

УДК 004.023

Марченко О.І., к.т.н., доцент, Марченко О.О., аспірант, Орлова М.М., к.т.н., доцент
Національний технічний університет України «Київський політехнічний інститут»

СТРУКТУРА ТА КРИТЕРІЇ КЛАСИФІКАЦІЇ СПОСОБІВ РЕАЛІЗАЦІЇ ТА ПОКРАЩЕННЯ ПОШУКУ ПО ДЕРЕВУ МЕТОДОМ МОНТЕ-КАРЛО

Марченко О.І., Марченко О.О. Структура та критерії класифікації способів реалізації та покращення пошуку по дереву методом Монте-Карло. У статті, на основі отриманого з різних джерел великого обсягу інформації про пошук по дереву методом Монте-Карло (MCTS), особливості його поведінки при використанні в різних випадках та реалізації цього методу у багатьох застосунках, пропонується структура та критерії класифікації різних способів реалізації методу MCTS, його покращення та паралелізації. Розглянутий підхід до класифікації як для суто теоретичних способів покращення та налагодження методу MCTS для конкретних ситуацій, так і для способів пришвидшення роботи програм, що реалізують цей метод, за допомогою апаратних засобів. Передбачається, що запропоновані структура та критерії класифікації можуть бути використані для систематизації знань про метод MCTS, детальної класифікації способів реалізації цього методу та виявлення нових можливостей його покращення.

Ключові слова: задачі штучного інтелекту, дерева ігор, пошук в дереві, метод Монте-Карло, MCTS, способи покращення MCTS.

Марченко А.И., Марченко А.А. Структура и критерии классификации способов реализации и улучшения поиска по дереву методом Монте-Карло. У статье, на основе полученного из разных источников большого объема информации о поиске по дереву методом Монте-Карло (MCTS), особенностях его поведения при использовании в разных случаях и реализации этого метода во многих приложениях, предлагаются структура и критерии классификации разных способов реализации метода MCTS, его улучшения и параллелизации. Рассмотрен подход к классификации как сугубо теоретических способов улучшения и настройки метода MCTS для конкретных ситуаций, так и способов ускорения работы программ, что реализуют этот метод, при помощи аппаратных средств. Предполагается, что предложенные структура и критерии классификации могут быть использованы для систематизации знаний о методе MCTS, детальной классификации способов реализации этого метода и выявления новых возможностей его улучшения.

Ключевые слова: задачи искусственного интеллекта, деревья игр, поиск в дереве, метод Монте-Карло, MCTS, методы улучшения MCTS.

Marchenko O.I., Marchenko O.O. Structure and criteria for classification of techniques for implementation and improvement of Monte-Carlo tree search. Basing on large information taken from various sources about Monte-Carlo tree search (MCTS) method, specific features of its behavior for various cases of usage, and implementation of this method for many applications, the article proposes structure and criteria for classification of various techniques for MCTS method implementation, improvement, and parallelization. An approach to classification of pure theoretical techniques for MCTS improving and tuning in particular situations as well as techniques for hardware speedup of programs, which implement the method, is discussed. It is supposed that the structure and criteria for classification can be used for systematization of knowledge about MCTS method, detailed classification of implementation techniques of the method, and discovering of new possibilities for its improvement.

Keywords: artificial intelligence tasks, game trees, tree search, Monte-Carlo method, MCTS, MCTS improvement methods.

Постановка наукової проблеми. Однією з найбільш поширених форм подання інформації в задачах штучного інтелекту є форма у вигляді дерев послідовних рішень, а одним з методів штучного інтелекту, що був запропонований відносно недавно і призначений для виконання швидкого пошуку правильних рішень у дереві інформації, є метод пошуку в дереві з використанням методу Монте-Карло (Monte Carlo Tree Search – MCTS) [2, 3]. Задачі штучного інтелекту відзначаються надзвичайно великим обсягом даних і високими вимогами до швидкості їх обробки. Характерними і популярними задачами штучного інтелекту, на яких, як правило, виконується апробація нових методів, алгоритмів, структур даних є складні інтелектуальні ігри, наприклад, такі як ігри в шахи та Го.

MCTS є методом пошуку оптимальних рішень у заданій області за допомогою випадкових значень із заданого простору значень і побудови дерева пошуку за отриманими результатами. Цей метод вже встиг суттєво вплинути на розв'язання задач штучного інтелекту, а особливо складних ігрових задач (шахи, Го, тощо).

Аналіз досліджень. За час існування методу MCTS було створено достатньо багато його варіантів та модифікацій з метою прискорення процесу пошуку. Одні з цих модифікацій зроблені на алгоритмічному рівні MCTS, в той час як інші – на рівні розпаралелення процесу пошуку. Цьому методу присвячено вже багато досліджень і публікацій, на декілька з них автори посилаються в даній

роботі. Найбільш ґрунтовною узагальнюючою роботою з методу MCTS є огляд 2012 року, підготовлений десятима провідними вченими під керівництвом Камерона Броуне, що досліджують цей метод [3], але і в ній також відсутня строга ієрархічна класифікація способів реалізації MCTS.

Автори вважають, що систематизація інформації про різні підходи до прискорення пошуку методом MCTS та знаходження нових критеріїв для виконання такого прискорення дозволять більш чітко виявити ще не використані можливості пошуку цим методом.

Метою даної роботи є з'ясування структури та критеріїв для класифікації та систематизації інформації про способи реалізації методу пошуку в дереві методом Монте-Карло (MCTS) та особливостей його поведінки при використанні в різних випадках.

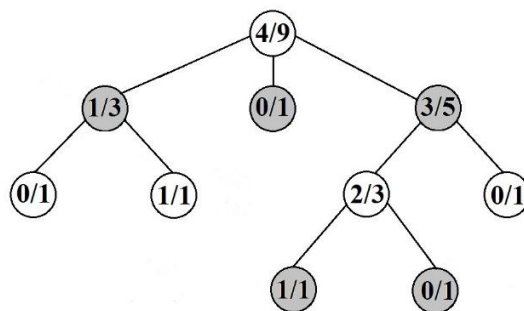
Пошук в дереві з використанням методу Монте-Карло (MCTS). Моделювання методом Монте-Карло [1] є стохастичним і за допомогою повторень такого моделювання можна отримати корисну інформацію. Особливо добре таке моделювання вирішує проблеми, у яких немає відомого детермінованого рішення, і тому його було вирішено застосувати також до рішення таких складних задач як моделювання ігор в шахи та Го.

Одне моделювання гри методом Монте-Карло виконує процес моделювання від деякої поточної позиції до кінця гри, вибираючи ігрові ходи випадковим чином. Як тільки моделювання досягає кінця гри, виконується обчислення балів для кожного гравця та встановлюється переможець. Процес моделювання починаючи від однієї й тієї самої позиції виконується багато разів, після чого відношення числа вигравів до числа програшів формує сумарну оцінку цієї позиції.

Незважаючи на те, що таке моделювання використовує тільки правила відповідної гри (шахи, Го, тощо) для вибору наступних ходів, за допомогою повторень воно в змозі надати корисну інформацію відносно того, який хід є найкращим в даній позиції [3, 4].

Для покращення ігрової потужності метод Монте-Карло був застосований до дерева пошуку однієї з ігор, і, в результаті, був отриманий метод пошуку в дереві з використанням методу Монте-Карло – Monte-Carlo Tree Search (MCTS). Вперше MCTS був застосований в комп'ютерній грі Го [3]. З того часу MCTS отримав широке використання, особливо в галузі загально-ігрових задач – General Game Playing (GGP) [5].

Нове під-дерево MCTS асоціюється з поточною позицією гри як коренева вершина, а всі вершини в цьому під-дереві будуть зберігати деяку кількість вигравів і програшів [3]. На рисунку 1 наведено приклад дерева MCTS, де в кожній вершині показано результат багатократного моделювання гри від цієї вершини. Всі термінальні (листові) вершини і нетермінальні вершини, до яких зберігається доцільність додавання нових листів-нащадків, формують межу (передній фронт) дерева



MCTS.

Рис.1. Приклад дерева MCTS

Загальний підхід методу MCTS. З метою більш глибокого аналізу особливостей методу MCTS розглянемо його загальний підхід у більш формалізованому вигляді.

Як було сказано, дерево пошуку будується шляхом додавання до нього кожен раз по одній вершині відповідно до результатів моделювання гри від заданої вершини до закінчення гри, а загальний підхід методу MCTS складається з чотирьох повторюваних кроків, показаних на рис.2 [3].

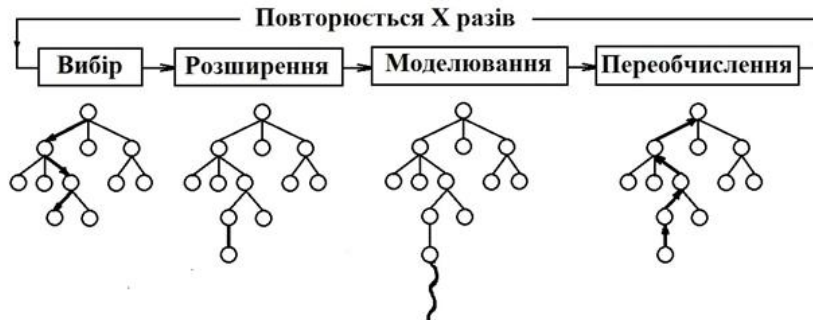


Рис.2. Загальний підхід методу MCTS

Крок 1. Вибір. Починаючи від поточного кореневого вузла R і використовуючи задану політику дерева (Tree Policy), вибираються оптимальні вершини-нащадки (пояснюється нижче) доки не буде досягнута довільна вершина-лист L.

Крок 2. Розширення. Якщо L є нетермінальною вершиною (тобто на ній гра ще не закінчується), то додається нова вершина V, тобто відбувається спроба виконати черговий хід згідно правил гри. Черговий хід із множини можливих в даному стані ходів береться згідно політики розширення (Expansion Policy), якою, як правило, є взяття випадкового ходу.

Крок 3. Моделювання. Починаючи від взятої вершини V виконується процес моделювання гри до досягнення результату Res згідно прийнятої політики моделювання (Simulation Policy). Найчастіше береться найпростіша політика моделювання, яка називається політикою за умовчанням (Default Policy) і яка полягає у взятті всіх наступних ходів випадковим чином.

Крок 4. Переобчислення. Враховуючи отриманий результат моделювання Res у зворотному порядку від вершини L до кореневої вершини R виконується корекція значень атрибутів у вершинах поточної послідовності ходів згідно політики зворотного переобчислення (Backpropagation Policy).

Як результат однократного виконання функції пошуку, повертається одна з нових вершин, отриманих під час процесу пошуку від кореня R, значення результату якої є найкращим.

Більш формально загальний підхід методу MCTS можна записати у вигляді функції пошуку MCTS_Search [3].

```
function MCTS_Search (S) {  
  Create (R, S);  
  while WithinComputationalBudget do {  
    L = TreePolicy (R);  
    V = ExpansionPolicy (L);  
    Res = SimulationPolicy (V);  
    BackpropagationPolicy (L, Res);  
  }  
  return BestChild (R);  
}
```

MCTS_Search (S) – функція пошуку в дереві методом MCTS, яка приймає у якості параметра поточний стан гри (позицію) S і повертає найкращу з нових, отриманих під час моделювання, вершин дерева, тобто новий найкращий хід від даної позиції.

Create (R, S) – функція створення нової кореневої вершини R для заданого поточного стану гри S.

WithinComputationalBudget – булева функція, яка повертає значення True, якщо обчислювальний бюджет (заданий час чи задана кількість повторень) ще не вичерпався.

TreePolicy (R) – функція, яка реалізує дії першого кроку методу MCTS.

ExpansionPolicy (L) – функція, яка реалізує дії другого кроку методу MCTS.

SimulationPolicy (V) – функція, яка реалізує дії третього кроку методу MCTS.

BackpropagationPolicy (L, Res) – функція, яка реалізує дії четвертого кроку методу MCTS.

BestChild (R) – функція, яка визначає і повертає найкращу з нових, отриманих під час моделювання від кореня R, вершин дерева, тобто новий найкращий хід від даної позиції.

Побудова довільного дерева MCTS (як правило, дерева гри) виконується інкрементальним і асиметричним способом. На кожній ітерації методу MCTS використовується політика дерева для знаходження найкращої для продовження гри вершини у поточному дереві. При розробці політики дерева намагаються збалансувати певним чином *кількість досліджень (exploration)*, виконуючи вибір нової вершини у тих частинах дерева, які були ще недостатньо добре опрацьовані, та *кількість використань (exploitation)*, виконуючи вибір нової вершини у тих частинах дерева, які вже є на даний момент найбільш перспективними. Після цього виконується моделювання (випадкова або статистично розподілена послідовність ходів гри до її закінчення), починаючи від обраної вершини, а потім значення деяких вершин дерева корегуються згідно з отриманим результатом. Під час такого процесу додається одна вершина-нащадок, яка відповідає ходу, виконаному від обраної вершини, а корекція значень вершин відбувається у її вершинах-предках. Нагадаємо, що під час моделювання ходи виконуються згідно політики моделювання, яка у найпростішому випадку за умовчанням (Default Policy) є політикою взяття однаково випадкових ходів.

Структура та критерії класифікації способів реалізації методу MCTS. Згідно розглянутого вище загального підходу метода MCTS логічним буде класифікування способів реалізації цього метода за критерієм покращення кожного з етапів роботи MCTS. З іншої сторони, в попередні роки було з'ясовано, що в прикладенні до гри Го, за наявності достатньої обчислювальної потужності, MCTS працює набагато краще, ніж інші методи, в результаті чого, цей метод став парадигмою для комп'ютерної гри Го. Крім того, в [3] було зазначено, що з нескінченим часом метод MCTS буде давати сходження до оптимальної гри, і що зі збільшенням числа моделювань, ефективність пошуку методом MCTS буде покращуватись [6]. Для збільшення числа моделювань потрібно або збільшувати час виконання моделювання або збільшувати швидкість моделювань. Це може бути досягнуто за допомогою паралелізації пошуку та обчислень. Відповідно до цього, виконання паралелізації є ще одним критерієм класифікації методів та способів реалізації MCTS. В свою чергу, за критерієм орієнтації способу паралелізації на конкретні апаратні ресурси, чи відсутності такої орієнтації, існуючі способи паралелізації можуть бути поділені на апаратно-незалежні способи та апаратно-орієнтовані способи.

Згідно розглянутих критеріїв структура класифікації буде такою, як показано на рисунку 3.

MCTS є загальним методом пошуку в дереві з використанням методу Монте-Карло, якому відповідає ціла множина досить різних алгоритмів, які відрізняються використаними політиками дерева, розширення, моделювання та зворотного переобчислення.

В залежності від використаної політики дерева, тобто вибору вершини в дереві для подальшого моделювання, виділяють наступні групи алгоритмів, що відповідають загальному методу MCTS [3]:

- Plain UCT (Upper Confidence bounds for Trees): прямий спосіб реалізації згідно методу MCTS з політикою вибору дерева, у якій використовується метод верхніх довірчих границь UCB (Upper Confidence Bounds) у вигляді, як початково було запропоновано в [4];
- UCT: UCT-орієнтоване під-сімейство способів реалізації MCTS з політикою вибору дерева, у якій використовується довільний варіант реалізації методу верхніх довірчих границь UCB;
- Non-UCT: сімейство способів реалізації, що працюють згідно наведеного вище загального підходу до пошуку в дереві з використанням методу Монте-Карло з довільною політикою вибору дерева, але не використовують UCT.

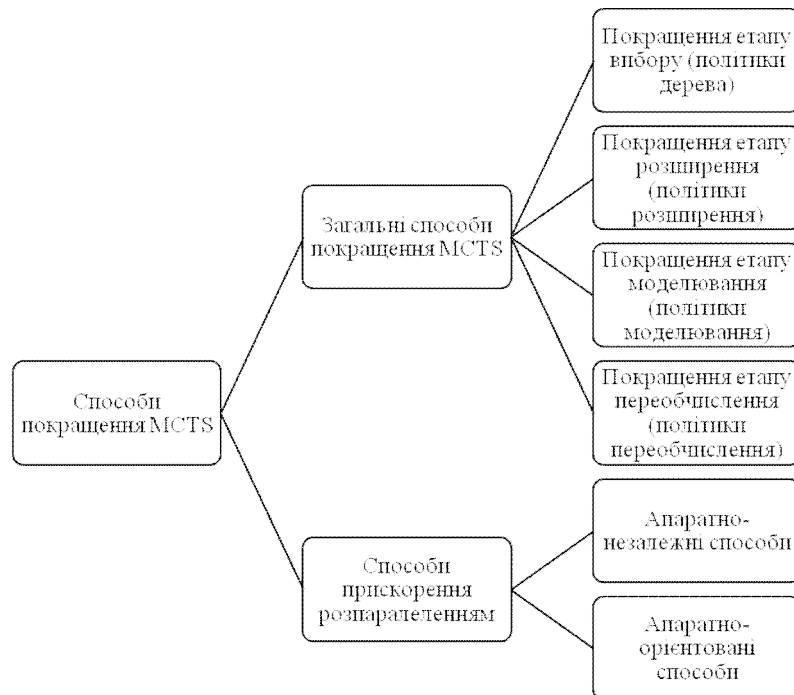


Рис. 3. Структура класифікації способів покращення MCTS

Аналогічно існують також різні варіанти реалізації MCTS, які орієнтовані на покращення інших етапів роботи MCTS. Це може бути предметом подальшої деталізації загальних способів покращення MCTS у запропонованій класифікації.

З часу створення методу MCTS пошуки кращих варіантів його реалізації відбувались, в основному, серед сімейства UCT-орієнтованих способів. Але зовсім нещодавно, у січні 2015 року, Трістаном Казенаве [7] був запропонований новий перспективний підхід, який він назвав SHOT (Sequential Halving applied to Trees), і який належить до Non-UCT сімейства способів MCTS. Зрозуміло, що дослідження покращення/пришвидшення цього способу реалізації MCTS ще тільки починаються.

Результуюча структура класифікації способів покращення етапу вибору (політики дерева) показана на рис.4.

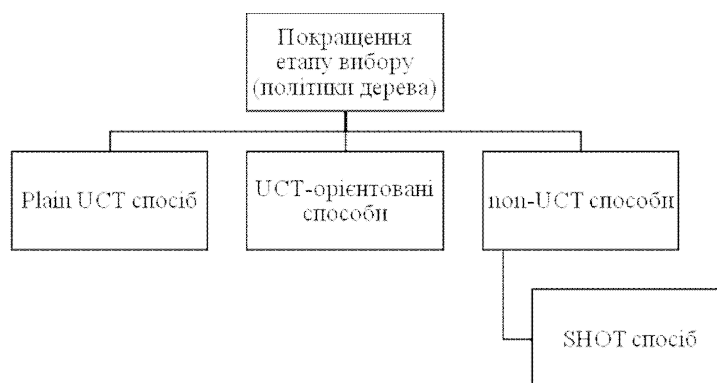


Рис. 4. Структура класифікації способів покращення етапу вибору (політики дерева)

На сьогоднішній день для розпаралелення пошуку в дереві методом MCTS на апаратно-незалежному рівні були запропоновані три загальних підходи, які були названі: листкова паралелізація, коренева паралелізація і деревна паралелізація [3].

Ці підходи показані на рисунках 5, 6, 7а, 7б.

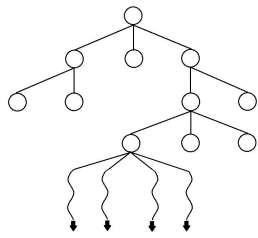


Рис.5. Листкова паралелізація

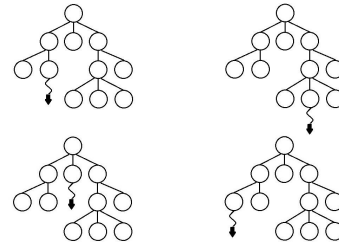


Рис.6. Коренева паралелізація

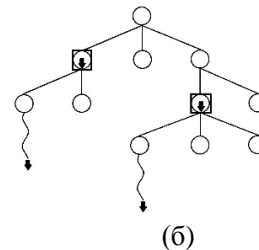
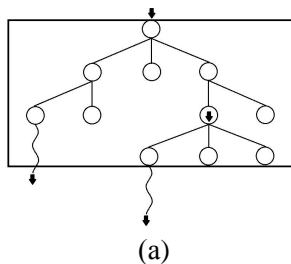


Рис.7. Деревна паралелізація з глобальним м'ютексом (а) та з локальними м'ютексами (б)

Розглянемо кожен з цих підходів паралелізації більш детально.

Листкова паралелізація як визначено в [8] – це виконання множини одночасних моделювань від вершини-листа, яка була додана після закінчення процесу пошуку актуальної вершини, використовуючи політику дерева. Ідея полягає у тому, щоб отримати кращу статистику на кожній вершині-листу за допомогою кращої початкової оцінки. В [9] таку схему назвали паралелізацією на листках (at-the-leaves parallelization). Пізніше вкоренився термін «листова паралелізація» (leaf parallelisation).

Кореневу паралелізацію [8] іноді називають багато-деревним MCTS, оскільки декілька дерев пошуку MCTS будуються одночасно (тобто паралелізуються на корені дерева). Як правило, для повідомлення про те, який хід був обраний алгоритмом у кожному з дерев, використовується інформація з першого рівня цих дерев. Однією з переваг такого підходу є те, що кожен паралельний потік може виконуватись протягом фіксованого часу і зупинений у будь-який момент.

Деревна паралелізація є процесом, який виконує декілька одночасних кроків моделювання на одному й тому ж дереві [8]. В цьому випадку виникає класична проблема паралельного програмування, що полягає у недопущенні одночасного доступу до дерева з різних потоків. Для цього потрібно забезпечити ексклюзивний доступ до певного під-дерева MCTS кожному потоку, а також зробити так, щоб всі потоки працювали у різних областях дерева, доки м'ютекс не буде звільнений. В одній схемі, запропонованій в [8], використовується глобальний м'ютекс на кореневій вершині (рис.3а). Такий підхід є резонним у тому випадку, якщо час виконання моделювання є набагато довшим, ніж час вибору вершини з дерева, оскільки один потік може проходити по дереву або робити переобчислення у дереві, в той час, як інші потоки виконують моделювання. В іншій схемі (рис.3б) використовуються локальні м'ютекси на кожній внутрішній вершині, яка блокується і звільняється кожен раз, коли якийсь потік приходить у цю вершину.

Як зазначалося вище, розробка, дослідження і паралелізація SHOT-орієнтованих способів є перспективними напрямками.

Відповідно до розглянутих вище критеріїв структура класифікації апаратно-незалежних способів реалізації MCTS має вигляд, показаний на рисунку 8.

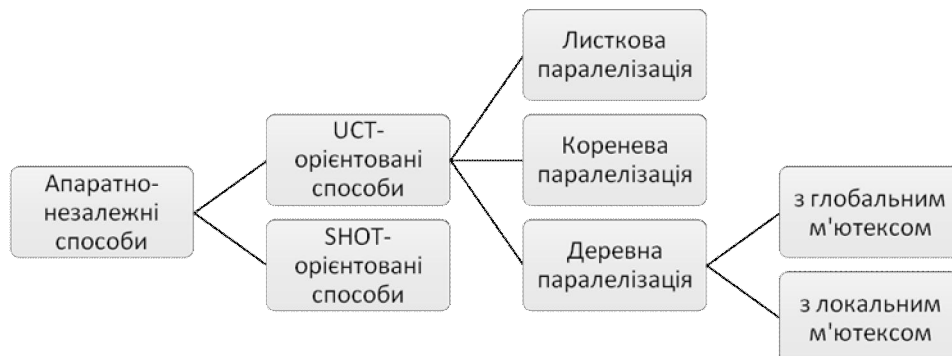


Рис.8. Структура класифікації апаратно-незалежних способів реалізації MCTS.

На апаратно-орієнтованому рівні покращення MCTS вже використовувався широкий спектр універсальних, спеціалізованих та змішаних (гібридних) апаратних платформ і мультикомп'ютерних систем. Тому початок цієї гілки класифікації може виглядати як показано на рисунку 9.

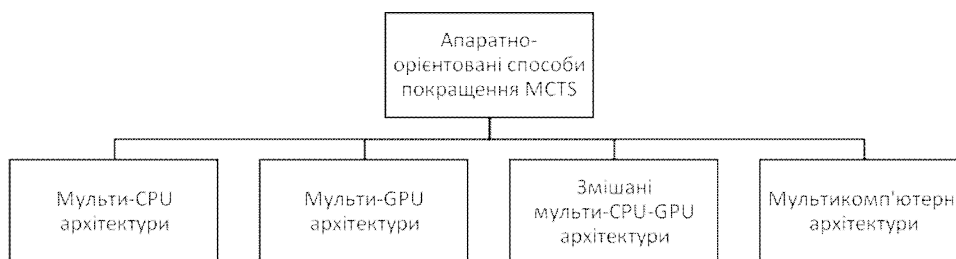


Рис. 9. Структура класифікації апаратно-орієнтованих способів реалізації MCTS.

Деталізація апаратно-орієнтованих способів паралелізації MCTS є напрямом подальших досліджень.

Висновки. Запропонована критерії та структура класифікації способів реалізації пошуку в дереві методом MCTS. Отримані результати дозволяють виконати систематизацію інформацію як про вже існуючі способи реалізації MCTS, так і виявити ще не дослідженні шляхи та варіанти реалізації цього методу. Подальшим напрямком дослідження може бути як подальша деталізація запропонованої структури класифікації, так власне класифікація існуючих способів реалізації методу MCTS згідно даних критеріїв та структури враховуючи особливості його поведінки при використанні в різних випадках.

1. Метод Монте-Карло [Електронний ресурс]. Режим доступу: http://uk.wikipedia.org/wiki/Метод_Монте-Карло
2. Monte Carlo tree search [Електронний ресурс]. Режим доступу: http://en.wikipedia.org/wiki/Monte-Carlo_tree_search.
3. Cameron Browne and others. A Survey of Monte Carlo Tree Search Methods // IEEE Trans. on Computational Intelligence and AI in Games. — vol. 4. — no. 1. — March 2012.
4. L. Kocsis and C. Szepesvarri, "Bandit based Monte-Carlo Planning," in Euro. Conf. Mach. Learn. Berlin, Germany: Springer, 2006, pp. 282–293.
5. J. Mehat and T. Cazenave, "Combining UCT and nested monte carlo search for single-player general game playing," IEEE Transactions on Computational Intelligence and AI in Games, vol. 2, no. 4, p. 271, 2010.
6. R. B. Segal, "On the scalability of parallel UCT," Lecture Notes in Computer Science, vol. 6515/2011, pp. 36–47, 2011.
7. T. Cazenave, "Sequential Halving Applied to Trees," IEEE Transactions on Computational Intelligence and AI in Games, vol. 7, no. 1, pp. 102-105, 2015.
8. G. M. J.-B. Chaslot, M. H. M. Winands, and H. J. van den Herik. Parallel Monte-Carlo Tree Search // Proc. Comput. And Games, LNCS 5131, Beijing, China. — 2008, pp. 60–71.
9. T. Cazenave and N. Jouandea. On the Parallelization of UCT // Proc. Comput. Games Workshop, Amsterdam, Netherlands. — 2007, pp. 93–101.