

DOI: <https://doi.org/10.36910/6775-2524-0560-2024-55-17>

УДК 004.75; 004.02

Марченко Олексій Олександрович, асистент

<https://orcid.org/0000-0002-5080-4811>

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

МЕТОД РОЗПАРАЛЕЛЕННЯ ПОШУКУ ПО ДЕРЕВУ МЕТОДОМ MCTS

Марченко О.О. Метод розпаралелення пошуку по дереву методом MCTS. Метод, який пропонується в цій статті, ґрунтується на двох напрямках подальшого розвитку пошуку по дереву методом Монте-Карло (MCTS): напрямку вдосконалення теоретичних підходів цього методу і напрямку розпаралелення процесу пошуку цим методом. Розпаралелення пошуку MCTS виконується на основі раніше запропонованих авторському варіанті MCTS-TSC (Monte-Carlo Tree Search with Tree Shape Control) з контролем форми дерева пошуку, узагальненій графово-структурній гетерогенній моделі динамічного розпаралелення пошуку по дереву методом Монте-Карло, а також моделі ресурсів гетерогенної розподіленої комп'ютерної системи з локальними зв'язками та її графу. Запропонований метод розпаралелення враховує всі три відомі методи апаратно-незалежного розпаралелення MCTS: кореневе, деревне і листкове розпаралелення і складається із семи етапів. Принциповою відмінністю запропонованого метода від методів інших дослідників є більш інтелектуальне прийняття рішення про вид розпаралелення та степінь розпаралелення у вершинах дерева пошуку на етапах 1-3, яке приймається як на основі наявних апаратних ресурсів, так і на основі поточної форми дерева пошуку, щоб спрямувати подальшу побудову дерева у бажаному напрямку (в глибину, чи в ширину), в той час, як іншими дослідниками, як правило, виконується тільки достатньо жорстко прив'язана до конкретної конфігурації обчислювальної системи реалізація кореневого розпаралелення між декількома комп'ютерами, реалізація деревного розпаралелення чи реалізація листового розпаралелення на платах GPU. Запропонований метод розпаралелення пошуку методом MCTS забезпечує більш ефективне завантаження наявних апаратних ресурсів і дозволяє як прискорити процес пошуку в цілому, так і підвищити ефективність пошуку методом MCTS.

Ключові слова: пошук в дереві методом Монте-Карло, MCTS, методи розпаралелення MCTS, паралельні та розподілені комп'ютерні системи, модель ресурсів комп'ютерної системи.

Marchenko O. Method of the MCTS Tree Search Method Parallelization. The method proposed in this article is based on two directions of further development of Monte Carlo tree search method (MCTS): the direction of improving the theoretical approaches of this method and the direction of parallelization of the search process by this method. Parallelization of MCTS search is performed on the basis of previously proposed author's variant MCTS-TSC (Monte-Carlo Tree Search with Tree Shape Control) with search tree shape control, a generalized graph-structural heterogeneous model of dynamic parallelization of Monte Carlo tree search, as well as a resource model of a heterogeneous distributed computer system with local connections and its graph. The proposed parallelization method takes into account all three known MCTS hardware-independent parallelization methods: root, tree, and leaf parallelization and consists of seven stages. The fundamental difference between the proposed method and the methods of other researchers is a more intelligent decision-making about the type of parallelization and the degree of parallelization at the vertices of the search tree at stages 1-3, which is made both on the basis of available hardware resources and on the basis of the current shape of the search tree in order to direct further building a tree in the desired direction (in depth or width), while other researchers, as a rule, perform only the implementation of root parallelization between several computers, which is strictly tied to a specific configuration of the computer system, the implementation of tree parallelization or implementation of leaf parallelization on GPU boards. The proposed method of MCTS search parallelization ensures more efficient loading of available hardware resources and allows both speeding up the search process as a whole and increasing the efficiency of MCTS search.

Keywords: Monte-Carlo tree search method, MCTS, MCTS parallelization methods, parallel and distributed computer systems, resource model for computer system.

Постановка наукової проблеми.

Задачі штучного інтелекту оперують дуже великими обсягами даних, обробка яких потребує значного часу, що часто перевищує ліміти часу, який є доцільним для обробки. Тому, напрями досліджень, які пропонують ефективні розв'язки таких задач є актуальними і залишаються такими надалі, оскільки обсяги інформації у світі, які потрібно оброблювати, постійно зростають.

Для розв'язку зазначених задач наразі широко використовується пошук по дереву інформації методом Монте-Карло (Monte Carlo Tree Search – MCTS) [1]. Цей метод показав себе з найкращої сторони при розв'язку багатьох надскладних задач штучного інтелекту, однією з яких є найскладніша з існуючих ігор – гра Го. Але, незважаючи на його теоретичну потужність, сам по собі цей метод не забезпечує достатньої швидкості обробки інформації, обсяг якої часто є надзвичайно великим. Тому, подальшому розвитку цього методу було присвячено багато досліджень, причому ці дослідження виконувалися як з точки зору вдосконалення його теоретичних засад, так і з точки зору розробки його варіантів, орієнтованих на певні класи задач, так і з точки зору розпаралелення роботи пошуку MCTS для прискорення процесу пошуку. По всіх цих напрямках вже були досягнуті вагомні успіхи, але й, як зазначалося вище, обсяги інформації, яку потрібно оброблювати, також зростають,

що вимагає ще більшої швидкості та ефективності обробки інформації. Тому продовження досліджень по всіх зазначених напрямках залишається актуальним, зокрема й по напрямку розпаралелення MCTS, чому й присвячена ця стаття.

Аналіз досліджень.

Метод, який пропонується в цій статті ґрунтується на двох напрямках подальшого розвитку методу пошуку по дереву MCTS: напрямку вдосконалення теоретичних підходів цього методу і напрямку розпаралелення процесу пошуку цим методом. Відомий, можна сказати вже класичний, огляд існуючих варіантів реалізації MCTS та підходів до його подальшого розвитку, як ефективних модифікацій основної формули UCS1 методу MCTS, так методів розпаралелення, зроблених в роботі [1]. Власний погляд на структуру та взаємозв'язки між підходами до підвищення ефективності пошуку MCTS був запропонований у вигляді класифікації [2]. В роботі [3] була запропонована авторська ідея контролю форми дерева пошуку під час процесу пошуку на основі критеріїв типу «глибина-ширина» DWC (Depth-Width Criteria), а також запропонований подальший розвиток методу MCTS, що базується на цій ідеї і який був названий пошуком по дереву методом Монте-Карло з контролем форми дерева (Monte-Carlo Tree Search with Tree Shape Control – MCTS-TSC) [3].

Запропонований в цій статті метод розпаралелення ґрунтується саме на варіанті MCTS-TSC, а також на раніше запропонованих узагальненій графово-структурній гетерогенній моделі динамічного розпаралелення пошуку по дереву методом Монте-Карло [4] та моделі ресурсів гетерогенної розподіленої комп'ютерної системи з локальними зв'язками (ГРКСЛЗ) та її графу [5]. Крім авторських досліджень, для розробки запропонованого методу розпаралелення є важливими дослідження, виконані в роботах [1, 6-12].

Як відомо, для апаратно-незалежного розпаралелення MCTS були запропоновані три основних методи: кореневе розпаралелення, деревне розпаралелення і листкове розпаралелення, що можуть бути застосовані на різних етапах процесу пошуку MCTS, який складається з чотирьох етапів [1]: 1) етапу вибору; 2) етапу розширення; 3) етапу моделювання; 4) етапу переобчислення.

Кореневе розпаралелення [6, 7] передбачає побудову декількох дерев пошуку MCTS від поточного кореня паралельно. Реалізація такого розпаралелення є доцільною або на окремих комп'ютерах обчислювального кластера, або на дуже потужних комп'ютерах з великою кількістю ядер.

Деревне розпаралелення полягає у паралельному виконанні кількох ітерацій процесу пошуку загальної схеми MCTS на одному й тому ж дереві пошуку [8, 9, 10]. Головною проблемою при деревному розпаралеленні є можливість виникнення конфліктів одночасного доступу до вершин дерева пошуку з різних потоків. Ця проблема розв'язується встановленням або глобального м'ютекса на всьому дереві, або локальних м'ютексів на кожній вершині дерева пошуку.

Листкове розпаралелення [9] полягає у паралельному виконанні множини ітерацій етапу моделювання загальної схеми MCTS, які виконуються від нової доданої листкової вершини дерева пошуку. Потрібно зазначити, що листкове розпаралелення може бути реалізоване ефективно як на центральних багатоядерних процесорах CPU, так і на графічних платах GPU.

У роботах [11, 12] пропонуються цікаві варіанти розпаралелення MCTS, в яких для планування розподілу окремих дій процесу пошуку MCTS між апаратними ресурсами обчислювальної системи використовується різного виду таблиці.

Зазначимо, що дослідники розпаралелення методу пошуку MCTS [6-12] спрямовували свої дослідження та реалізації на окремі аспекти розпаралелення: на реалізацію кореневого, деревного чи листкового розпаралелення на певній комп'ютерній системі, на залежності прискорення пошуку від кількості процесорів та ядер, на варіанти реалізації листкового розпаралелення на GPU тощо. Але комплексний підхід на основі теоретичних моделей в цих дослідженнях відсутній.

Метою даної роботи розробка нового методу розпаралелення MCTS, який ґрунтується на раніше запропонованих моделі динамічного розпаралелення пошуку по дереву методом Монте-Карло [4], моделі ресурсів ГРКСЛЗ та її графу [5] і подальшого розвитку методу пошуку по дереву MCTS на основі контролю форми дерева пошуку, базуючись на критеріях типу «глибина-ширина» [3], що забезпечить більш ефективний розподіл наявних апаратних ресурсів для розпаралелення пошуку методом MCTS на різних етапах його роботи та прискорить процес пошуку в цілому.

Метод розпаралелення пошуку по дереву методом MCTS

В класифікації [2], зазначено, що серед методів розпаралелення MCTS можна виділити дві

групи методів: апаратно-незалежні методи та апаратно-орієнтовані методи. Запропонований метод є гібридним, оскільки об'єднує головні підходи апаратно-незалежного розпаралелення MCTS з орієнтацією на наявні для розпаралелення в поточний момент апаратні ресурси. Зі сторони апаратно-незалежних методів розпаралелення він охоплює всі відомі методи: кореневе розпаралелення, деревне розпаралелення і листкове розпаралелення. Апаратно-залежна орієнтація цього методу полягає у тому, що він ґрунтується на узагальненій графово-структурній гетерогенній моделі динамічного розпаралелення DPM(t), яка вперше була запропонована в [4], але пізніше була вдосконалена і розширена.

Модель DPM(t) [4] визначається як

$$DPM(t) = (STM(t), RM(t), TP(t), EP(t), SP(t), BPP(t), PP(t), t),$$

де STM(t) (Search Tree Model) – модель дерева пошуку MCTS в момент часу t;

RM(t) (Resource Model) [5] – модель ресурсів ГРКСЛЗ в момент часу t;

TP(t) (Tree Policy) – політика дерева (політика вибору вершини дерева для розширення) в момент часу t;

EP(t) (Expansion Policy) – політика кроку розширення в момент часу t;

SP(t) (Simulation Policy) – політика кроку моделювання в момент часу t;

BPP(t) (BackPropogation Policy) – політика зворотного обчислення параметрів передбачення (фактично є певною формулою переобчислення) в момент часу t;

PP(t) (Parallelization Policy) – політика розпаралелення в момент часу t.

Підхід запропонованого метода полягає у співставленні моделі дерева STM(t) дерева пошуку MCTS та політик виконання кожного етапу пошуку MCTS TP(t), EP(t), SP(t), BPP(t) в момент часу t узагальненої моделі DPM(t) до графу моделі ресурсів ГРКСЛЗ RM(t) в момент часу t за допомогою політики розпаралелення PP(t) в момент часу t.

Базовий варіант застосування запропонованого методу розпаралелення показаний на рис. 1. На цьому рисунку показані перші 4 етапи методу, на яких відбувається власне процес розпаралелення.

Розглянемо запропонований метод детально.

Початковий стан. Процес пошуку MCTS в момент часу t знаходиться перед черговою ітерацією цього процесу у вершині дерева пошуку V, яка відповідає певній позиції гри (стану прикладної задачі) і від якої потрібно знайти найкращий наступний хід (найкращу подальшу дію).

Етап 1. Для кореневого вузла Node[0] (Host) 0-го рівня моделі ресурсів RM(t) в момент часу t на основі моделі піддерева STM(t) вершини V, критеріїв DWC(t), кількості та типів наявних ресурсів 1-го рівня моделі ресурсів RM(t) визначається степінь кореневого розпаралелення RPD(t) і, в залежності від типів вузлів Node[i], можливо деревного розпаралелення TPD(t) політики розпаралелення PP(t).

Після цього відбувається створення відповідної кількості паралельних потоків, які передають необхідні структури даних вузлам моделі ресурсів RM(t) наступного рівня для паралельного виконання етапів вибору та розширення процесу пошуку MCTS.

Етап 2. Для кожного вузла Node[i] 1-го рівня моделі ресурсів RM(t) в момент часу t на основі моделі піддерева STM(t) вершини V, критеріїв DWC(t), кількості та типів наявних ресурсів 2-го рівня моделі ресурсів RM(t) визначається степінь деревного розпаралелення TPD(t) і, можливо, листкового розпаралелення LPD(t) політики розпаралелення PP(t).

Далі відбувається створення відповідної кількості паралельних потоків, які на ядрах Core_i виконують різні варіанти етапів вибору та розширення дерева пошуку MCTS (тобто деревне розпаралелення) і визначення множини нових доданих вершин для виконання етапу моделювання процесу пошуку MCTS. Після цього ці потоки передають необхідні структури даних вузлам моделі ресурсів RM(t) типу GBoards для паралельного виконання етапу моделювання від нових доданих на етапі розширення вершин дерева пошуку MCTS. Крім того, частина ядер Core_i також може бути розподілена для виконання етапу моделювання.

Етап 3. Для кожного вузла типу GBoards 2-го рівня моделі ресурсів RM(t) в момент часу t на основі моделі піддерева STM(t) вершини V, критеріїв DWC(t), кількості та типів наявних ресурсів 3-го рівня моделі ресурсів RM(t) визначається степінь листкового розпаралелення LPD(t) політики розпаралелення PP(t).

Після цього для кожного ресурсу 3-го рівня типу GPU моделі ресурсів RM(t) відбувається створення відповідної кількості паралельних потоків для виконання етапу моделювання від нових

доданих на етапі розширення вершин дерева пошуку MCTS.

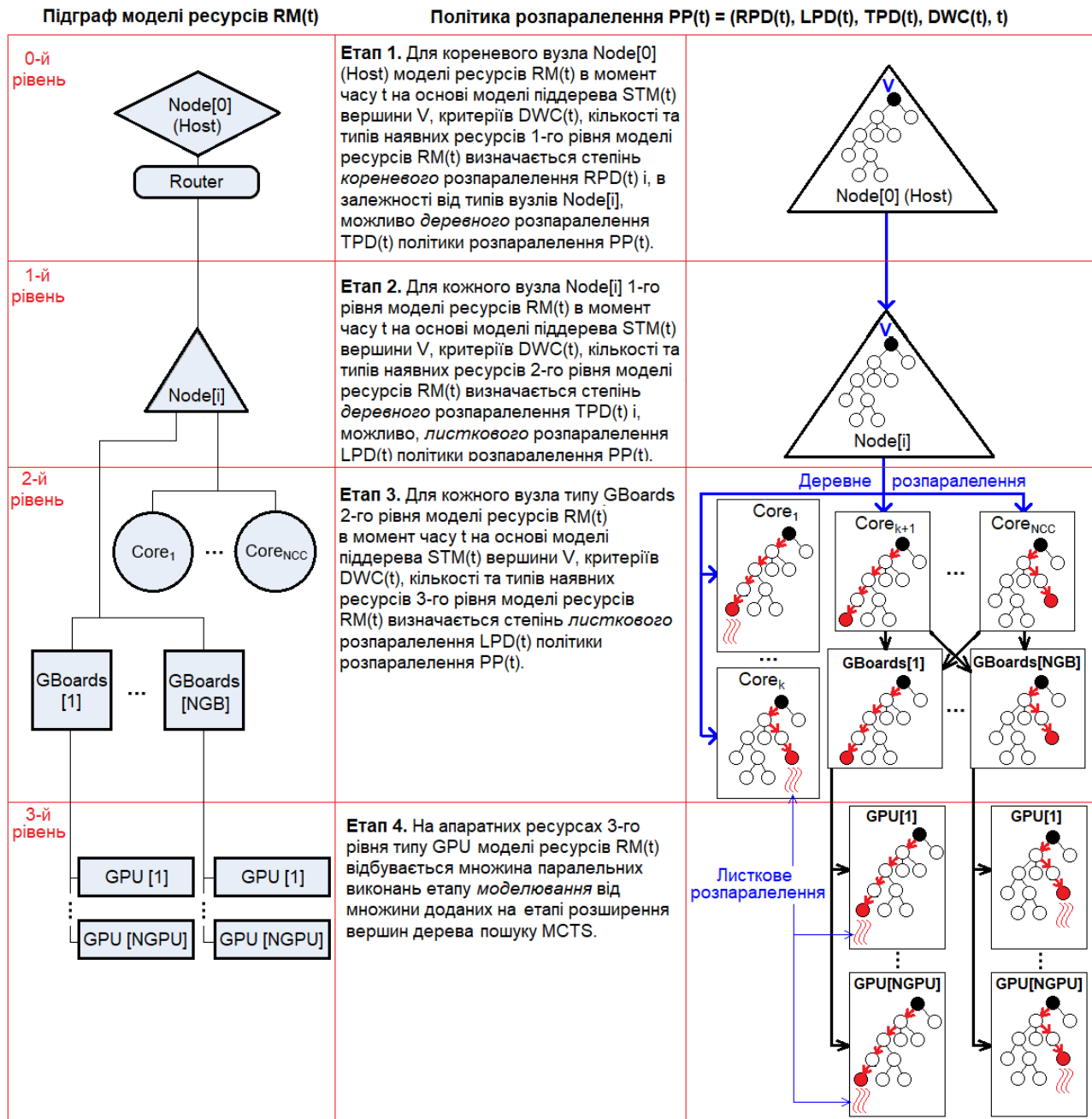


Рис.1. Метод підвищення ефективності пошуку по дереву методом MCTS розпаралеленням на основі моделі DPM(t).

Етап 4. На апаратних ресурсах 3-го рівня типу GPU моделі ресурсів RM(t) відбувається множина паралельного виконання етапу моделювання від множини доданих на етапі розширення вершин дерева пошуку MCTS.

Етап 5. На 5-му етапі методу відбувається повернення результатів виконаних моделювань з ресурсів 3-го рівня моделі ресурсів RM(t) на 2-й рівень, на якому кожен із потоків виконує етап переобчислення пошуку методом MCTS на спільному дереві пошуку так, щоб не виникало колізій з виконанням свого переобчислення іншими потоками.

Етап 6. На 6-му етапі методу відбувається повернення результуючого дерева пошуку з 2-го рівня моделі ресурсів RM(t) на 1-й, на якому це дерево об'єднується з результатами моделювання та переобчислення, виконаних на цьому рівні (це випадок, якщо на 1-му рівні було виконане як *деревне*, так і *листяне* розпаралелення). кожен із потоків виконує етап переобчислення пошуку методом

MCTS на спільному дереві пошуку так, щоб не виникало колізій з виконанням свого переобчислення іншими потоками.

Етап 7. На 7-му етапі методу відбувається повернення остаточно сформованого у вузлі Node[i] результуючого дерева пошуку з 1-го рівня моделі ресурсів $RM(t)$ на 0-й, тобто на вузол Node[0] (Host), на якому згідно з прийнятим способом оцінювання результатів пошуку в різних деревах визначається найкращий наступний хід гри (наступна дія прикладної програми) для поточної позиції (стану прикладної програми), якій відповідає вершина V дерева пошуку MCTS.

Принциповою відмінністю запропонованого метода розпаралелення MCTS від методів інших дослідників є більш інтелектуальне прийняття рішення про вид розпаралелення та степінь розпаралелення у вершинах дерева пошуку на етапах 1-3, яке приймається як на основі наявних апаратних ресурсів, так і на основі поточної форми дерева пошуку, щоб спрямувати подальшу побудову дерева у бажаному напрямку (в глибину, чи в ширину), в той час, як іншими дослідниками, як правило, виконується тільки достатньо жорстко прив'язана до конкретної конфігурації обчислювальної системи реалізація кореневого розпаралелення між декількома комп'ютерами, реалізація деревного розпаралелення чи реалізація листкового розпаралелення на платах GPU.

Висновки. На основі узагальненої графово-структурної гетерогенної моделі динамічного розпаралелення DPM(t) вперше запропоновано метод підвищення ефективності пошуку по дереву методом MCTS, який полягає у розпаралеленні процесу такого пошуку з врахуванням наявних апаратних ресурсів гетерогенної розподіленої комп'ютерної системи в кожен момент часу та використанням критеріїв виду «глибина-ширина» для прийняття рішення про вибір наступної вершини дерева пошуку.

Використання запропонованого методу динамічного розпаралелення пошуку MCTS забезпечує більш ефективне завантаження наявних апаратних ресурсів і дозволяє як досягти більш високого рівня розпаралелення та прискорити процес пошуку в цілому, так і підвищити ефективність пошуку методом MCTS.

Список бібліографічного опису

1. Cameron Browne. A Survey of Monte Carlo Tree Search Methods / Cameron Browne, Edward Powley, Daniel Whitehouse, and others // IEEE Trans. on Computational Intelligence and AI in Games. – vol. 4. – no. 1. – March 2012. – P. 1-49.
2. Марченко О.І. Класифікація способів реалізації та покращення пошуку по дереву методом Монте-Карло / Марченко О.І., Марченко О.О., Орлова М.М. // Штучний інтелект. – 2016. – №2(72). – С. 59-69.
3. Oleksandr I. Marchenko. Monte-Carlo Tree Search with Tree Shape Control. / Oleksandr I. Marchenko, Oleksii O. Marchenko // 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON). Conference Proceedings. May 29 – June 2, 2017., Kyiv, Ukraine. – 2017. – P. 812-8173.
4. Марченко О.О. Модель динамічного розпаралелення пошуку в дереві методом Монте-Карло для grid-систем. / Марченко О.О., Марченко О.І. // Системний аналіз та інформаційні технології: матеріали 19-ї Міжнародної науково-технічної конференції SAIT 2017, Київ, 22 – 25 травня 2017 р. / ННК "ІПСА" НТУУ "КПІ ім. Ігоря Сікорського". – К.: ННК "ІПСА" НТУУ "КПІ ім. Ігоря Сікорського", 2017., с.213-214. – Текст: укр.
5. Марченко О.О. Модель ресурсів неоднорідної розподіленої комп'ютерної системи з локальними зв'язками та її граф. / Марченко О.О., Марченко О.І. // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2020. – № 39. – с.83-88.
6. T. Cazenave. On the Parallelization of UCT / T. Cazenave and N. Jouandeau // Proc. Comput. Games Workshop, Amsterdam, Netherlands. – 2007. – P. 93–101.
7. Y. Soejima. Evaluating Root Parallelization in Go / Y. Soejima, A. Kishimoto, and O. Watanabe // IEEE Trans. Comp. Intell. AI Games. – vol. 2. – no. 4. – 2010. – P. 278–287.
8. E. Steinmetz. More Trees or Larger Trees: Parallelizing Monte Carlo Tree Search / E. Steinmetz, M. Gini // IEEE Transactions on Games, vol. 13, No. 3. – 2021. – P. 315–320.
9. G. M. J.-B. Chaslot. Parallel Monte-Carlo Tree Search / G. M. J.-B. Chaslot, M. H. M. Winands, and H.J. van den Herik // Proc. Comput. And Games, LNCS 5131, Beijing, China. – 2008. P.60–71.
10. A. Bourki. Scalability and Parallelization of Monte-Carlo Tree Search / A. Bourki, G. M. J.-B. Chaslot, M. Coulm, V. Danjean, H. Doghmen, J.-B. Hoock, T. Herrault, A. Rimmel, F. Teytaud, O. Teytaud, P. Vayssie`re, and Z. Yu // Proc. Int. Conf. Comput. and Games, LNCS 6515, Kanazawa, Japan. – 2010. – P. 48–58.
11. Xiufeng Yang. Practical Massively Parallel Monte-Carlo Tree Search Applied to Molecular Design / Xiufeng Yang, Tanuj Kr Aasawat, Kazuki Yoshizoe // ICLR 2021 – The Ninth International Conference on Learning Representations, May 3-7. – Access: <https://iclr.cc/virtual/2021/poster/2528>
12. L. Schaefers. Distributed Monte Carlo Tree Search: A Novel Technique and its Application to Computer Go / L. Schaefers, M. Platzner // IEEE Transactions on Computational Intelligence and AI in Games, vol. 7, No. 4. – 2015. – P. 361–374.

References

1. Cameron Browne. A Survey of Monte Carlo Tree Search Methods / Cameron Browne, Edward Powley, Daniel

- Whitehouse, and others // IEEE Trans. on Computational Intelligence and AI in Games. – vol. 4. – no. 1. – March 2012. – P. 1-49.
2. Oleksandr I. Marchenko Classification of Monte-Carlo tree search enhancement techniques oriented to specifics of the method / Oleksandr I. Marchenko, Oleksii O. Marchenko, Mariia M. Orlova // Artificial Intelligence. – 2016. – №2 (72). – P. 59-69.
 3. Oleksandr I. Marchenko. Monte-Carlo Tree Search with Tree Shape Control / Oleksandr I. Marchenko, Oleksii O. Marchenko // 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON). Conference Proceedings. May 29 – June 2, 2017., Kyiv, Ukraine. – 2017. – P. 812-8173.
 4. Oleksii O. Marchenko. Dynamic Parallelizing Model of the Monte-Carlo Tree Search Method for grid-systems / Oleksii O. Marchenko, Oleksandr I. Marchenko // System analysis and information technology: 19-th International conference SAIT 2017, Kyiv, Ukraine, May 22 – 25, 2017. Proceedings. – ESC “IASA” NTUU “Igor Sikorsky Kyiv Polytechnic Institute”, 2017. – P. 213-214.
 5. Oleksii O. Marchenko. Resource model for heterogenous distributed computer system with local connections and its graph. / Oleksii O. Marchenko, Oleksandr I. Marchenko // Computer-Integrated Technologies: Education, Science, Production. – 2020. – № 39. – P. 83-88.
 6. T. Cazenave. On the Parallelization of UCT / T. Cazenave and N. Jouandeau // Proc. Comput. Games Workshop, Amsterdam, Netherlands. – 2007. – P. 93–101.
 7. Y. Soejima. Evaluating Root Parallelization in Go / Y. Soejima, A. Kishimoto, and O. Watanabe // IEEE Trans. Comp. Intell. AI Games. – vol. 2. – no. 4. – 2010. – P. 278–287.
 8. E. Steinmetz. More Trees or Larger Trees: Parallelizing Monte Carlo Tree Search / E. Steinmetz, M. Gini // IEEE Transactions on Games, vol. 13, No. 3. – 2021. – P. 315–320.
 9. G. M. J.-B. Chaslot. Parallel Monte-Carlo Tree Search / G. M. J.-B. Chaslot, M. H. M. Winands, and H.J. van den Herik // Proc. Comput. And Games, LNCS 5131, Beijing, China. – 2008. P.60–71.
 10. A. Bourki. Scalability and Parallelization of Monte-Carlo Tree Search / A. Bourki, G. M. J.-B. Chaslot, M. Coulm, V. Danjean, H. Doghmen, J.-B. Hooek, T. Herrault, A. Rimmel, F. Teytaud, O. Teytaud, P. Vayssi`re, and Z. Yu // Proc. Int. Conf. Comput. and Games, LNCS 6515, Kanazawa, Japan. – 2010. – P. 48–58.
 11. Xiufeng Yang. Practical Massively Parallel Monte-Carlo Tree Search Applied to Molecular Design / Xiufeng Yang, Tanuj Kr Aasawat, Kazuki Yoshizoe // ICLR 2021 – The Ninth International Conference on Learning Representations, May 3-7. – Access: <https://iclr.cc/virtual/2021/poster/2528>
 12. L. Schaefers. Distributed Monte Carlo Tree Search: A Novel Technique and its Application to Computer Go / L. Schaefers, M. Platzner // IEEE Transactions on Computational Intelligence and AI in Games, vol. 7, No. 4. – 2015. – P. 361–374.