

ОПТИМИЗАЦИЯ ПРОЦЕССА ВЫЧИСЛЕНИЯ ХОЛОСТОГО ХОДА РЕЖУЩЕГО ИНСТРУМЕНТА ЗА СЧЕТ РАСПАРАЛЛЕЛИВАНИЯ МЕТАЭВРИСТИЧЕСКОГО АЛГОРИТМА

OPTIMIZATION OF THE PROCESS OF CALCULATING THE CUTTING TOOL IDLING BY METAHEURISTIC ALGORITHM PARALLELIZATION

**A. Subbotin
R. Murzakaev**

Summary. The article considers optimization of idle run of machining tool calculation for cutting problem. Idle run calculation based on the Great Deluge metaheuristic algorithm. Statistic analysis showed modest dispersion of criterion. It means that this algorithm is sustainable. Testing showed decrease of criterion by 1,1%.

Keywords: cutting problem, parallel calculation, threads, the Great Deluge algorithm, calculation of idle run, island models.

Субботин Андрей Михайлович

Пермский национальный исследовательский
политехнический университет
andrey.05.12.93@mail.ru

Мурзакаев Рустам Талгатович

К.т.н., доцент, Пермский национальный
исследовательский политехнический университет
rustmur@gmail.com

Аннотация. В статье рассматривается оптимизация холостого хода режущего инструмента для задачи плоского раскроя материалов. Вычисление холостого хода осуществляется на основе метаэвристического алгоритма Всемирного потопа. Проведенный статистический анализ показал незначительный разброс критерия, что говорит о стабильности алгоритма. Тестирование показало уменьшение критерия в среднем на 1,1%.

Ключевые слова: задача раскроя, параллельные вычисления, потоки, алгоритм Всемирного потопа, вычисление холостого хода, островная модель.

Задача раскроя материала, как подзадача сложной системы автоматизации производства, относится к классу NP-трудных задач комбинаторной оптимизации [1,2,3,4,16]. Это означает, что алгоритма полиномиальной сложности для её оптимального решения до сих пор не найдено, и точный результат в общем случае может быть получен только за экспоненциальное время. [5,6,7] Приближенные алгоритмы решения данной задачи зачастую используют различные методы сокращения перебора [8,9,10]. Поэтому важным этапом при разработке алгоритма раскроя является его быстроедействие, которое позволит перебрать больше вариантов при тех же временных затратах.

Часто для построения холостого хода режущего инструмента используются метаэвристические алгоритмы [11,12,17]. Использование метаэвристик может потребовать больших ресурсов. Оценка особей может длиться достаточно долго: это может быть запуск модели, либо анализ структуры сложного химического соединения. Поэтому параллельные методы являются способом ускорить работу таких алгоритмов [12].

Из методов стохастической оптимизации наиболее подходящими к распараллеливанию являются те, которые используют популяцию решений, поскольку в них изначально одновременно рассматривается множество потенциальных решений, каждое из которых необходи-

мо оценить. Также можно распараллелить методы с одним состоянием (локальный поиск, имитация отжига, Всемирный потоп и др.), но более сложным, неестественным с точки зрения логики построения алгоритма способом.

Целью данной работы является минимизация длины холостого хода режущего инструмента за счёт распараллеливания алгоритма Всемирного потопа (*TheGreatDelugeAlgorithm, GDA*) [14, 15].

В [13,15] описан алгоритм построения холостого хода режущего инструмента программного комплекса *Itas Nesting*. Программе необходимо построить путь перемещения режущего инструмента по листу так, чтобы перемещение инструмента между деталями было минимальным. Режущий инструмент перемещается от одной детали до другой по потенциальным точкам врезки. Точки врезки располагаются в углах деталей. На окружности можно врезаться в любом месте, т.е. на окружности бесконечно много возможных точек врезки. Также учитывается порядок выреза деталей: если есть вложенные детали, то они вырезаются раньше [19].

Представим метод, описанный в [13,15], в виде схемы алгоритма построения холостого хода режущего инструмента (рисунок 1). Его недостатком является то, что он выполняется последовательно.

Рассмотрим возможные методы распараллеливания данного алгоритма.

1. Самым простым способом применения параллельных вычислений к метаэвристическим алгоритмам является использование множественных потоков [12].

Представим схему параллельного алгоритма в виде диаграммы деятельности (*activity diagram*). Данная модель описывает распараллеливание алгоритма при помощи множественных потоков (рисунок 2), в каждом из которых независимо выполняются вычисления холостого хода.

Сначала происходит построение начального решения для метаэвристического алгоритма. Далее при помощи программных средств определяется доступное количество процессоров (k). После чего создаются k потоков. В каждом потоке цикл выполняет n итераций, в которых на каждой детали выбирается случайным образом одна из возможных точек врезки. После этого они записываются в массив *newPath*. Далее происходит сравнение текущего для данного потока решения с новым для определения более подходящего. Программа вычисляет длину холостого хода как сумму расстояний между двумя соседними точками маршрута и сравнивает между собой значения решений в каждом из потоков. Лучшее решение записывается в массив *newPath*. Далее этот массив используется для построения холостого хода режущего инструмента.

2. В рассматриваемой задаче возможно применение островной модели, в которой каждое ядро процессора будет представлять собой «остров» [12] (рисунок 3). Аналогично предыдущему методу, в программе создаётся начальное решение для метаэвристического алгоритма. Далее создаётся k потоков (k равно количеству доступных процессоров), которые будут соответствовать «островам». В каждом потоке цикл выполняет n итераций, в которых строится новое решение. После того, как все потоки отработают, определяется лучшее из полученных решений. Это решение распространяется на все «острова», и алгоритм продолжает расчёт новых решений во всех потоках. Лучшее решение записывается в массив *newPath*. Далее этот массив используется для построения холостого хода режущего инструмента.

Программный модуль параллельной обработки метаэвристического алгоритма был разработан для системы раскроя *Itas Nesting* на языке программирования *Java*. Чтобы узнать, количество доступных процессоров в *Java* есть функция *Runtime.getRuntime().availableProcessors()*. Для реализации параллельных вычислений использовались *thread*'ы.



Рисунок 1. Схема алгоритма построения холостого хода режущего инструмента

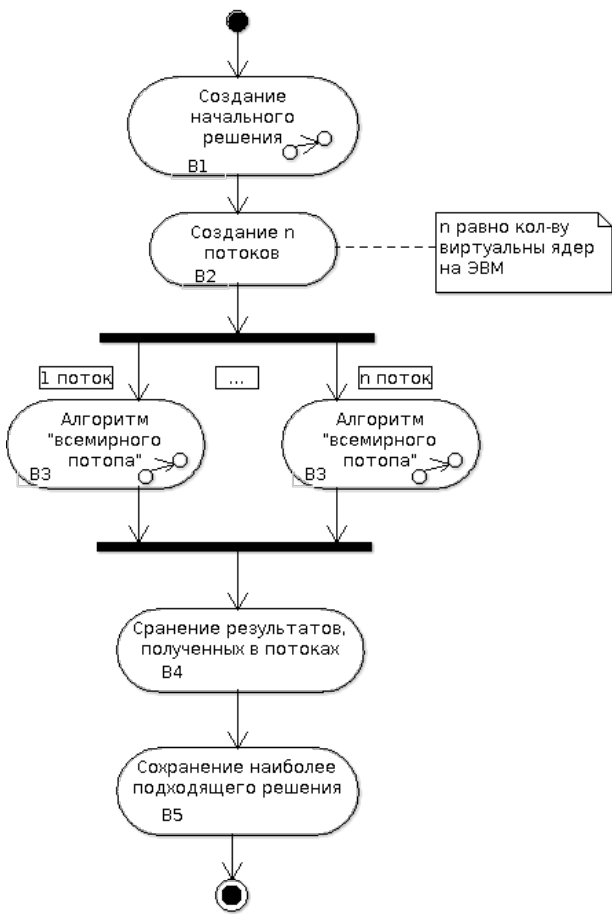


Рисунок 2. Модель параллельного алгоритма построения холостого хода режущего инструмента

Тестирование данного модуля проводилось на компьютере *Lenovo G50*, ОС *Windows8.1*. Количество процессоров в системе равно 4.

Тестирование проводилось для листа размером 1500*6000 мм, содержащем 224 детали различной формы и размера. Основные детали в сборке:

- ◆ небольшие прямоугольники (250*112 мм) с двум зубцами на длинной стороне;
- ◆ детали в форме подковы (габаритом 240*232 мм) с двумя небольшими отверстиями на краях;
- ◆ прямоугольные треугольники со срезанными углами (100*200 мм);
- ◆ детали в форме окружности (диаметр 190 мм) с большим отверстием в центре и шестью небольшими отверстиями по периметру;
- ◆ небольшие детали (\approx габаритом 30*30мм) различной формы.

Тестирование было проведено для трёх случаев: в один поток, в четыре потока (по числу процессоров

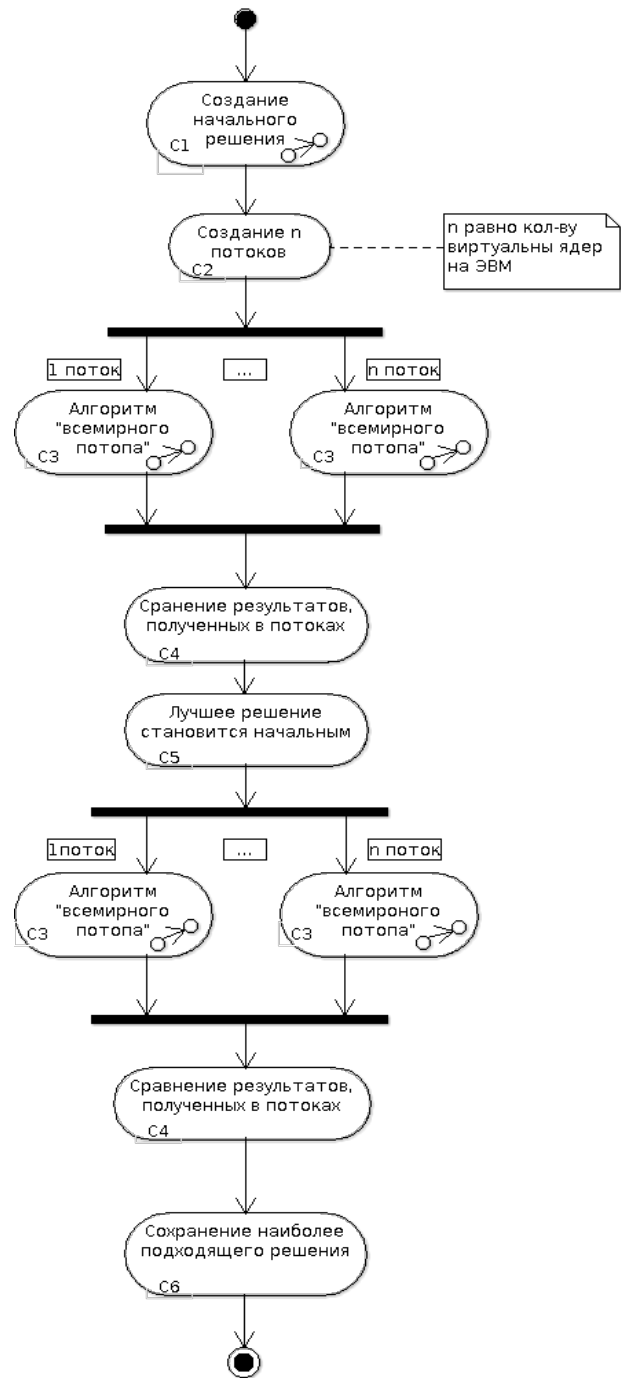


Рисунок 3. Модель параллельного алгоритма построения холостого хода режущего инструмента с использованием островной модели

в системе) для модели множественных потоков и для островной модели. В качестве критерия использовалась длина пути холостого хода режущего инструмента (минимум). Результат тестирования представлен в таблице 1 и на рисунке 4. В каждом тесте алгоритм рассчитывал значение длины холостого хода 100 раз. Так как крите-

Таблица 1. Результаты тестирования

№ теста	Минимальное значение критерия (мм)	Максимальное значение критерия (мм)	Среднее значение критерия (мм)
1	59842,93	59893,78	59864,01
2	59828,42	59867,64	59852,19
3	59779,34	59825,27	59801,88

Тест № 1 — работа алгоритма в один поток.

Тест № 2 — работа алгоритма в 4 потока.

Тест № 3 — работа алгоритма в 4 потока с использованием островной модели.

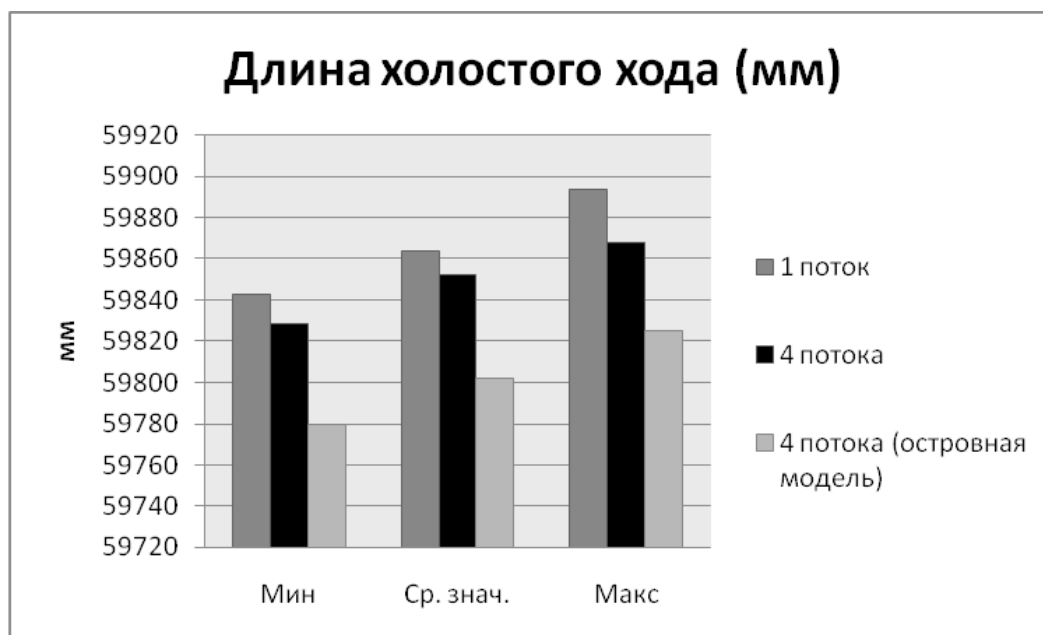


Рисунок 4. Результаты тестирования

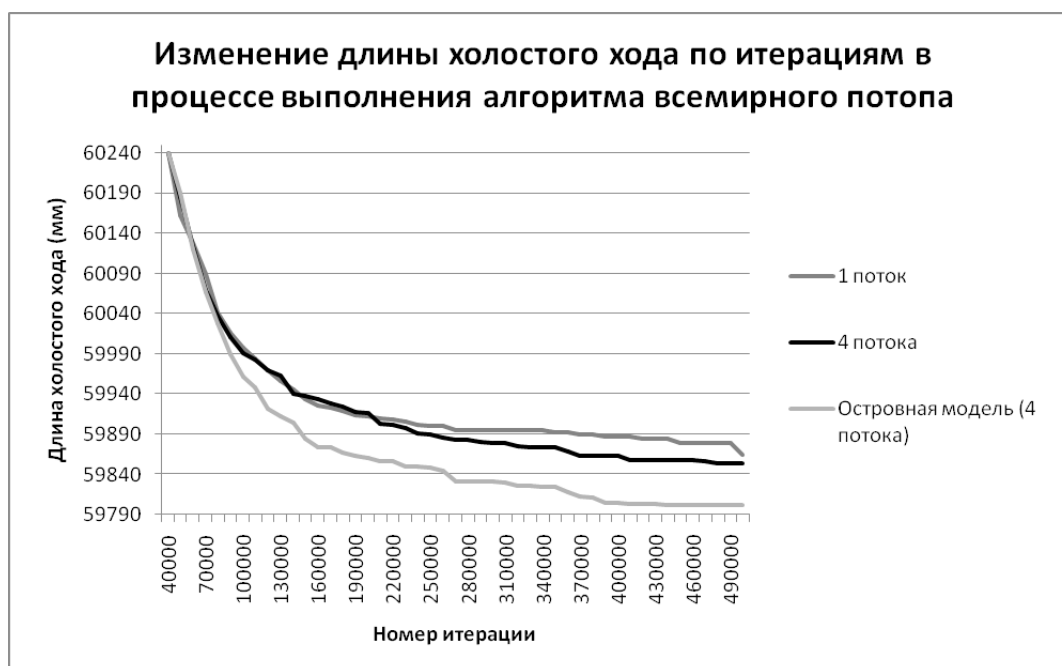


Рисунок 5. Изменение длины холостого хода по итерациям в процессе выполнения алгоритма всемирного потопа

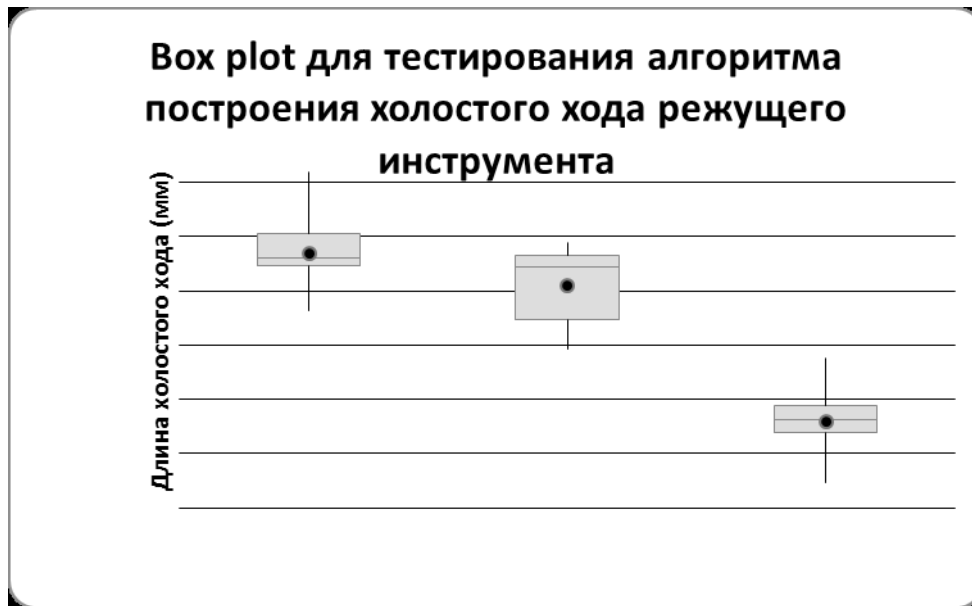


Рисунок 6. Box plot для тестирования алгоритма построения холостого хода режущего инструмента

рий рассчитывается при помощи метаэвристического алгоритма, то в таблице с результатами тестирования отображены только минимальное, максимальное и среднее значения критерия.

Примем среднее значение критерия за исходный сигнал, а значения конкретных тестов за сигнал с шумом. Среднее значение критерия при использовании островной модели равно 59801,88. Это значение меньше, чем минимальное значение критерия при последовательном запуске алгоритма. Это позволяет сделать вывод о том, что разработанный модуль действительно улучшает работу программного комплекса, минимизируя холостой ход режущего инструмента. Если же посчитать изменения в процентном соотношении, то использование островного алгоритма для 4 потоков для данной сборки уменьшает значение холостого хода в среднем на 1,1%.

На рисунке 5 изображено изменение длины холостого хода по итерациям в процессе выполнения алгоритма всемирного потока тремя исследуемыми методами. Как видно на рисунке, уменьшение критерия при использовании островной модели происходит значительно быстрее.

Данная диаграмма позволяет сделать вывод об эффективности использования островной модели для построения холостого хода режущего инструмента, т.к. меньшее значение критерия достигается за меньшее число итераций. То есть для подобных карт раскроя можно будет получить меньшее значение холостого хода. А для карт раскроя с меньшим количеством дета-

лей или с деталями более простой формы (с меньшим числом потенциальных точек врезки) оптимальный вариант построения холостого хода будет достигнут быстрее.

Представим результаты тестирования в виде *box plot* (рисунок 6).

Данная диаграмма в удобной форме показывает медиану, нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы [18]. Расстояния между различными частями ящика позволяют определить степень разброса (дисперсии) и асимметрии данных и выявить выбросы. Точками на графике обозначены средние значения для каждого варианта тестирования алгоритма.

На рисунке 6 видно, что значение медианы и среднее значение для островной модели практически совпадают, а размер самого ящика достаточно мал. Это значит, что использование алгоритма основанного на данной модели в большинстве случаев будет давать результаты (длину холостого хода) примерно равные среднему значению и медиане. А в случае использования алгоритма в 4 потока высока вероятность, что при выполнении программы результаты могут сильно отличаться друг от друга. То есть существует вероятность того, что в данном случае результат может получиться аналогичен результату работы алгоритма в 1 поток; простое распараллеливание может просто замедлить работу программы из-за синхронизации потоков, но не дать какого-либо значимого улучшения работы алгоритма.

Таким образом, по результатам тестирования разработанного алгоритма можно сделать вывод об его эффективности, которая выражена в уменьшении длины холостого хода режущего инструмента. Для карты раскроя, содержащей 224 детали, значение критерия

уменьшается в среднем на 1,1%. Использование островной модели позволяет снизить вероятность «выброса» при работе метаэвристического алгоритма Всемирного потопа, что даёт возможность продолжить работу без его повторного запуска.

ЛИТЕРАТУРА

1. Мурзакаев, Р.Т., Шилов, В.С., Буркова, А. В. Основные методы решения задачи фигурной нерегулярной укладки плоских деталей. // Инженерный вестник Дона. — 2013 — No. 4. — URL: ivdon.ru/magazine/archive/n4y2013/204.
2. Петунин А. А. Методологические и теоретические основы автоматизации проектирования раскроя листовых материалов на машинах с числовым программным управлением: дис.д.т.н.: 05.13.12. Екатеринбург, 2009. — 348 с.
3. Петунин А. А., Мухачева Э. А., Филиппова А. С. Метод прямоугольной аппроксимации для решения задач нерегулярного фигурного раскроя-упаковки. // Информационные технологии. — № 1. — 2008. с. 28–31.
4. Fayzrakhmanov R.A., Murzakaev R. T., Mezentsev A. S., Shilov V. S. Applying the greedy algorithm for reducing the dimensionality of the dynamic programming method in solving the one-dimensional cutting stock problem // Middle-East Journal of Scientific Research. -№ 19 (3). — P. 412–416. — 2014 — URL: [http://www.idosi.org/mejsr/mejsr19\(3\)14/14.pdf](http://www.idosi.org/mejsr/mejsr19(3)14/14.pdf) (дата обращения: 12.03.2014)
5. Валиахметова Ю. И. Мультиметодная технология моделирования ортогональной упаковки и размещения прямоугольно-ориентированных заготовок [Текст]. — автореф. к.т. н.: 05.13.18 / Юлия Ильясовна Валиахметова. — Уфа, 2008. — 19 с.
6. Sykora A. M. Nesting problems: exact and heuristic algorithms. // A Thesis for the degree of Doctor of Philosophy in the University of Valencia, Valencia, 2012–187 p.
7. Fayzrakhmanov R.A., Murzakaev R. T., Mezentsev A. S., Shilov V. S. Application of the Group Decoder for Solving the Orthogonal Materials Cutting Problem // World Applied Sciences Journal 28 (10): 1361–1365. — 2013. — URL: [www.idosi.org/wasj/wasj28\(10\)13/4.pdf](http://www.idosi.org/wasj/wasj28(10)13/4.pdf).
8. Вальковский В. А. Распараллеливание алгоритмов и программ. Структурный подход / Вальковский В. А. — М.: Радио и связь, 1989. 176с.
9. Воеводин В. В. Параллельные вычисления / Воеводин В. В., Воеводин Вл.В. — СПб.: БХВ-Петербург, 2002. 608 с.
10. Quinn M. J. Designing Efficient Algorithms for Parallel Computers. — McGraw-Hill, 1987.
11. Щербина О. А. Метаэвристические алгоритмы для задач комбинаторной оптимизации (обзор). Симферополь: Таврический национальный университет, 2014–17с.
12. Люк Ш. Основы метаэвристик. 2009 [электронный ресурс]. URL: <http://qai.narod.ru/GA/metaheuristics.html> (дата обращения: 09.11.2015).
13. Мурзакаев Р.Т., Шилов В. С., Брюханова А. А. Программный комплекс фигурного раскроя материала Itas Nesting. — Вестник ПНИПУ. Электротехника, информационные технологии, системы управления. № 13, 2015. — 15–25с.
14. Dueck G. New Optimization Heuristics. The great deluge algorithm and the record-to-record travel. Journal of Computational Physics 104, 86–92 (1993).
15. Мурзакаев Р.Т., Приступов В. С. Построение маршрута режущего инструмента на основе алгоритма «всемирного потопа». — В мире научных открытий № 10.2(70), Красноярск: Научно-инновационный центр, 2015. — 828–839 с.
16. Субботин А.М., Шилов В. С. Параллельные вычисления в задачах раскроя материалов. — В мире научных открытий № 10.2(70), Красноярск: Научно-инновационный центр, 2015. — 907–920 с.
17. Валиахметова Ю.И., Филиппова А. С. Теория оптимального использования ресурсов Л. В. Канторовича в задачах раскроя-упаковки: обзор и история развития методов решения. — Вестник УГАТУ Т. 18, № 1(62), 2014. — 186–197 с.
18. McGill Robert, Tukey John W., Larsen Wayne A. Variations of Box Plots. The American Statistician 32 (1): 12–16, 1978.
19. Файзрахманов Р.А., Мурзакаев Р. Т., Бурылов А. В., Приступов В. С. Минимизация общего времени резки с учетом технологий для станков с ЧПУ. — Электротехника № 11, 2016. — 7–12 с.

© Субботин Андрей Михайлович (andrey.05.12.93@mail.ru), Мурзакаев Рустам Талгатович (rustmur@gmail.com).

Журнал «Современная наука: актуальные проблемы теории и практики»