



7. Langtangen H. A Primer on Scientific Programming with Python (Texts in Computational Science and Engineering) 3rd ed. Springer, 2012, 798 p.
8. Wielemaker J., Beek W., Hildebrand M., Ossenbruggen J. ClioPatria: A SWI-Prolog infrastructure for the Semantic Web. Semantic Web, 2016, Vol. 7 (5), pp. 529–541.
9. Lager T., Wielemaker J. Pengines: Web Logic Programming Made Easy. Theory and Practice of Logic Programming, 2014, Vol. 14 (4-5).
10. Wielemaker J., Schreiber G., Wielinga B. Prolog-Based Infrastructure for RDF: Scalability and Performance. In: Fensel D., Sycara K., Mylopoulos J. (eds) The Semantic Web - ISWC 2003. ISWC 2003. Lecture Notes in Computer Science. 2003, Vol. 2870. Springer, Berlin, Heidelberg.
11. Nefedova Yu.S. Arkhitektura gibridnoi rekomendatel'noi sistemy GEFEST (Generation–Expansion–Filtering–Sorting–Truncation) [The architecture of the hybrid GEFEST recommendation system (Generation-Expansion-Filtering-Sorting-Truncation)]. Sistemy i sredstva informatiki [Systems and Means of Informatics], 2012, Vol. 22, Issue 2, pp. 176–196.
12. Beel J., Gripp B., Langer S., Breiting C. Research-paper recommender systems: a literature survey. International Journal on Digital Libraries. 2016. Vol. 17. Pr. 305. (access date: 12.12.2016).
13. Kuć R., Rogoziński M. Mastering Elasticsearch - Second Edition. Packt Publishing, 2015, 372 p.
14. Capadislis S., Guy A., Verborgh R., Lange C., Auer S., Berners-Lee T. Decentralised Authoring, Annotations and Notifications for a Read-Write Web with dokieli. Procs of ICWE international conference, 5-8 June, 2017, Rome, Italy. (to appear) [Electronic resource] Preprint. URL:<http://csarven.ca/dokieli-rww>. (access date: 12.12.2016).
15. Heino N., Tramp S., Auer S., et al. Managing Web Content using Linked Data Principles – Combining semantic structure with dynamic content syndication. Computer Software and Applications Conference (COMPSAC), IEEE 35th Annual, 2011, pp. 245–250. [Electronic resource] URL:[http://svn.aksw.org/papers/2011/COMPSAC\\_lod2.eu/public.pdf](http://svn.aksw.org/papers/2011/COMPSAC_lod2.eu/public.pdf) (access date: 30.05.2013).

### Информация об авторах

*Черкашин Евгений Александрович* - к. т. н., старший научный сотрудник, Институт динамики систем и теории управления СО РАН им. В.М. Матросова, г. Иркутск, e-mail: [eugeneai@irmok.net](mailto:eugeneai@irmok.net)

*Орлова Ирина Витальевна* - к. ф.-м. н., доцент, Иркутский национальный исследовательский технический университет, г. Иркутск, e-mail: [soobshenie\\_1@mail.ru](mailto:soobshenie_1@mail.ru)

### Authors

*Evgeny Alexandrovich Cherkashin* – Ph.D. in Engineering Science, Senior Research Officer, Federal Government Budgetary Institution of Science, Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, Irkutsk, e-mail: [eugeneai@irmok.net](mailto:eugeneai@irmok.net)

*Irina Vitalievna Orlova* – Ph.D. in Physics and Mathematics, Assoc. Prof., Irkutsk National Research Technical University, Irkutsk, e-mail: [soobshenie\\_1@mail.ru](mailto:soobshenie_1@mail.ru)

### Для цитирования

Черкашин Е. А. Инструментарий создания цифровых архивов документов на основе связанных данных / Е. А. Черкашин, И. В. Орлова // Современные технологии. Системный анализ. Моделирование. - 2017. -Т. 56, № 4. -С. 100–107. -DOI: 10.26731/1813-9108.2017.4(56).100-107.

### For citation

Cherkashin E. A., Orlova I. V. Instrumentarii sozdaniya tsifrovyykh arkhivov dokumentov na osnove svyazannykh dannykh [Instrumental tools for construction of the digital archives of the documents based on linked data]. *Sovremennye tekhnologii. Sistemyi analiz. Modelirovaniye [Modern Technologies. System Analysis. Modeling]*, 2017. Vol. 56, No. 4, pp. 100–107. DOI: 10.26731/1813-9108.2017.4(56).100-107.

УДК 004.421

DOI: 10.26731/1813-9108.2017.4(56).107-114

**А. Д. Колосов, В. О. Горовой, В. В. Кондратьев**

*Иркутский национальный исследовательский технический университет, г. Иркутск, Российская Федерация*

*Дата поступления: 8 ноября 2017 г.*

## СЕРВИС ДЛЯ РЕШЕНИЯ ЗАДАЧ БУЛЕВОЙ ВЫПОЛНИМОСТИ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ NVIDIA CUDA

**Аннотация.** В статье представлен краткий обзор существующих решателей задач булевой выполнимости, ускоренных с применением технологии NVIDIA CUDA. Рассматриваются основные подходы к распараллеливанию решателей, основанных на методах как полного, так и локального поиска. Представлено описание решателя, разработанного автором статьи, основанного на алгоритме локального поиска. Рассмотрены наиболее заметные реализации решателей, распараллеленных при помощи GPU. Также приведён решатель, основанный на методе локального поиска, разработанный автором. Описан сервис, разработанный с использованием инструментария HrcSoMas Framework. С помощью данного сервиса проведено большое количество экспериментов, результаты которых были обработаны в автоматическом режиме. Сервис показал свою работоспособность и значительно упростил проведение экспериментов. Приведён пошаговый алгоритм, использованный для реализации. Описан способ ускорения алгоритма с применением графического ускорителя. Рассматривается один из вариантов распределения памяти между центральным и графическим процессором. Приведено описание сервиса для автоматизации решения задач булевой выполнимости и обработки результатов. Данный сервис обеспечивает удаленный доступ к вычислительному ресурсу, предоставляет пользовательский интерфейс для формирования задания по запросу пользователя, осуществляет мониторинг выполнения задания и уведомляет пользователя о статусе запроса. Другой важной функцией разработанного сервиса является возможность статистической обработки полученных результатов и визуализации интересующих значений в виде графиков. В заключительной части статьи представлены результаты вычислительных экспериментов, полученные с помощью описанного сервиса. К дальнейшим направлениям исследования можно отнести расширение функциональных воз-



возможностей сервиса, а также разработку дополнительных инструментальных средств для решения дискретных комбинаторных задач на основе SAT-подхода.

**Ключевые слова:** SAT-задача, сервис, CUDA, моделирование, булева выполнимость, автоматизация, автоматизация вычислений.

**A. D. Kolosov, V. O. Gorovoy, V. V. Kondratiev**

Irkutsk National Research Technical University, Irkutsk, the Russian Federation

Received: November 08, 2017

## A SERVICE FOR SOLVING BOOLEAN SATISFIABILITY PROBLEM USING NVIDIA CUDA TECHNOLOGY

**Abstract.** The article presents a review of existing real Mace task feasibility, the result of using the technology NVIDIA CUDA. Examines the main approaches to the parallelization of real rooms the method both full and local search. The description of these developed by the author of the article, the nano at the local search algorithm. The most current implementation, all with the help of the GPU. Also, given real numbers of local search method developed by the author. Open the tools developed using the tool HpcSoMasFramework. Using this service, provide a large number of experiments, the results of which were processed in automatic mode. The service showed its performance significantly and post the experiments. Given the chic algorithm, a plan for implementation. Way to open the scoring algorithm with the use of graphics URL. Describes one of the variants of the memory allocation between the Central and graphics processor. Give a description of the service to automate the solving of problems, the Mace feasibility and treatment results. This service provides access to Alan wise resource provides the interface plates for the formation of the task at the user's request, monitors execution of the job and notifies the user of the status of the request. Another way Funky designed service is the possibility of statistic processing of the obtained results and visualize the values of interest in a graph. In KL-Italy part of the article will be presented the experimental results obtained through the service Pan. The Dalai areas of research include the extension of functionalities of the service, and the development of additional tools for the solution of a discrete combination of tasks based on the SB-approach.

**Keywords:** SAT, service, solver, NVIDIA CUDA, modeling, Boolean feasibility, automation, automation of calculations..

### Введение

Широкий спектр задач, как чисто прикладных, так и теоретических, может быть решен путем применения SAT-подхода, представляющего собой мощный инструмент для их решения [1]. Задача булевой выполнимости (SAT-задача), в терминах которой могут быть сформулированы эти задачи, состоит в ответе на вопрос, выполняли ли данная булева формула, представленная в конъюнктивной нормальной форме (КНФ). Высокая вычислительная сложность SAT-задачи актуализирует разработку программных средств, повышающих эффективность ее решения с помощью высокопроизводительных вычислений.

Для решения SAT-задач активно разрабатываются различные программы-решатели. Часть решателей основывается на алгоритмах с высокой степенью параллелизма. Такие решатели рассчитаны на использование большого количества одинаковых процессорных ядер. В качестве другого популярного направления можно выделить решатели, основанные на гибридной архитектуре [2, 3]. Гибридные решатели объединяют использование стандартных процессорных ядер и различных сопроцессоров. Одним из перспективных вариантов сопроцессора может выступать GPU (Graphic Processor Unit). Наиболее распространенной технологией программирования графических ускорителей является NVIDIA CUDA [4], позволяющая проводить вычисления общего назначения на графических ускорителях. Большинство современных GPU поддерживают NVIDIA CUDA.

Сложность применения гибридного подхода для решения задач булевой выполнимости обу-

славливает актуальность разработки инструментальных средств, автоматизирующих проведение вычислительных экспериментов. В статье представлен гибридный SAT-решатель и сервис для проведения сравнительного анализа работы SAT-решателей на ряде тестовых задач с последующей статистической обработкой и визуализацией результатов. Использование данных средств позволяет проводить вычислительные эксперименты в высокопроизводительной распределенной вычислительной среде, не требуя при этом глубоких знаний в области параллельных архитектур и низкоуровневых средств разработки приложений. Представленные в статье программные средства разработаны в рамках такого подхода с помощью сервис-ориентированного инструментария HpcSoMas Framework [5, 6].

### Обзор существующий GPU-решателей

Графические ускорители занимают крепкие позиции в некоторых областях ресурсоемких вычислений. Не последнюю роль в этом играет активно развивающаяся технология NVIDIA CUDA. CUDA – это архитектура параллельных вычислений от NVIDIA, позволяющая существенно увеличить вычислительную производительность благодаря использованию GPU. На текущий момент большое количество графических ускорителей поддерживают данную технологию. С помощью GPU ускоряется не только обработка изображений или видео, но и приложения для научного моделирования, медицинские приложения и многие другие. Решение SAT-задач не стало исключением.

SAT-решатели можно разделить на 2 больших класса – решатели, основанные на алгоритме



полного поиска, и решатели, основанные на алгоритме локального поиска. При реализации обоих алгоритмов нашли применение GPU-ускорители [7-16].

Большинство современных решателей полного поиска основано на алгоритме DPLL, представляющем собой алгоритм поиска с возвратом. DPLL плохо поддается распараллеливанию из-за своей древовидной структуры. Тем не менее возможно распараллелить процедуру распространения булевых ограничений – BCP (Boolean Constraint Propagation). BCP процедура занимает значительную часть вычислительного времени, и подходит для распараллеливания с использованием GPU [10, 17-20]. Примером полного алгоритма, ускоренного с помощью GPU, является CUD@SAT [11, 21-24]. Данный решатель имеет три режима работы: 0 – все вычисления выполняются на GPU, 1 – BCP-процедура ускоряется с помощью GPU, 2 – на определенном этапе ветви дерева поиска распределяются по GPU-потокам. В работе [12, 25-26] представлена иная схема использования CPU и GPU, в данном решателе основная часть работы выполняется на GPU – на устройстве запускаются различные решатели, от полного перебора до DPLL. CPU в данном решателе используется для вспомогательных операций, таких как анализ конфликта. Для оптимизации работы GPU предусмотрен специальный конвертер, приводящий структуры данных CPU в формат, наиболее эффективный для GPU.

Помимо DPLL, решатели, основанные на полном поиске, используют другие алгоритмы, например, в работе [9] представлен решатель D&C-3SAT, основанный на методе Divide-and-conquer, распараллеливаемый с помощью GPU. В работе [10] предложен вариант GPU-ускорения BCP-процедуры для решателя D&C-3SAT.

К наиболее удачным GPU-решателям, основанным на полном поиске, можно отнести MESP (Minisat Enhanced with Survey Propagation) [7]. В MESP используется эвристика выбора переменной Survey Propagation [8], ускоренная с помощью GPU. Данный подход позволил получить ускорение более чем в два раза по сравнению с minisat.

Наряду с полными методами решения задач булевой выполнимости активно ведутся работы по распараллеливанию методов, основанных на локальном поиске. В качестве примера можно привести работы [4, 13], в которых производится распараллеливание генетических алгоритмов. Также в данной статье описывается локальный алгоритм, разработанный автором. Алгоритм основан на локальном поиске, используемом в известном решате-

теле gsat. Автор предлагает версию данного алгоритма, ускоренную с помощью GPU.

На основании приведенного краткого обзора можно сделать вывод, что графические ускорители нередко применяются для распараллеливания SAT-решателей, однако пока не удаётся получить значительных результатов.

#### **GPU-решатель, основанный на методе локального поиска**

Методы полного поиска достаточно плохо поддаются распараллеливанию, тем более, когда дело касается графических ускорителей. По этой причине автором для реализации был выбран алгоритм локального поиска, используемый в решателе gsat. Данный алгоритм на каждом шаге меняет значение одного из атомов решения на противоположное (flip). Алгоритм заканчивает работу, когда найдено решение, либо, когда достигает ограничения на количество шагов.

В зависимости от параметров запуска, применяются различные эвристики выбора переменной для изменения значения. Полностью случайный выбор переменной – самая быстрая эвристика, но на практике применяется только для уменьшения вероятности попадания в локальный минимум. Более эффективной является эвристика, при использовании которой выбирается случайная переменная, входящая в невыполнимую клаузу. В базовом алгоритме gsat используется выбор переменной, изменение значения которой приводит к минимальному количеству невыполненных клауз. Gsat быстро доходит до минимума, чаще всего локального, для больших размерностей задачи. Для достижения наилучшего результата на практике не принято использовать какую-либо эвристику в чистом виде, используются различные эвристики с разной вероятностью.

#### **CPU+GPU-реализация алгоритма**

В данном алгоритме основные потери процессорного времени приходится на подсчет количества и составление списка невыполненных клауз. Данный этап алгоритма содержит в себе ресурс параллелизма. Для ускорения вычислений была выбрана технология NVIDIA CUDA. CPU+GPU-реализация алгоритма работает следующим образом:

- 1) считывание КНФ в формате DIMACS CNF во внутреннее представление;
- 2) выделение памяти CPU и GPU;
- 3) генерация случайного вектора решения;
- 4) копирование данных из памяти CPU в память GPU;
- 5) вычисление количества и составление списка невыполненных клауз при подстановке вектора решения на GPU;



6) проверка условий выхода на CPU (нет невыполненных клауз, количество итераций достигло ограничения) если одно из условий выполняется, переход на пункт 12;

7) выбор атома для изменения значения на CPU;

8) изменение значения выбранного атома;

9) вычисление количества и составление списка невыполненных клауз при подстановке вектора решения на GPU;

10) сравнение качества полученного решения с лучшим на данный момент;

11) переход к пункту 6;

12) проверка найденного решения;

13) вывод найденного решения и результатов проверки.

#### Использование памяти

При программировании графических ускорителей всегда остро встает вопрос работы с памятью. Не всегда легко определить, какой из видов памяти позволит получить наибольшую эффективность при решении данной задачи. Кроме того, для получения наибольшего ускорения необходимо минимизировать обмен данными между CPU и GPU, а в идеальном случае - полностью избежать промежуточного обмена данными. К сожалению, часто не представляется возможным избежать обмена данными в процессе работы программы. В данной программе, помимо памяти CPU и GPU, используется так называемая «нуль-копируемая память». Она представляет собой память, физически находящуюся в памяти хоста, к которой можно напрямую обращаться с устройства. По результатам тестов, использование нуль-копируемой памяти значительно ускоряет работу программы.

КНФ во внутреннем представлении однократно передается в глобальную память GPU перед началом основного цикла и остается там без изменений до окончания работы программы, используется только для чтения. Количество и список невыполненных клауз хранится в нуль-копируемой памяти. Такое хранение обусловлено тем, что эти данные изменяются при каждой итерации и невыгодно каждый раз инициировать передачу данных с GPU на CPU. Стоит отметить, что запись в нуль-копируемую память происходит в асинхронном режиме и требует обязательной синхронизации перед обращением к данной памяти. Текущий вектор решения хранится в глобальной памяти GPU. Изменение значения выбранного атома происходит с помощью запуска отдельного CUDA-ядра. Вектор текущего решения копируется в память CPU только в случае, если текущее решение оказалось качественнее лучшего.

#### Сервис для автоматизации проведения вычислительных экспериментов

На практике часто приходится проводить множество экспериментов, связанных с выбором наиболее подходящего решателя, подбором параметров решателя, большим количеством задач, требующих решения. В свете этого возникает потребность в автоматизации проведения экспериментов. С использованием инструментария HpcSoMas Framework [5] был сконструирован сервис для решения задач булевой выполнимости. Сервис удобен для настройки параметров и сравнения эффективности SAT-решателей на тестовом наборе КНФ.

Сервис обеспечивает следующие возможности:

- веб-интерфейс для постановки задачи пользователем;
- выбор решателя из списка;
- редактирование списка решателей;
- задание параметров запуска решателя;
- выбор набора КНФ для решения;
- генерация потока заданий, порожденных запросом пользователя;
- мониторинг выполнения потока заданий пользователем;
- информирование пользователя о статусе запроса.
- обработка и визуализация результатов решения.

Данный сервис позволяет проводить большое количество вычислительных экспериментов без постоянного участия пользователя, что значительно экономит рабочее время предметного специалиста. С помощью сервиса удобно анализировать результаты решения, представляя их в виде графиков.

#### Результаты экспериментов

Как было сказано выше, разработанный сервис позволяет автоматически обрабатывать результаты экспериментов и представлять их в виде графиков или таблиц.

Для проведения экспериментов были выбраны два описанных выше решателя, последовательная и GPU-версия базового алгоритма gsat. В качестве платформы для проведения экспериментов был использован один из вычислительных узлов ВК «Академик В. М. Матросов», оборудованный графическим ускорителем (процессор AMD Opteron 6276 2,3 GHz, 2 GB регистровой памяти DDR3-1600 на одно ядро процессора, объем памяти в узле – 64 GB, графический ускоритель Nvidia Tesla C2070 «Fermi»).



Рис. 1. Сравнение времени выполнения одной итерации CPU (крестики) и CPU-GPU(окружности) версий алгоритма. Малые размерности

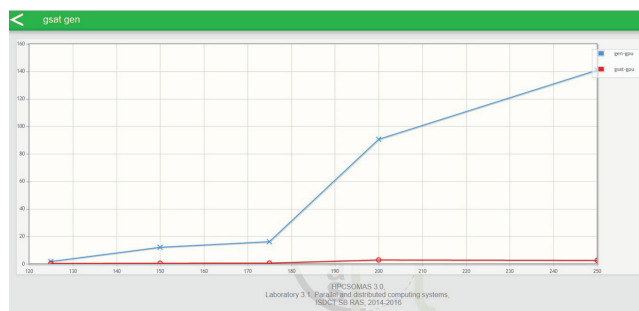


Рис. 3. Сравнение времени решения задачи для GPU алгоритмов gsat (точки) и генетического алгоритма (крестики). Малые размерности

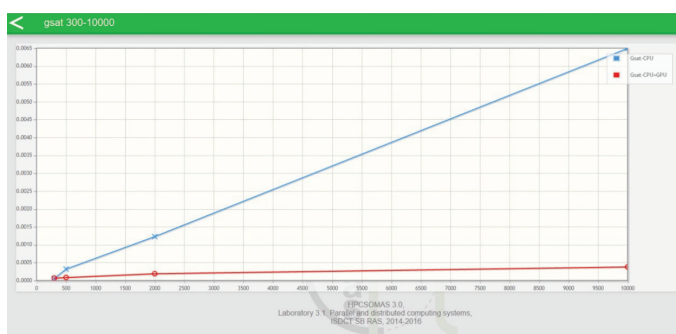


Рис. 2. Сравнение времени выполнения одной итерации CPU(крестики) и CPU-GPU(точки) версий алгоритма. Средние размерности

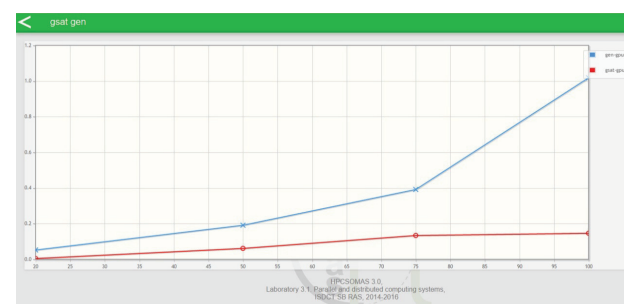


Рис. 4. Сравнение времени решения задачи для GPU алгоритмов gsat (точки) и генетического алгоритма (крестики). Средние размерности

Эксперимент состоит в решении пакета SAT-задач размерности от трех до десяти тысяч переменных. На рис. 1 и 2 представлены графики зависимости времени выполнения одной итерации от размерности задачи для малых и средних размерностей задачи. На горизонтальной оси приведена размерность задачи, на вертикальной – время выполнения одной итерации в секундах.

Как видно по графикам, использование GPU для малых размерностей закономерно приводит к заметному падению производительности. Это вызвано значительным увеличением накладных расходов при использовании графического ускорителя. Для размерности задачи более 300 переменных использование GPU становится оправданным, что также видно на графике, построенном с помощью сервиса.

В качестве другого эксперимента приведем сравнение двух различных решателей, которые используют GPU. В этой части сравнивается решатель, описанный в данной статье ранее, и решатель, основанный на генетическом алгоритме [13]. Эксперимент состоит в решении тестового пакета SAT-задач Uniform Random-3-SAT [16], в качестве параметра для сравнения было выбрано среднее

время решения для каждой из размерностей. В общей сложности было решено более двухсот задач размерности от двадцати до двухсот пятидесяти переменных. С помощью описанного выше сервиса результаты оформлены в виде графика и представлены на рис. 3 и 4. На горизонтальной оси показана размерность задачи, на вертикальной – время, затраченное на поиск решения.

На представленных графиках хорошо видно, что описанный в статье алгоритм показывает значительно лучшую скорость решения задач различных размерностей по сравнению с генетическим алгоритмом [13].

#### Выводы

В работе рассмотрены наиболее заметные реализации решателей, распараллеленных при помощи GPU. Также приведён решатель, основанный на методе локального поиска, разработанный автором. Описан сервис, разработанный с использованием инструментария HrcSoMas Framework. С помощью данного сервиса проведено большое количество экспериментов, результаты которых были обработаны в автоматическом режиме. Сервис показал свою работоспособность и значительно упростил проведение экспериментов. К дальней-



шим направлениям исследования можно отнести расширение функциональных возможностей сервиса, а также разработку дополнительных инструментальных средств для решения дискретных комбинаторных задач на основе SAT-подхода.

### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Gu J., Purdom P.W., Franco J., and Wah B. Algorithms for the Satisfiability problem: a survey // *DIMACS Series on Discrete Mathematics and Theoretical Computer Science, American Mathematical Society*. – 1997. – V. 35. – P.19-151.
2. Опарин Г.А., Богданова В.Г. Инструментальные средства автоматизации параллельного решения булевых уравнений на многоядерных процессорах // Программные продукты и системы. – 2012. – № 1. – С. 10-14.
3. Богданова В.Г., Горский С.А. Технология параллельного решения нелинейных систем булевых уравнений на вычислительном кластере // *Современные технологии. Системный анализ. Моделирование*. – 2013. – № 1 (37). – С. 54-60.
4. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA // М.: ДМКПресс, 2010. — 232 с. ISBN 978-5-94074-578-5.
5. Богданова В.Г., Горский С.А., Пашинин А.А. Сервис-ориентированные инструментальные средства для решения задач булевой выполнимости // *Фундаментальные исследования*. – 2015. – № 2 (часть 6). – С.1151-1156.
6. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Корсуков А.С. Сервис-ориентированный подход к организации распределенных вычислений с помощью инструментального комплекса DISCENT // *Информационные технологии и вычислительные системы*. – 2014. – № 2. – С. 7-15.
7. Gulati K., Khatri S. P. Boolean Satisfiability on a Graphics Processor // *In Proceedings of the 20th Great Lakes Symposium on VLSI (GLSVLSI'10)*. – 2010. – P. 123-126.
8. Braunstein A., Mezard M., Zecchin R. Survey propagation: An algorithm for satisfiability // *Random Structures and Algorithms*. – 2005 (27) . – P.201-226.
9. Meyer Q., Schonfeld F., Stamminger M. et al. 3-SAT on CUDA: Towards a Massively Parallel SAT Solver // *High Performance Computing and Simulation (HPCS)*. – 2010. – P. 306-313.
10. Hironori F., Noriyuki F. GPU Acceleration of BCP Procedure for SAT Algorithms // *The 2012 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'12)*. 16 July 2012. – Las Vegas, 2012. – P. 764-773.
11. Dal Palù, Dovier A., Formisano A., and Pon-telli E. CUD@SAT: SAT Solving on GPUs // *JETA I (Journal of Experimental & Theoretical Artificial Intelligence)*. – 2015. – V.27. – Issue 3. – P. 293-316.
12. Costa C.S. Parallelization of SAT Algorithms on GPUs. Technical report, INESC-ID, Technical University of Lisbon. – 2013.
13. Wang Y. NVIDIA CUDA Architecture-based Parallel Incomplete SAT Solver - 2010 - cs.rit.edu.
14. Kolomycki M. Use NVIDIA CUDA technology to create genetic algorithms with extensive population - 2013 - hgpu.org
15. Z. Luo. G.A., Z. Yang, H. Liu et al. Computation of 3-SAT problem on Graphic Process Unit // *Proceedings of the International Symposium on Intelligent Computation and its Application*. – Wuhan: China University of Geosciences Press, 2005. – P. 27-31.
16. <http://www.cs.ubc.ca/~hoos/SATLIB/benchm.tml>. (дата обращения: 05.08.2016).
17. Елисеев С.В., Банина Н.В., Ахмадеева А.А., Гозбенко В.Е. Математические модели и анализ динамических свойств механических систем // депонированная рукопись № 782-В2009 08.12.2009.
18. Хоменко А.П., Елисеев С.В., Гозбенко В.Е., Банина Н.В. Устройство для управления состоянием объекта защиты // патент на полезную модель RU 56858 21.04.2006.
19. Гозбенко В.Е., Карлина А.И., Каргапольцев С.К. Главные координаты в решении задач вертикальной динамики транспортного средства // *Системы. Методы. Технологии*. 2016. № 3 (31). С. 58-62.
20. Gozbenko V.E., Kargapoltsev S.K., Kornilov D.N., Minaev N.V., Karlina A.I. Definition of the main coordinates of the car with two-level spring suspension // *International Journal of Applied Engineering Research*. 2016. T. 11. № 20. С. 10367-10373.
21. Карлина А.И., Каргапольцев С.К., Гозбенко В.Е. Приведение обобщенных сил в математических моделях транспортных систем // *Современные технологии. Системный анализ. Моделирование*. 2016. № 3 (51). С. 175-179.
22. Гозбенко В.Е., Каргапольцев С.К., Карлина А.И. Приведение динамической системы с тремя степенями свободы к главным координатам // *Современные технологии. Системный анализ. Моделирование*. 2016. № 3 (51). С. 35-38.
23. Карлина А.И., Гозбенко В.Е. Моделирование объектов машиностроения для снижения влияния внешних вибрационных воздействий // *Вестник Иркутского государственного технического университета*. 2016. Т. 20. № 10. С. 35-47.
24. Gozbenko V.E., Kargapoltsev S.K., Minaev N.V., Karlina A.I. Simulation of the vibration of the carriage asymmetric parameters in mathcad // *International Journal of Applied Engineering Research*. 2016. T. 11. № 23. С. 11132-11136.
25. Gozbenko V.E., Kargapolcev S.K., Kondratiev V.V., Minaev N.V., Karlina A.I. Vertical dynamics of the vehicle taking into account roughness gauge // *Fundamental and Applied Studies in the Modern World Papers and Commentaries*. Editor-in-Chief: Prof. Richard Dixon, D. Litt. et Phil. (UK), Publication Director: Prof. Kevin Barrington, D. S. Sc. (UK), Technical Editors: Molly Carpenter, Jeff Goodwin (UK) . 2016. С. 373-383.
26. Гозбенко В.Е., Карлина А.И. Математическая модель вагона с двумя степенями свободы, находящегося под действием периодической вынуждающей силы // *Известия Транссиба*. 2016. № 3 (27). С. 23-31.

### REFERENCES

1. Gu J., Purdom P.W., Franco J., and Wah B. Algorithms for the Satisfiability problem: a survey. *DIMACS Series on Discrete Mathematics and Theoretical Computer Science, American Mathematical Society*, 1997, Vol. 35, pp. 19-151.
2. Опарин Г.А., Богданова В.Г. Instrumental'nye sredstva avtomatizatsii parallelnogo resheniya bulevykh uravnenii na mnogoyadernykh protsessorakh [Automation tools for parallel solution of Boolean equations using multi-core processors]. *Programmnye produkty i sistemy* [Software & Systems], 2012, No. 1, pp. 10-14.



3. Bogdanova V.G., Gorskii S.A. Tekhnologiya paralel'nogo resheniya nelineinykh sistem bulevykh uravnenii na vychislitel'nom klustere [Technology of parallel solution of nonlinear systems of Boolean equations on a computing cluster]. *Sovremennye tekhnologii. Sistemnyi analiz. Modelirovanie* [Modern Technologies. System Analysis. Modeling], 2013, No. 1 (37), pp. 54–60.
4. Borekov A.V., Kharlamov A.A. Osnovy raboty s tekhnologiei CUDA [Fundamentals of using the CUDA technology]. Moscow: DMKPress Publ., 2010, 232 p.
5. Bogdanova V.G., Gorskii S.A., Pashinin A.A. Servis-orientirovannye instrumental'nye sredstva dlya resheniya zadach bulevoi vypolnimosti [Service-oriented tools for solving Boolean satisfiability problems]. *Fundamental'nye issledovaniya* [Fundamental research], 2015, No. 2, Part 6, pp. 1151–1156.
6. Bychkov I.V. et al. Servis-orientirovanniy podkhod k organizatsii raspredelennykh vychislenii s pomoshch'yu instrumental'nogo kompleksa DISCENT [A service-oriented approach to the organization of distributed computing using the DISCENT toolkit]. *Informatsionnye tekhnologii i vychislitel'nye sistemy* [Journal of Information Technologies and Computing Systems], 2014, No. 2, pp. 7–15.
7. Gulati K., Khatri S. P. Boolean Satisfiability on a Graphics Processor. In *Proceedings of the 20th Great Lakes Symposium on VLSI (GLSVLSI'10)*, 2010, pp. 123–126.
8. Braunstein A., Mezard M., Zecchin R. Survey propagation: An algorithm for satisfiability. *Random Structures and Algorithms*, 2005, No. 27, pp. 201–226.
9. Meyer Q., Schonfeld F., Stamminger M. et al. 3-SAT on CUDA: Towards a Massively Parallel SAT Solver. *High Performance Computing and Simulation (HPCS)*, 2010, pp. 306–313.
10. Hironori F., Noriyuki F. GPU Acceleration of BCP Procedure for SAT Algorithms. *The 2012 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'12)*. Las Vegas, 2012, pp. 764–773.
11. Dal Palù, Dovier A., Formisano A., and Pontelli E. CUD@SAT: SAT Solving on GPUs. *JETAI (Journal of Experimental & Theoretical Artificial Intelligence)*, 2015, Vol. 27, Issue 3, pp. 293–316.
12. Costa C.S. Parallelization of SAT Algorithms on GPUs. Technical report, INESC-ID, Technical University of Lisbon, 2013.
13. Wang Y. NVIDIA CUDA Architecture-based Parallel Incomplete SAT Solver 2010. URL: <https://www.cs.rit.edu/>. (access date: 05.08.2016).
14. Kolomycki M. Use NVIDIA CUDA technology to create genetic algorithms with extensive population - 2013 - hgpu.org.
15. Z. Luo. G.A., Z. Yang, H. Liu et al. Computation of 3-SAT problem on Graphic Process Unit. *Proceedings of the International Symposium on Intelligent Computation and its Application. Wuhan: China University of Geosciences Press*, 2005, pp. 27–31.
16. SATLIB - Benchmark Problems. URL: <http://www.cs.ubc.ca/~hoos/SATLIB/benchm.tml>. (access date: 05.08.2016).
17. Елисеев С.В., Банина Н.В., Ахмадеева А.А., Гозбенко В.Е. Математические модели и анализ динамических свойств механических систем // депонированная рукопись № 782-B2009 08.12.2009.
18. Хоменко А.П., Елисеев С.В., Гозбенко В.Е., Банина Н.В. Устройство для управления состоянием объекта защиты // патент на полезную модель RUS 56858 21.04.2006.
19. Гозбенко В.Е., Карлина А.И., Каргапольцев С.К. Главные координаты в решении задач вертикальной динамики транспортного средства // Системы. Методы. Технологии. 2016. № 3 (31). С. 58-62.
20. Gozbenko V.E., Kargapol'tsev S.K., Kornilov D.N., Minaev N.V., Karlina A.I. Definition of the main coordinates of the car with two-level spring suspension // *International Journal of Applied Engineering Research*. 2016. Т. 11. № 20. С. 10367-10373.
21. Карлина А.И., Каргапольцев С.К., Гозбенко В.Е. Приведение обобщенных сил в математических моделях транспортных систем // *Современные технологии. Системный анализ. Моделирование*. 2016. № 3 (51). С. 175-179.
22. Гозбенко В.Е., Каргапольцев С.К., Карлина А.И. Приведение динамической системы с тремя степенями свободы к главным координатам // *Современные технологии. Системный анализ. Моделирование*. 2016. № 3 (51). С. 35-38.
23. Карлина А.И., Гозбенко В.Е. Моделирование объектов машиностроения для снижения влияния внешних вибрационных воздействий // *Вестник Иркутского государственного технического университета*. 2016. Т. 20. № 10. С. 35-47.
24. Gozbenko V.E., Kargapol'tsev S.K., Minaev N.V., Karlina A.I. Simulation of the vibration of the carriage asymmetric parameters in mathcad // *International Journal of Applied Engineering Research*. 2016. Т. 11. № 23. С. 11132-11136.
25. Gozbenko V.E., Kargapol'tsev S.K., Kondratiev V.V., Minaev N.V., Karlina A.I. Vertical dynamics of the vehicle taking into account roughness gauge // *Fundamental and Applied Studies in the Modern World Papers and Commentaries*. Editor-in-Chief: Prof. Richard Dixon, D. Litt. et Phil. (UK), Publication Director: Prof. Kevin Barrington, D. S. Sc. (UK), Technical Editors: Molly Carpenter, Jeff Goodwin (UK). 2016. С. 373-383.
26. Гозбенко В.Е., Карлина А.И. Математическая модель вагона с двумя степенями свободы, находящегося под действием периодической вынуждающей силы // *Известия Транссиба*. 2016. № 3 (27). С. 23-31.

## Информация об авторах

Колосов Александр Дмитриевич - аспирант, институт высоких технологий, инженер НИЧ, Иркутский национальный исследовательский технический университет, г. Иркутск, e-mail: akoloso.irk@gmail.com

Горовой Валерий Олегович - аспирант, институт высоких технологий, инженер НИЧ, Иркутский национальный исследовательский технический университет, г. Иркутск, e-mail: 123valera321@gmail.com

Кондратьев Виктор Викторович - к. т. н., руководитель инновационно-технологического центра, Иркутский национальный исследовательский технический университет, г. Иркутск, e-mail: kvv@istu.edu

## Authors

Alexander Dmitrievich Kolosov – Ph.D. student, Institute of High Technology, Research Department engineer, Irkutsk National Research Technical University, Irkutsk, e-mail: akoloso.irk@gmail.com

Valery Olegovich Gorovo y– Ph.D. student, Institute of High Technology, Research Department engineer, Irkutsk National Research Technical University, Irkutsk, e-mail: 123valera321@gmail.com

Viktor Viktorovich Kondratyev – Ph.D. in Engineering Science, Head of the Innovation and Technology Center, Irkutsk National Research Technical University, Irkutsk, e-mail: kvv@istu.edu



## Для цитирования

## For citation

Колосов А. Д. Сервис для решения задач булевой выполнимости с применением технологии NVIDIA CUDA // Современные технологии. Системный анализ. Моделирование. - 2017. -Т. 56, № 4. -С. 107–114. -DOI: 10.26731/1813-9108.2017.4(56).107-114.

Kolosov A.D. Servis dlya resheniya zadach bulevoi vypolnimosti s primeneniem tekhnologii NVIDIA CUDA [A service for solving boolean satisfiability problem using NVIDIA CUDA technology]. *Sovremennye tekhnologii. Sistemnyi analiz. Modelirovanie* [Modern Technologies. System Analysis. Modeling], 2017. Vol. 56, No.4, pp. 107–114. DOI: 10.26731/1813-9108.2017.4(56).107-114.

УДК 621.865.8; 681.51

DOI: 10.26731/1813-9108.2017.4(56).114-122

**С. П. Круглов, С. В. Ковыришин, И. Е. Ведерников**

*Иркутский государственный университет путей сообщения, г. Иркутск, Российская Федерация*

*Дата поступления: 18 ноября 2017 г.*

### АДАПТИВНОЕ УПРАВЛЕНИЕ ПЕРЕМЕЩЕНИЕМ ГРУЗА МОСТОВЫМ КРАНОМ С ИДЕНТИФИКАЦИОННЫМ АЛГОРИТМОМ

**Аннотация.** Представлен вариант автоматизации технологического процесса на обогатительной угольной фабрике по перемещению груза мостовым краном на заданное расстояние с парированием раскачки в условиях априорной неопределенности о весе груза, длине троса, внешних неконтролируемых возмущениях, связанных с изменяющимся трением тележки и воздействием ветра на груз. В настоящее время существует достаточно большое число методов управления мостовым краном, но до сих пор вопросы автоматизации управления далеки от полного разрешения. В основном это связано с априорной неопределенностью. Предлагается построить адаптивную систему управления с идентификационным алгоритмом и неявной эталонной моделью с использованием «упрощенных» условий адаптируемости, предъявляющих к идентификатору несложные требования по оцениванию неизвестных параметров. При этом описание объекта производится на основе аппроксимационной параметрической модели. Закон управления также строится на основе последовательного приближения с аппроксимирующими свойствами. Это обеспечивает устойчивость замкнутой системы управления. Для текущей параметрической идентификации используется рекуррентный алгоритм наименьших квадратов с фактором забывания. Приводится обоснование условий сходимости невязки идентификации в замкнутом контуре системы управления. Даже при неточных оценках неизвестных параметров объекта управления сходимость невязки идентификации обеспечивает целевое функционирование всей системы управления. Даны рекомендации для построения алгоритмического обеспечения системы управления с учетом реальных факторов: вырожденность движения, точность позиционирования груза при воздействии ветра и другое. Приведен пример, показывающий эффективность предлагаемого подхода.

**Ключевые слова:** мостовой кран, система автоматического управления, гашение колебаний, адаптивное управление, алгоритм текущей идентификации, оценки неизвестных параметров, эталонная модель.

**S. P. Kruglov, S. V. Kovyryshin, I. E. Vedernikov**

*Irkutsk State Transport University, Irkutsk, the Russian Federation*

*Received: November 18, 2017*

### THE ADAPTIVE CONTROL OF MOVEMENT OF CARGO BY THE OVERHEAD CRANE WITH AN IDENTIFICATION ALGORITHM

**Abstract.** The variant of automation of technological process at concentrating coal factory on movement of cargo by the overhead crane at a given distance with parrying of the swing in the conditions of aprioristic uncertainty about the cargo weight, the cable length, external uncontrollable perturbations connected with the changing friction of the cart, and impact of wind on cargo is presented in the article. At present, there is rather large number of methods for controlling the overhead crane, but questions of control automation are still far from full solution. Generally, it is connected with aprioristic uncertainty. It is suggested to construct an adaptive control system with an identification algorithm and implicit reference model with use of the "simplified" conditions of adaptability imposing simple requirements for estimation of unknown parameters to the identifier. At the same time the description of an object is made on the basis of approximating parametrical model. The law of control is also under construction on the basis of the consecutive approach with the approximating properties. Recommendations for creation of algorithmic providing a control system taking into account real factors are made: degeneracy of the movement, accuracy positioning of the cargo at impact of wind, et al. It provides stability of the closed control system. For the current parametrical identification the recurrent algorithm of the smallest squares with a forgetting factor is used. The justification of conditions of the identification error convergence is given in the closed control system contour. Even at inexact estimates of unknown parameters of an object of control, the identification error convergence provides target functioning of all control system. The example showing efficiency of the proposed approach is given in the paper.