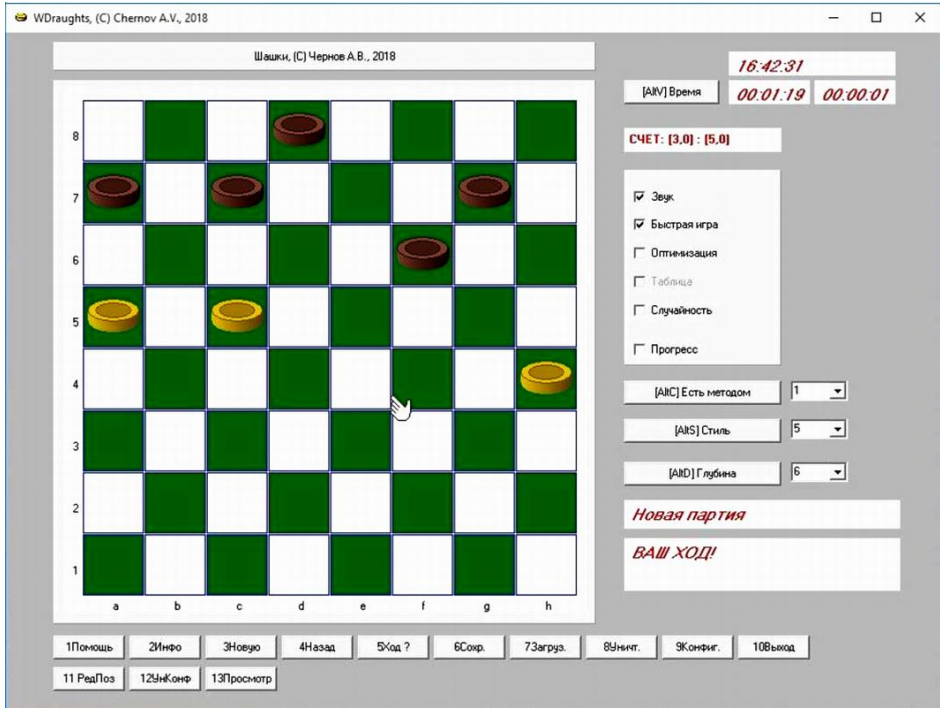


Рисунок 7. Окно программы WDraughts

4. Случай частично зондированных переходных вершин

Рассмотрим, как наилучшим (при доступной нам априорной информации) образом вычислять метки частично зондированных (сначала на один уровень, потом – на два уровня) переходных вершин. Естественно, что речь идет о способе II (поскольку для способа I метки не требуются). Для простоты (но не ограничивая общности рассуждений) будем предполагать, что решается вопрос о выборе очередной вершины для ветвления в ситуации, когда мы находимся в корне дерева. Иначе говоря, переходными вершинами являются вершины второго уровня. В общей ситуации все будет аналогично, с той лишь разницей, что в качестве корня будет выступать прозондированная вершина, только что подвергнутая ветвлению и тем самым,

породившая новые переходные вершины, для которых требуется вычислить метки.

Итак, будем предполагать, что веса a_i , $i = \overline{1, k}$, дуг первого уровня нам известны, и для каждой i -й переходной вершины на конце i -й дуги первого уровня нам известно количество k_i исходящих из нее дуг. При этом веса b_{ij} , $i = \overline{1, k}$, $j = \overline{1, k_i}$, нам неизвестны и рассматриваются нами как случайные величины, имеющие равномерное распределение на (для простоты – одном и том же) отрезке $[\sigma_1; \sigma_2]$. В этом случае в качестве метки i -й переходной вершины естественно принять математическое ожидание

$$\mu_i = M\left[\min_{j=\overline{1, k_i}}(a_i - b_{ij})\right].$$

Получим конкретную формулу для вычисления меток μ_i .

Теорема 4.1. *Имеем:*

$$\mu_i = a_i - k_i I[k_i; \sigma_1, \sigma_2], \quad I[k_i; \sigma_1, \sigma_2] = \int_{\sigma_1}^{\sigma_2} \frac{z(z - \sigma_1)^{k_i - 1}}{(\sigma_2 - \sigma_1)^{k_i}} dz, \quad i = \overline{1, k}. \quad (4.1)$$

Иначе говоря,

$$\mu_i = a_i - \frac{k_i \sigma_2 + \sigma_1}{k_i + 1}, \quad i = \overline{1, k}. \quad (4.2)$$

Доказательство. Выберем произвольно индекс $i \in \overline{1, k}$. В соответствии с известными свойствами математического ожидания, очевидно, что

$$\mu_i = M\left[a_i + \min_{j=\overline{1, k_i}}(-b_{ij})\right] = a_i - \nu_i, \quad \nu_i = M\left[\max_{j=\overline{1, k_i}} b_{ij}\right].$$

Согласно [2, глава 9, §9.6, с.377–378, задача 4], для независимых непрерывных случайных величин X_1, \dots, X_n , имеющих функцию распределения $F_s(y)$ и плотность распределения $f_s(y)$, $s = \overline{1, n}$, случайная величина

$$Y = \max_{s=\overline{1, n}} X_s$$

имеет функцию распределения

$$G(y) = \prod_{s=1}^n F_s(y)$$

и плотность распределения

$$g(y) = \sum_{s=1}^n f_s(y) \prod_{t \neq s} F_t(y).$$

В частности, для $F_s(y) = F(y)$, $f_s(y) = f(y)$, $s = \overline{1, n}$, получаем:

$$G(y) = [F(y)]^n, \quad g(y) = n f(y) [F(y)]^{n-1}.$$

В нашем случае случайные величины b_{ij} , $j = \overline{1, k_i}$, имеют функцию распределения

$$F(z) = \begin{cases} \frac{z - \sigma_1}{\sigma_2 - \sigma_1}, & z \in [\sigma_1; \sigma_2], \\ 0, & z < \sigma_1, \\ 1, & z > \sigma_2, \end{cases}$$

и плотность распределения

$$f(z) = \begin{cases} \frac{1}{\sigma_2 - \sigma_1}, & z \in [\sigma_1; \sigma_2], \\ 0, & z < \sigma_1, z > \sigma_2. \end{cases}$$

Соответственно, случайная величина $Z_i = \max_{j=\overline{1, k_i}} b_{ij}$ имеет плотность распределения

$$g(z) = k_i f(z) [F(z)]^{k_i-1} = \begin{cases} \frac{k_i (z - \sigma_1)^{k_i-1}}{(\sigma_2 - \sigma_1)^{k_i}}, & z \in [\sigma_1; \sigma_2], \\ 0, & z < \sigma_1, z > \sigma_2, \end{cases}$$

и математическое ожидание

$$\nu_i = M[Z_i] = \int_{-\infty}^{+\infty} z g(z) dz = k_i I[k_i; \sigma_1, \sigma_2].$$

Таким образом, справедлива формула (4.1).

Докажем формулу (4.2). Рассмотрим интеграл

$$\int_{\sigma_1}^{\sigma_2} z (z - \sigma_1)^{k_i-1} dz = \int_{\sigma_1}^{\sigma_2} (z - \sigma_1)^{k_i} dz + \sigma_1 \int_{\sigma_1}^{\sigma_2} (z - \sigma_1)^{k_i-1} dz =$$

$$= \frac{(z - \sigma_1)^{k_i+1}}{k_i + 1} \Big|_{\sigma_1}^{\sigma_2} + \sigma_1 \frac{(z - \sigma_1)^{k_i}}{k_i} \Big|_{\sigma_1}^{\sigma_2} = \frac{(\sigma_2 - \sigma_1)^{k_i+1}}{k_i + 1} + \sigma_1 \frac{(\sigma_2 - \sigma_1)^{k_i}}{k_i}.$$

Таким образом,

$$I[k_i; \sigma_1, \sigma_2] = \frac{\sigma_2 - \sigma_1}{k_i + 1} + \frac{\sigma_1}{k_i} = \frac{\sigma_2}{k_i + 1} + \frac{\sigma_1}{k_i(k_i + 1)},$$

следовательно,

$$k_i I[k_i; \sigma_1, \sigma_2] = \frac{k_i \sigma_2 + \sigma_1}{k_i + 1}.$$

Отсюда получаем формулу (4.2). Теорема доказана.

Замечание 4.1. Из формулы (4.2) видно, что в случае постоянного числа $k_i = r$ для всех $i = \overline{1, k}$ задача максимизации $\mu_i \rightarrow \max_{i=\overline{1, k}}$ равносильна задаче $a_i \rightarrow \max_{i=\overline{1, k}}$. Поэтому в таком случае оптимальный способ вычисления меток будет совпадать с описанным в разделе 2.

Далее рассмотрим случай, когда переходные вершины (второго уровня) являются частично зондированными на два уровня. Соответственно, будем предполагать, что веса a_i , $i = \overline{1, k}$, дуг первого уровня нам известны, и для каждой i -й переходной вершины на конце i -й дуги первого уровня нам известно количество k_i исходящих из нее дуг, а для конца j -й дуги, исходящей из i -переходной вершины, известно количество k_{ij} исходящих из него дуг, $i = \overline{1, k}$, $j = \overline{1, k_i}$. При этом веса b_{ij} , а также a_{ijs} , $s = \overline{1, k_{ij}}$, $i = \overline{1, k}$, $j = \overline{1, k_i}$, нам неизвестны и рассматриваются нами как случайные величины, имеющие равномерное распределение на (для простоты – одном и том же) отрезке $[\sigma_1; \sigma_2]$. В этом случае в качестве метки i -й переходной вершины естественно принять математическое ожидание

$$\gamma_i = M \left[\min_{j=\overline{1, k_i}} (a_i - b_{ij} + a_{ijs}) \right].$$

Получим конкретную формулу для вычисления меток γ_i .

Лемма 4.1. *Имеем:*

$$\gamma_i = a_i - \delta_i, \quad \delta_i = \delta_i[k_i; k_{ij}, j = \overline{1, k_i}; \sigma_1, \sigma_2] = \int_{-\infty}^{+\infty} z g_i(z) dz,$$

$$g_i(z) = \sum_{j=1}^{k_i} f_{ij}(z) \prod_{t \neq j} F_{it}(z),$$

$$F_{ij}(z) = \int_{\sigma_1}^{z+\sigma_2} \varphi(x) dx \int_{x-z}^{\sigma_2} \psi_{ij}(y) dy, \quad z \geq \sigma_1 - \sigma_2; \quad (4.3)$$

$$F_{ij}(z) = 0, \quad z < \sigma_1 - \sigma_2;$$

$$f_{ij}(z) = \int_{\sigma_1}^{z+\sigma_2} \varphi(x) \psi_{ij}(x-z) dx, \quad z \geq \sigma_1 - \sigma_2; \quad f_{ij}(z) = 0, \quad z < \sigma_1 - \sigma_2; \quad (4.4)$$

$$\varphi(x) = \frac{1}{\sigma_2 - \sigma_1}, \quad \varphi(x) = 0, \quad x \notin [\sigma_1; \sigma_2]; \quad (4.5)$$

$$\psi_{ij}(y) = \frac{k_{ij}(y - \sigma_1)^{k_{ij}-1}}{(\sigma_2 - \sigma_1)^{k_{ij}}}, \quad y \in [\sigma_1; \sigma_2], \quad \psi_{ij}(y) = 0, \quad y \notin [\sigma_1; \sigma_2]. \quad (4.6)$$

Доказательство. Выберем произвольно индекс $i \in \overline{1, k}$. Очевидно, что $\gamma_i = a_i - \delta_i$, где

$$\delta_i = M \left[\max_{j=1, k_i} \min_{s=1, k_{ij}} (b_{ij} - a_{ijs}) \right] = M[Z_{ij}],$$

$$Z_{ij} = b_{ij} + Y_{ij}, \quad Y_{ij} = -S_{ij}, \quad S_{ij} = \max_{s=1, k_{ij}} a_{ijs}.$$

Обозначим $\varphi(x)$ – плотность распределения (каждой из) случайных величин b_{ij} ; $\psi_{ij}(y)$ – плотность распределения случайной величины S_{ij} , а $\Psi_{ij}(y)$ – соответствующую функцию распределения; $\omega_{ij}(y)$ – плотность распределения случайной величины Y_{ij} . Тогда, учитывая, что b_{ij} имеют равномерное распределение на $[\sigma_1; \sigma_2]$, для функции $\varphi(x)$ имеем формулу (4.5). И аналогично доказательству теоремы 4.1, для функции $\psi_{ij}(y)$ получаем формулу (4.6). Рассмотрим вероятность

$$P(Y_{ij} < y) = P(-S_{ij} < y) = P(S_{ij} > -y) = 1 - P(S_{ij} \leq -y) =$$

$$= 1 - P(S_{ij} < -y) = 1 - \Psi_{ij}(-y).$$

Дифференцируя, получаем:

$$\omega_{ij}(y) = \psi_{ij}(-y).$$

Обозначим $F_{ij}(z)$ и $f_{ij}(z)$ – функцию и плотность распределения случайной величины Z_{ij} ;

$$D(z) = \{(x, y) \in \mathbb{R}^2 : x + y < z\}.$$

Имеем [2, формула (7.4.5)]:

$$F_{ij}(z) = P(b_{ij} + Y_{ij} < z) = \iint_{D(z)} \varphi(x)\omega_{ij}(y) dx dy.$$

Заметим, что $b_{ij} \in [\sigma_1; \sigma_2]$, $Y_{ij} \in [-\sigma_2; -\sigma_1]$. С учетом этого, $F_{ij}(z) = 0$ при $z < \sigma_1 - \sigma_2$. Предполагая, что $z \geq \sigma_1 - \sigma_2$, рассмотрим

$$F_{ij}(z) = \int_{\sigma_1}^{z+\sigma_2} \varphi(x) dx \int_{-\sigma_2}^{z-x} \omega_{ij}(t) dt = \int_{\sigma_1}^{z+\sigma_2} \varphi(x) dx \int_{-\sigma_2}^{z-x} \psi_{ij}(-t) dt.$$

Делая замену $t = -y$, получаем формулу (4.3). Дифференцируя, имеем

$$f_{ij}(z) = F'_{ij}(z) = \varphi(z + \sigma_2) \int_{\sigma_2}^{\sigma_2} \psi_{ij}(y) dy + \int_{\sigma_1}^{z+\sigma_2} \varphi(x) \psi_{ij}(x - z) dx,$$

откуда следует формула (4.4). После этого остается лишь сослаться на [2, глава 9, §9.6, с.377–378, задача 4]. Лемма доказана.

Теорема 4.2. *Имеем:*

$$\gamma_i = a_i - \delta_i, \quad \delta_i = \delta_i[k_i; k_{ij}, j = \overline{1, k_i}; \sigma_1, \sigma_2] = \int_{-\Delta}^{\Delta} z g_i(z) dz,$$

$$g_i(z) = \sum_{j=1}^{k_i} f_{ij}(z) \prod_{t \neq j} F_{it}(z),$$

$$F_{ij}(z) = F[k_{ij}](z), \quad f_{ij}(z) = g[k_{ij}](z), \quad \Delta = \sigma_2 - \sigma_1,$$

$$F[r](z) = \begin{cases} 0, & z < -\Delta, \\ 1 + \frac{z}{\Delta} - \frac{1}{r+1} \left\{ 1 - \frac{(-z)^{r+1}}{\Delta^{r+1}} \right\}, & z \in [-\Delta; 0), \\ 1 - \frac{1}{r+1} \left\{ 1 - \frac{z}{\Delta} \right\}^{r+1}, & z \in [0; \Delta], \\ 1, & z > \Delta; \end{cases}$$

$$f[r](z) = \begin{cases} \frac{1}{\Delta} - \frac{(-z)^r}{\Delta^{r+1}}, & z \in [-\Delta; 0), \\ \frac{(\Delta - z)^r}{\Delta^{r+1}}, & z \in [0; \Delta], \\ 0, & z \notin [-\Delta; \Delta]. \end{cases}$$

Доказательство. В соответствии с леммой 4.1, все сводится к конкретизации формулы (4.3) с учетом формул (4.5), (4.6). Выберем произвольно индексы $i \in \overline{1, k}$, $j \in \overline{1, k_i}$ и положим $r = k_{ij}$. По лемме 4.1, $F[r](z) = F_{ij}(z) = 0$ при $z < -\Delta$. Рассмотрим следующие три случая.

1) Пусть $z \in [-\Delta; 0] = [\sigma_1 - \sigma_2; 0]$. Тогда $z + \sigma_2 \in [\sigma_1; \sigma_2]$, $x - z \in [\sigma_1; \sigma_2]$ при всех $x \in [\sigma_1; \sigma_2]$, и формула (4.3) конкретизируется следующим образом:

$$\begin{aligned} F[r](z) &= \int_{\sigma_1}^{z+\sigma_2} \varphi(x) dx \int_{x-z}^{\sigma_2} \psi_{ij}(y) dy = \frac{r}{\Delta^{r+1}} \int_{\sigma_1}^{z+\sigma_2} dx \int_{x-z}^{\sigma_2} (y - \sigma_1)^{r-1} dy = \\ &= \frac{1}{\Delta^{r+1}} \int_{\sigma_1}^{z+\sigma_2} dx (y - \sigma_1)^r \Big|_{x-z}^{\sigma_2} = \\ &= \frac{1}{\Delta^{r+1}} \left\{ \Delta^r \int_{\sigma_1}^{z+\sigma_2} dx - \int_{\sigma_1}^{z+\sigma_2} (x - z - \sigma_1)^r dx \right\} = \\ &= \frac{1}{\Delta^{r+1}} \left[\Delta^r (z + \Delta) - \frac{(x - z - \sigma_1)^{r+1}}{r + 1} \Big|_{\sigma_1}^{z+\sigma_2} \right] = \\ &= \frac{z + \Delta}{\Delta} - \frac{1}{r + 1} \left[1 - \frac{(-z)^{r+1}}{\Delta^{r+1}} \right]. \end{aligned}$$

2) Пусть $z \in [0; \Delta] = [0; \sigma_2 - \sigma_1]$. Тогда $\sigma_1 + z \in [\sigma_1; \sigma_2]$, $x - z \leq \sigma_1$ при всех $x \in [\sigma_1; \sigma_1 + z]$, $x - z \in [\sigma_1; \sigma_2]$ при всех $x \in [\sigma_1 + z; \sigma_2]$, и формула (4.3) конкретизируется следующим образом:

$$\begin{aligned} F[r](z) &= \int_{\sigma_1}^{\sigma_1+z} \varphi(x) dx \int_{\sigma_1}^{\sigma_2} \psi_{ij}(y) dy + \int_{\sigma_1+z}^{\sigma_2} \varphi(x) dx \int_{x-z}^{\sigma_2} \psi_{ij}(y) dy = \\ &= \frac{r}{\Delta^{r+1}} \left\{ z \int_{\sigma_1}^{\sigma_2} (y - \sigma_1)^{r-1} dy + \int_{\sigma_1+z}^{\sigma_2} dx \int_{x-z}^{\sigma_2} (y - \sigma_1)^{r-1} dy \right\} = \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\Delta^{r+1}} \left\{ z\Delta^r + \int_{\sigma_1+z}^{\sigma_2} dx [\Delta^r - (x-z-\sigma_1)^r] \right\} = \\
&= \frac{1}{\Delta^{r+1}} \left\{ \Delta^{r+1} - \frac{(x-z-\sigma_1)^{r+1}}{r+1} \Big|_{\sigma_1+z}^{\sigma_2} \right\} = 1 - \frac{(\Delta-z)^{r+1}}{(r+1)\Delta^{r+1}}.
\end{aligned}$$

3) Пусть $z > \Delta = \sigma_2 - \sigma_1 \geq 0$. Тогда $z + \sigma_2 \geq \sigma_2$, $x - z \leq \sigma_1$ при всех $x \in [\sigma_1; \sigma_2]$, и формула (4.3) конкретизируется следующим образом:

$$F[r](z) = \int_{\sigma_1}^{\sigma_2} \varphi(x) dx \int_{\sigma_1}^{\sigma_2} \psi_{ij}(y) dy = \frac{r}{\Delta^r} \int_{\sigma_1}^{\sigma_2} (y - \sigma_1)^{r-1} dy = \frac{(y - \sigma_1)^r}{\Delta^r} \Big|_{\sigma_1}^{\sigma_2},$$

то есть $F[r](z) = 1$. Выражение плотности распределения получаем дифференцированием: $f[r](z) = F'[r](z)$. Теорема доказана.

Замечание 4.2. Числа k_i, k_{ij} являются натуральными. Величины σ_1, σ_2 во многих случаях можно считать целыми. Для таких случаев имеет смысл составить таблицу значений $\delta_i[k_i; k_{ij}, j = \overline{1, k_i}; \sigma_1, \sigma_2]$ и использовать эту таблицу в компьютерных программах, реализующих способ II в рассматриваемой ситуации.

5. Модификация способа II для игры в шашки

В применении к тем или иным конкретным играм идею, положенную в основу способа II, можно использовать в различных модификациях. Так, например, в свое время автором была написана программа `WDraughts`, реализующая игру в шашки, см. рис.7. В качестве одной из опций этой программы в ней присутствовал режим «Быстрой игры». Этот режим был основан на сходной идее. А именно, при заданном «горизонте событий» (скажем, шесть ходов) сначала производилось построение дерева на два хода вперед, после чего дальнейшему ветвлению до заданного горизонта подвергались лишь те вершины, на пути к которым имелись дуги с уничтожением чьих-либо шашек, то есть имелись признаки более динамичной игры¹. Для сравнения

¹Вообще говоря, можно и в общем случае подвергать ветвлению вершины с наибольшей *по модулю* меткой, отслеживая таким образом ветви наиболее динамичного развития игры.

позиций в игре опционально использовались следующие критерии (и соответствующие «стили игры»).

1. Вспомогательные параметры (по степени важности): а) количество живых шашек; б) их расстояния до последней горизонтали (king line); в) количество шашек «под боем». Для каждого параметра подобрана соответствующая цена (весовой коэффициент); критерий оптимальности – сумма параметров с весовыми коэффициентами.
2. Вспомогательные параметры: а) перспектива окончания игры; б) количество дамек; в) количество обычных шашек; г) количество дамек «под боем»; д) количество простых шашек «под боем»; е) количество шашек, способных двигаться; ж) количество шашек, способных есть; з) расстояние «не дамек» до последней горизонтали. Критерий оптимальности – сумма соответствующих балансов (разностей между игроками) с весовыми коэффициентами.
3. Вспомогательные параметры – те же. Принцип оценки оптимальности: сначала производится оптимизация по наиболее важному параметру; если улучшения нет и нет ухудшения – по менее важному и т.д. (используется метод ранжирования частных критериев).
4. Вспомогательные параметры (см. пункт 2): а),б),в) – анализируются по степени важности (методом ранжирования); из балансов по параметрам е),ж),з) берется минимум, если хотя бы один из них отрицательный, иначе – максимум из них.
5. Критерий оптимальности – количество контролируемых клеток (у каждой – своя цена; цена увеличивается, если клетка занята шашкой или дамкой).
6. Случайный выбор предыдущих критериев.

При этом зачастую оказывалось, что в режиме «Быстрой игры» программа играла не только существенно быстрее (скажем, вместо одной-двух минут тратилась одна секунда на все расчеты за партию),

но и лучше, чем при подробном анализе ходов. Собственно, исходной мотивацией написания этой статьи было желание автора разобраться в том, почему так происходит. В качестве примера приведем далее листинг программы для одной из партий, сыгранной в режиме «Быстрой игры» (здесь компьютер играет черными, пользователь – белыми; стиль игры – 5).

```
UserTime=00:01:31
```

```
ComputerTime=00:00:01
```

```
Original position:
```

```
White:
```

```
a3 % ND
```

```
c3 % ND
```

```
e3 % ND
```

```
g3 % ND
```

```
b2 % ND
```

```
d2 % ND
```

```
f2 % ND
```

```
h2 % ND
```

```
a1 % ND
```

```
c1 % ND
```

```
e1 % ND
```

```
g1 % ND
```

```
Black:
```

```
b8 % ND
```

```
d8 % ND
```

```
f8 % ND
```

```
h8 % ND
```

```
a7 % ND
```

```
c7 % ND
```

```
e7 % ND
```

```
g7 % ND
```

```
b6 % ND
```

```
d6 % ND
```

```
f6 % ND
```

```
h6 % ND
```

History of Moves:

a3-b4 % Actor: White (ND)
 f6-g5 % Actor: Black (ND)
 b4-a5 % Actor: White (ND)
 g5-h4 % Actor: Black (ND)
 b2-a3 % Actor: White (ND)
 e7-f6 % Actor: Black (ND)
 c3-b4 % Actor: White (ND)
 d6-e5 % Actor: Black (ND)
 c1-b2 % Actor: White (ND)
 f6-g5 % Actor: Black (ND)
 d2-c3 % Actor: White (ND)
 g7-f6 % Actor: Black (ND)
 c3-d4 % Actor: White (ND)
 e5:c3 % Actor: Black (ND), Killed: d4 (ND)
 b2:d4 % Actor: White (ND), Killed: c3 (ND)
 f8-g7 % Actor: Black (ND)
 d4-c5 % Actor: White (ND)
 b6:d4 % Actor: Black (ND), Killed: c5 (ND)
 e3:c5 % Actor: White (ND), Killed: d4 (ND)
 g5-f4 % Actor: Black (ND)
 g3:e5 % Actor: White (ND), Killed: f4 (ND)
 f6:d4 % Actor: Black (ND), Killed: e5 (ND)
 d4:b6 % Actor: Black (ND), Killed: c5 (ND)
 f2-g3 % Actor: White (ND)
 h4:f2 % Actor: Black (ND), Killed: g3 (ND)
 e1:g3 % Actor: White (ND), Killed: f2 (ND)
 h6-g5 % Actor: Black (ND)
 g3-h4 % Actor: White (ND)
 g5-f4 % Actor: Black (ND)
 a1-b2 % Actor: White (ND)
 g7-f6 % Actor: Black (ND)
 b2-c3 % Actor: White (ND)
 h8-g7 % Actor: Black (ND)
 g1-f2 % Actor: White (ND)

f4-g3 % Actor: Black (ND)
 h2:f4 % Actor: White (ND), Killed: g3 (ND)
 b6-c5 % Actor: Black (ND)
 b4:d6 % Actor: White (ND), Killed: c5 (ND)
 c7:e5 % Actor: Black (D), Killed: d6 (ND)
 e5:g3 % Actor: Black (D), Killed: f4 (ND)
 g3:e1 % Actor: Black (D), Killed: f2 (ND)
 e1:b4 % Actor: Black (D), Killed: c3 (ND)
 a3:c5 % Actor: White (ND), Killed: b4 (D)
 b8-c7 % Actor: Black (ND)
 h4-g5 % Actor: White (ND)
 f6:h4 % Actor: Black (ND), Killed: g5 (ND)
 a5-b6 % Actor: White (ND)
 c7:a5 % Actor: Black (ND), Killed: b6 (ND)
 c5-d6 % Actor: White (ND)
 a5-b4 % Actor: Black (ND)
 d6-c7 % Actor: White (ND)
 d8:b6 % Actor: Black (ND), Killed: c7 (ND)

Игра заканчивается уничтожением всех белых шашек при том, что остается пять черных, см. рис. 7. Пользователь (автор) потратил на свои расчеты за всю игру около полутора минут, программа – порядка одной секунды.

С точки зрения проблемы дефицита времени важно из двух или более путей, ведущих к одинаковой величине финального выигрыша, уметь выбирать наиболее короткий. Отметим в связи с этим, что описанная выше идея, положенная в основу реализации «быстрой игры», будучи примененной в тех или иных конкретных играх, может приводить к укорачиванию реализовавшейся траектории. В частности, если в игре в шашки каждый раз подвергать дальнейшему анализу лишь те позиции, которые связаны с большим сиюминутным выигрышем, и производить выбор стратегии, выбрасывая из множества доступных альтернатив те варианты, которые не были проанализированы, то количество шашек, остающихся на доске, будет убывать, вероятно, быстрее, чем при иной организации выбора, а это, очевидно, способствует, вообще говоря, более быстрому продвижению к финалу партии.

СПИСОК ЛИТЕРАТУРЫ

1. Васин А.А., Морозов В.В. *Введение в теорию игр с приложениями к экономике*. М.: МГУ, 2003. 278 с.
2. Вентцель Е.С., Овчаров Л.А. *Теория вероятностей и ее инженерные приложения*. М.: Высшая школа, 2000. 480 с.
3. Горелик В.А., Кононенко А.Ф. *Теоретико-игровые модели принятия решений в эколого-экономических системах*. М.: Радио и связь, 1982. 145 с.
4. Жуковский В.И. *Кооперативные игры при неопределенности и их приложения*. М.: Эдиториал УРСС, 1999. 336 с.
5. Кононенко А.Ф., Халезов А.Д., Чумаков В.В. *Принятие решений в условиях неопределенности*. М.: ВЦ АН СССР, 1991. 196 с.
6. Петросян Л.А., Кузютин Д.В. *Устойчивые решения позиционных игр*. СПб.: СПбГУ, 2008. 327 с.
7. Петросян О.Л. *Модель многошаговых игр с учетом времени реализации альтернативы* // МТИиП, 2015. Т.7. Вып.2. С.49–68.
8. Таха Х.А. *Введение в исследование операций*. М.: Издательский дом «Вильямс», 2001. 912 с.
9. Угольницкий Г.А. *Иерархическое управление устойчивым развитием*. М.: ФИЗМАТЛИТ, 2010. 336 с.
10. Шубик М. *Настоящее и будущее теории игр* // МТИиП, 2012. Т.4. Вып.1. С.93–116.
11. Harsanyi J.C. *Games with Incomplete Information Played by “Bayesian” Players. Part I, II and III* // Management Science. 1967, 1968. Vol.14. P.159–182, 320–334 and 486–502.

ON THE PROBLEM OF SOLVING MULTIMOVE GAMES UNDER TIME DEFICIT

Andrey V. Chernov, Nizhnii Novgorod State University, Nizhnii Novgorod State Technical University, Cand.Sc., associate professor (chavnn@mail.ru).

Abstract: On example of antagonistic multimove game we consider the problem of choice by a first player of optimal by possibility strategy of behaviour under deficit of time available for him to make this choice. We suppose that each player makes his move in turn and before each move time available is insufficient to construct all the game tree from this move but it is sufficient to construct a subtree consisting of a limited amount of arcs. Thus we have the problem of a best choice of constructing strategy for this subtree and corresponding restriction of a subsequent game. We suppose that after the subtree pointed out above is constructed each player chooses his move on the base of calculations with the help of Kuhn algorithm for a game on this subtree. We consider two ways for constructing the subtree. The first one is naive way based on constructing a full tree for “uniform” restriction of the game to lesser amount of moves. The second one is based on probabilistic selection for subsequent branching of most “perspective” arcs with origin at current vertex. We prove that the first way can lead to arbitrary big miscalculations of a player. As to the second way we prove that the first player using it, in spite of restricting his prior possibilities, forces the opponent player to act in the frame of a subgame selected and calculated for essentially larger amount of moves by the first player and this selection is made as expected best (in the probabilistic sense). Theoretical (obvious enough) reasoning is confirmed by some concrete example of antagonistic game on a tree graph and also by work results of an author computer program realizing the draughts game.

Keywords: multimove game under time deficit, Kuhn algorithm, expected best choice of game subtree.