

DOI: 10.36910/6775-2524-0560-2020-39-14

УДК: 004.8

Марченко Олександр Іванович, к.т.н., доцент,

<https://orcid.org/0000-0002-4537-3420>

Марченко Олексій Олександрович, асистент

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

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

МОДЕЛЬ РЕСУРСІВ НЕОДНОРІДНОЇ РОЗПОДІЛЕНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ З ЛОКАЛЬНИМИ ЗВ'ЯЗКАМИ ТА ЇЇ ГРАФ

Марченко О.О., Марченко О.І. Модель ресурсів неоднорідної розподіленої комп'ютерної системи з локальними зв'язками та її граф. На основі аналізу методу MCTS та підходів до його паралелізації у статті запропоновані нова модель ресурсів неоднорідної розподіленої комп'ютерної системи з локальними зв'язками (НРКСЛЗ) та її граф. Показано відмінності цієї моделі та її графу порівняно з відомим узагальненим графом системи ресурсів. Передбачається, що запропоновані модель та її граф дозволять розробити нові способи планування ресурсів НРКСЛЗ при реалізації різних видів паралелізації MCTS.

Ключові слова: паралельні та розподілені комп'ютерні системи, модель ресурсів комп'ютерної системи, задачі штучного інтелекту, дерева ігор, пошук в дереві методом Монте-Карло, MCTS, паралелізація MCTS.

Марченко А.А., Марченко А.И. Модель ресурсов неоднородной распределенной компьютерной системы с локальными связями и ее граф. На основе анализа метода MCTS и подходов к его параллелизации в статье предложены новая модель ресурсов неоднородной распределенной компьютерной системы с локальными связями (НРКСЛС) и ее граф. Показано отличия этой модели и ее графа по сравнению с известным обобщенным графом системы ресурсов. Предполагается, что предложенные модель и ее граф позволят разработать новые способы планирования ресурсов НРКСЛС при реализации различных видов параллелизации MCTS.

Ключевые слова: параллельные и распределенные компьютерные системы, модель ресурсов компьютерной системы, задачи искусственного интеллекта, деревья игр, поиск в дереве методом Монте-Карло, MCTS, параллелизация MCTS.

Marchenko O.O., Marchenko O.I. Resource model for heterogenous distributed computer system with local connections and its graph. Basing on the MCTS method analysis and approaches to its parallelization the paper proposes a new resource model of a heterogeneous distributed computer system with local connections (HDCSLC) and its graph. Differences of the model and its graph from the known generalized graph of a resource system are shown. It is supposed that the proposed model and graph would allow to create new resource planning techniques for HDCSLC with the aim of implementation of various kinds of MCTS parallelization.

Keywords: Parallel and distributed computer systems, resource model for computer system, artificial intelligence tasks, game trees, Monte-Carlo tree search method, MCTS, MCTS parallelization.

Рис. 1. Лім.14.

Постановка наукової проблеми.

Прискорення обчислення задач штучного інтелекту, в яких використовуються надзвичайно великі обсяги даних і час рішення яких зараз є занадто великим, було і залишається актуальною проблемою. Апробація нових методів, способів, алгоритмів, структур даних в цій галузі виконується, як правило, на складних інтелектуальних іграх (шахи, Go), а формами подання інформації в задачах штучного інтелекту зазвичай є форми у вигляді дерев послідовних рішень. Одним з методів штучного інтелекту, що був запропонований порівняно нещодавно 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) [1]. Вдосконалення методу MCTS з метою прискорення процесу пошуку кращих рішень може бути виконано як на алгоритмічному рівні, так і на рівні його розпаралелення. В той час, як ефективна паралелізація інших способів пошуку по дереву, що застосовуються в тих же задачах (наприклад, мінімакний пошук), є проблематичною [2], то, на противагу цим способам, MCTS дозволяє зробити розпаралелення достатньо ефективно, оскільки виконує пошук по дереву напіввипадковим чином і здійснює множину моделювань подальшого розвитку подій від заданого стану, які є незалежними. На сьогодні вже були запропоновані декілька методів паралелізації методу MCTS, які дозволяють отримати непогані результати, але це, очевидно, далеко ще не всі можливості розпаралелення MCTS, особливо враховуючи велику різноманітність сучасних багатокomp'ютерних, багато-процесорних та багатоядерних обчислювальних систем.

Особливий інтерес представляє пошук підходів до ефективної паралелізації MCTS для неоднорідних розподілених комп'ютерних систем з локальними зв'язками (НРКСЛЗ), оскільки такі системи можуть бути зібрані на основі комп'ютерного парку навіть невеликих організацій і тому є найбільш доступними для більшості дослідників.

Аналіз досліджень.

Для нашого дослідження важливі два різних напрямки попередніх досліджень. Одним з них є дослідження з підвищення ефективності пошуку MCTS на основі його базової теорії, а другим – підвищення ефективності цього пошуку на основі розпаралелення і використання сучасних комп'ютерних систем.

Найбільш фундаментальною узагальнюючою роботою з методу MCTS першого напрямку є огляд 2012 року, підготовлений десятима провідними вченими під керівництвом Камерона Броуне, що досліджують цей метод [1]. В наступні роки з'явилися не тільки нові модифікації та покращення існуючих способів, але й принципово нові варіанти реалізації MCTS. Ще один погляд на структуру та взаємозв'язки між ідеями існуючих способів покращення MCTS був запропонований авторами у вигляді класифікації [3, 4]. В подальшому в [5, 6, 7, 8] авторами були запропоновані прості та логарифмічні критерії типу «глибина-ширина» DWCs (Depth-Width Criteria) для контролю форми дерева пошуку при використанні методу Монте-Карло, а також спосіб покращення ефективності роботи MCTS, названий пошуком по дереву методом Монте-Карло з контролем форми дерева (Monte-Carlo Tree Search with Tree Shape Control – MCTS-TSC) [6]. Гілмар Фіннссон та Інгві Бьорнссон [9] досліджували ситуацію, чи віддає MCTS перевагу деревам з певною формою, більш глибоким чи більш широким, чи дотримує якогось певного балансу між глибиною і шириною. Крім того, використовуючи критерії типу «глибина-ширина», можна більш ефективно, з точки зору пошуку MCTS, виконувати паралелізацію роботи цього алгоритму. Тому, другим важливим напрямком попередніх досліджень є паралелізація роботи MCTS. Як зазначалося у [3, 4], за критерієм орієнтації способу паралелізації на конкретні апаратні ресурси, чи відсутності такої орієнтації, існуючі способи паралелізації пошуку в дереві методом MCTS можуть бути поділені на апаратно-незалежні способи та апаратно-орієнтовані способи. Для розпаралелення на апаратно-незалежному рівні були запропоновані три загальних підходи: листкова паралелізація, коренева паралелізація і деревна паралелізація [1, 10, 11].

Як відомо, схема роботи методу MCTS складається з наступних чотирьох послідовних етапів [1].

Етап 1: вибір.

Етап 2: розширення.

Етап 3: моделювання.

Етап 4: переобчислення.

Листкова паралелізація реалізується на етапі моделювання (рисунок 1) і, за визначенням даним в [10], представляє собою виконання множини одночасних моделювань від вершини-листа, яка була додана після закінчення процесу пошуку актуальної вершини, використовуючи політику дерева. Така паралелізація може бути ефективно реалізована на графічних процесорах GPU.

Кореневу паралелізацію [10, 11, 12] іноді називають багато-деревним MCTS, оскільки декілька дерев пошуку MCTS будуються одночасно, тобто розпаралелюються на поточному корені дерева. Такий вид паралелізації MCTS є більш високорівневим і доцільним для реалізації на окремих центральних процесорах CPU або комп'ютерах.

При деревній паралелізації виконується декілька одночасних ітерацій пошуку наступного ходу на одному й тому ж дереві [10, 13]. 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 певної частини дерева MCTS кожному потоку, що також більш ефективно можна реалізувати на більш високому рівні комп'ютерів, процесорів CPU та ядер.

Метою даної роботи є побудова моделі ресурсів неоднорідної розподіленої комп'ютерної системи з локальними зв'язками та її графу, яка є розвитком моделі архітектури grid-системи (GAM), запропонованої авторами раніше в рамках моделі динамічної паралелізації [8], з метою подальшого узагальнення процесу розпаралелення пошуку по дереву методом Монте-Карло на основі способу покращення MCTS з контролем форми дерева пошуку, базуючись на критеріях типу «глибина-ширина»

[5, 6, 7], що забезпечить більш ефективний розподіл наявних апаратних ресурсів для розпаралелення пошуку методом MCTS на різних етапах його роботи.

Модель ресурсів неоднорідної розподіленої комп'ютерної системи з локальними зв'язками та її граф

Для однозначності розуміння викладеного далі матеріалу сформулюємо визначення НРКСЛЗ.

Неоднорідна розподілена комп'ютерна система з локальними зв'язками (НРКСЛЗ) – це неоднорідна розподілена комп'ютерна система (НРКС) у вигляді локальної комп'ютерної мережі (LAN), комп'ютери якої об'єднані у LAN стандартними мережевими засобами.

Модель ресурсів (RM – Resource Model) НРКСЛЗ визначимо як

$$RM(t) = (NN, Node[1...NN], Link[1...NN, 1...NN], t),$$

де

NN(Nodes Member) – загальна кількість вузлів мережі НРКСЛЗ;

Node[1...NN] – масив дескрипторів вузлів мережі НРКСЛЗ, в якому кожен вузол цієї мережі визначається як

Node[i] = (Free(t), RAM, NCC, CF, NGPU, GPU[1...NGPU]), де

Free(Flag "free/busy") - прапорець «вільний/зайнятий» в момент часу t;

RAM - розмір оперативної пам'яті комп'ютера i-го вузла;

NCC(Number of CPU Cores) – кількість ядер центрального процесора i-го вузла;

CF (Cores Frequency) – частота ядер центрального процесора i-го вузла;

NGPU(Number of GPU) – кількість GPU плат i-го вузла;

GPU[1...NGPU] – масив дескрипторів i-го вузла, якщо він має GPU тип;

(тобто NGPU>0), де

GPU[ii] = (Free(t), M, NGC, GCF...?)*, де

Free(Flag "free/busy") – прапорець «вільний/зайнятий» в момент часу t;

M(Memory) – розмір пам'яті GPU;

NGC(Number of GPU Cores) – кількість ядер GPU;

GCF(GPU Core Frequency) - частота ядер GPU;

* – при необхідності кількість параметрів GPU у дескрипторі GPU[ii] може бути збільшена.

Link[1...NN, 1...NN] – масив дескрипторів ліній зв'язку між вузлами мережі НРКСЛЗ, в якому кожна лінія зв'язку (тип кабеля, мережна карта, протокол передачі даних) між j-м (j=1...NN) та k-м (k=1...NN) вузлами якої визначається як,

Link[j, k] = (B)**, де

B(Bandwidth) – пропускна здатність лінії зв'язку.

** – в даній моделі обмежимося тільки одним параметром в дескрипторі, але при необхідності кількість параметрів у дескрипторі може бути розширена.

Визначену модель ресурсів (RM) НРКСЛЗ можна також представити у вигляді нижченаведеного графа моделі ресурсів GRM (Graph of Resource Model) (рис.1).

Зазначимо, що при визначенні цього графа оперувати тільки абстрактними ваговими коефіцієнтами, як зроблено для узагальненого графа в [14] буде недостатньо, оскільки в даному випадку дуже важливими є також якісні характеристики ресурсів НРКСЛЗ.

Розглянемо граф GRM детально.

0-й рівень – це рівень головного комп'ютера (Host), на якому починається пошук MCTS, виконується основний його цикл та визначається результат пошуку.

Зауваження: в якості головного комп'ютера використовувати комп'ютер з потужним графічним процесором недоцільно.

1-й рівень – це рівень вузлових комп'ютерів (Nodes) мережі НРКСЛЗ.

2-й рівень – це рівень ресурсів вузлових комп'ютерів (GPUs, Cores) мережі НРКСЛЗ.

3-й рівень – це рівень ресурсів графічних процесорів, які можуть бути встановлені в певних вузлових комп'ютерах НРКСЛЗ.

Відзначимо, що граф моделі ресурсів (GRM) НРКСЛЗ відповідає узагальненому графу системи ресурсів, визначеному в [14]. Граф GRM, так само, як і в [14] визначається четвіркою категорій:

$$GR = (VR, ER, WVR, WER)$$

з наступними відповідностями, які потрібні для більш детального і спеціалізованого погляду на ресурси:

- єдиній множині однотипних вершин VR в графі GRM відповідають вершини чотирьох категорій (Node, Core, GPU, GCore), але в даній відповідності їх також можна розглядати як однотипні, оскільки категорія вершини визначається її дескриптором (вагою), а не власне самою вершиною.

- єдиній множині однотипних ребер ER в графі GRM, в принципі, відповідають ребра чотирьох категорій, але їх так само, як і вершини, можна також розглядати як однотипні, оскільки категорія ребра також фактично визначається його дескриптором (вагою).

- єдиній множині однотипних ваг вершин WVR в графі GRM відповідають ваги чотирьох категорій. Вага вершини категорії Node визначається її дескриптором Node[i]. Вага вершини категорії Core визначається значенням параметра частоти цього ядра CF в дескрипторі Node[i]. Вага вершини категорії GPU визначається її дескриптором GPU[ii]. Вага вершини категорії GCore визначається значенням параметра частоти графічного процесора GCF в дескрипторі GPU[ii].

- єдиній множині однотипних ваг ребер WER в графі GRM в загальному випадку можуть відповідати ваги чотирьох категорій. Вага ребра, що з'єднує дві вершини j та k категорії Node, визначається дескриптором Link[j, k]. Вагу інших ребер в графі GRM можна не враховувати, оскільки призначення завдань на ці ресурси при розпаралеленні MCTS буде залежати тільки від категорій відповідних вершин (Core, GPU, GCore).

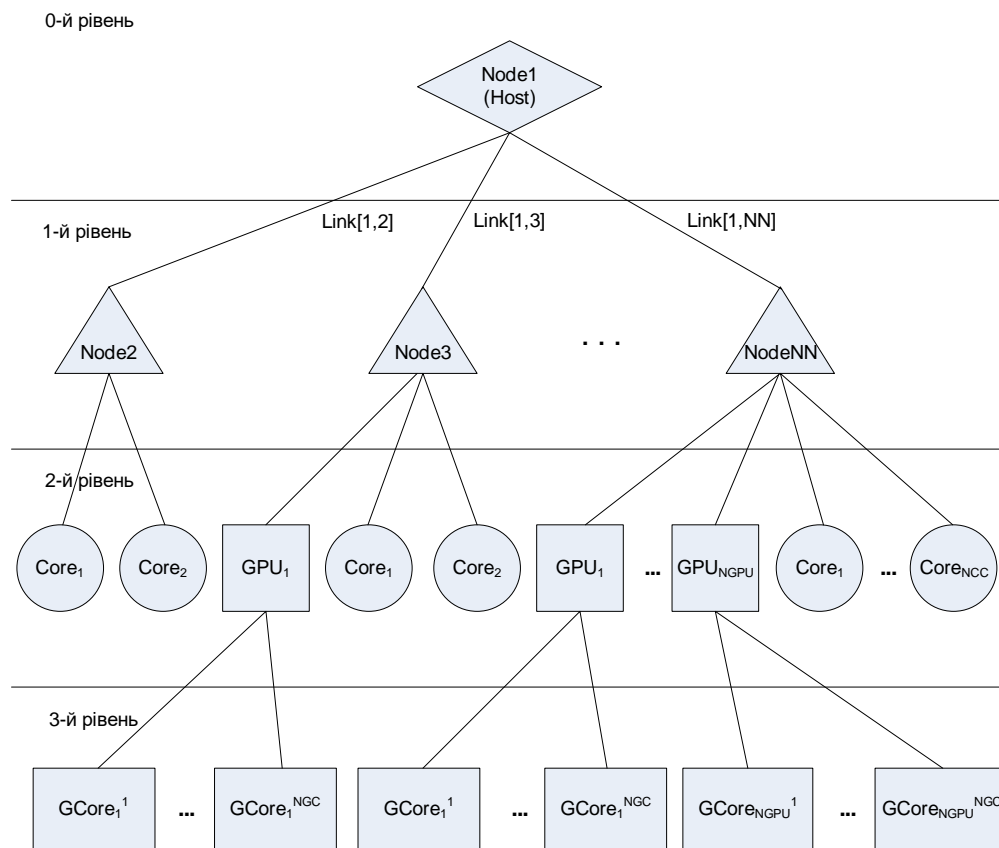


Рис.1. Граф моделі ресурсів GRM

Висновки

Паралелізація пошуку в дереві з використанням методу MCTS може бути виконана згідно декількох різних підходів. На основі аналізу методу MCTS та підходів до його паралелізації у статті запропонована нова модель ресурсів неоднорідної розподіленої комп'ютерної системи з локальними зв'язками

(НРКСЛЗ) та її граф. Показано відмінності цієї моделі та її графу порівняно з узагальненим графом системи ресурсів, визначеному в [14].

Подальшим дослідженням в даному напрямку може бути розробка різних способів планування ресурсів НРКСЛЗ при реалізації різних видів паралелізації МСТS (кореневої, деревної, листкової) на основі запропонованих моделі ресурсів НРКСЛЗ та її графу.

Список бібліографічних посилань

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. Schaeer, J. The APHID Parallel algorithm / Schaeer, J., Brockington M. G // Proceedings of the 8th IEEE Symposium on Parallel and Distributed Processing. – 1996. – P. 428-432.
3. Марченко О. І. Структура та критерії класифікації способів реалізації та покращення пошуку по дереву методом Монте-Карло. / Марченко О.І., Марченко О.О., Орлова М.М. // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2015. – № 21. – С. 51–57.
4. 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.
5. Марченко О.О. Критерій «глибина-ширина» для контролю форми дерева пошуку при використанні методу Монте-Карло. / Марченко О.І., Марченко О.О. // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2016. – № 24-25. – С.42-47.
6. 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.
7. Марченко О.О. Логарифмічний критерій контролю форми дерева для покращення пошуку по дереву методом Монте-Карло. / Марченко О.І., Марченко О.О. // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2017. – № 27. – с.37-43.
8. Марченко О.О. Спосіб динамічного розпаралелення пошуку в дереві методом Монте-Карло в grid-системах. / Марченко О.І., Марченко О.О. // Вимірювальна та обчислювальна техніка в технологічних процессах. – 2017. – №3(59). – с.194-200.
9. Hilmar Finnsson. Game-Tree Properties and MCTS Performance. / Hilmar Finnsson and Yngvi Björnsson // GIGA 2011: Proceedings of the 2nd International General Game Playing Workshop, 2011, pp.23-30.
10. 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.
11. T. Cazenave. On the Parallelization of UCT / T. Cazenave and N. Jouandeaude // Proc. Comput. Games Workshop, Amsterdam, Netherlands. – 2007. – P. 93–101.
12. 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.
13. 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.
14. Симоненко В.П. Организация вычислительных процессов в ЭВМ, комплексах, сетях и системах. // Киев, ТОО «ВЕК+». – 1997. – 302 с.

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. Schaeer, J. The APHID Parallel algorithm / Schaeer, J., Brockington M. G // Proceedings of the 8th IEEE Symposium on Parallel and Distributed Processing. – 1996. – P. 428-432.
3. O.I.Marchenko. Structure and criteria for classification of techniques for implementation and improvement of Monte-Carlo tree search./ O.I.Marchenko, O.O.Marchenko // Komp'uterno-integrovani tekhnologii: osvita, nauka, vyrobnytstvo, No.21, 2015, pp.51-57 (in Ukrainian).
4. O.I.Marchenko. Classification of Monte-Carlo Tree Search Enhancement Techniques Oriented to Specifics of the Method./ O.I.Marchenko, O.O.Marchenko, M.M.Orlova // Artificial Intelligence, No.2(72), 2016, pp.59-69 (in Ukrainian).
5. O.O.Marchenko. "Depth-Width" Criterion for Control of the Search Tree Shape Using Monte-Carlo Method./ O.O.Marchenko, O.I.Marchenko // Komp'uterno-integrovani tekhnologii: osvita, nauka, vyrobnytstvo, No.24-25, 2016, pp.42-47 (in Ukrainian).
6. 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.
7. O.O.Marchenko. Logarithmic criterion for tree shape control for improvement of the Monte-Carlo tree search method./ O.O.Marchenko, O.I.Marchenko // Komp'uterno-integrovani tekhnologii: osvita, nauka, vyrobnytstvo, No.27, 2017, pp.37-43 (in Ukrainian).
8. O.O.Marchenko. Technique for MCTS dynamic parallelization for grid-systems./ O.O.Marchenko, O.I.Marchenko // Vymiriuvalna ta obchysliuvalna tekhnika v tekhnolohichnykh procesakh, No.3(59), 2017, pp.37-43 (in Ukrainian).
9. Hilmar Finnsson. Game-Tree Properties and MCTS Performance. / Hilmar Finnsson and Yngvi Björnsson // GIGA 2011: Proceedings of the 2nd International General Game Playing Workshop, 2011, pp.23-30.

10. 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.
11. T. Cazenave. On the Parallelization of UCT / T. Cazenave and N. Jouandeau // Proc. Comput. Games Workshop, Amsterdam, Netherlands. – 2007. – P. 93–101.
12. 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.
13. 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. Vayssi`re, and Z. Yu // Proc. Int. Conf. Comput. and Games, LNCS 6515, Kanazawa, Japan. – 2010. – P. 48–58.
14. Simonenko V.P. Organization of computing processes in computers, complexes, networks, and systems. // Kiev, TOO "VEK+". – 1997. – 302 p.