

ПРИБОРОСТРОЕНИЕ, МЕТРОЛОГИЯ И ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ ПРИБОРЫ И СИСТЕМЫ

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ СИСТЕМЫ

DOI 10.21672/2074-1707.2021.55.3.098-109
УДК 004.021, 004.94

ИССЛЕДОВАНИЕ АЛГОРИТМА РАСПАРАЛЛЕЛИВАНИЯ ОПРЕДЕЛЕНИЯ ПУТЕЙ В ЗАДАЧЕ ГРУППОВОГО УПРАВЛЕНИЯ РОБОТАМИ¹

Статья поступила в редакцию 11.05.2021, в окончательной версии – 17.07.2021.

Попов Сергей Геннадьевич, Санкт-Петербургский политехнический университет Петра Великого, 195251, Российская Федерация, г. Санкт-Петербург, ул. Политехническая, 29, доцент, кандидат технических наук, e-mail: popovserge@spbstu.ru
Крашенинников Александр Сергеевич, Санкт-Петербургский политехнический университет Петра Великого, 195251, Российская Федерация, Санкт-Петербург, Политехническая ул., 29, аспирант, e-mail: contr239@mail.ru

Работа посвящена построению и исследованию реализации управляемого менеджера потоков для многопоточных вычислительных систем в контексте транспортной задачи управления гетерогенной группой роботов. Целью работы является исследование параметров менеджера потоков для нахождения субоптимальной конфигурации для решения задачи распределения ресурсов между вычислительными устройствами среды выполнения для формирования практических рекомендаций по настройке разработанного программного обеспечения. В ходе работы проведена серия экспериментов с менеджером потоков, установлены экспериментальные зависимости производительности вычислительной системы от параметров конфигурации менеджера. При проведении экспериментов установлена субоптимальная конфигурация менеджера потоков на основе следующих параметров: количество логических потоков, размер пакета подзадач для вычисления, количество разбиений задачи на подзадачи. Результаты использованы при построении подсистемы управления вычислительными потоками в программном обеспечении менеджера потоков вычисления маршрутов.

Ключевые слова: робот, алгоритм, многопоточность, производительность, оптимизация, гетерогенный, поток, субоптимальный

STUDY THE ALGORITHM OF PATHS CALCULATION PARALLELIZATION IN THE PROBLEM OF GROUP CONTROL OF ROBOTS

The article has been received by editorial board 11.05.2021, in the final version – 17.07.2021.

Popov Sergey G., Peter the Great St. Petersburg Polytechnic University, 29 Polytekhnicheskaya St., St. Petersburg, 195251, Russian Federation, Associate Professor, Cand. Sci. (Engineering), e-mail: popovserge@spbstu.ru
Krashennnikov Alexander S., Peter the Great St. Petersburg Polytechnic University, 29 Polytekhnicheskaya St., St. Petersburg, 195251, Russian Federation, post-graduate student, e-mail: contr239@mail.ru


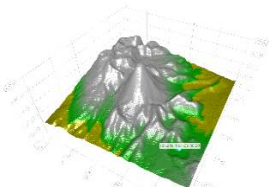

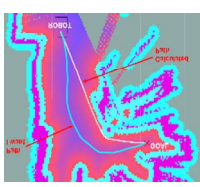
The work is about development and study of the implementation of a multithreaded flow manager in computing systems for the transport problem of controlling a heterogeneous group of robots. The aim of the work is to study the parameters of the thread manager in order to find a suboptimal configuration for solving the task of managing computation resources and formulating practical recommendations for configuration the developed software as well. In the work, a series of experiments were carried out on the thread manager, experimental dependences of the computing system performance on the manager's configuration parameters were established. In the course of the experiments was established that it is possible to find the suboptimal configuration of the thread manager on the basis of the parameters, such as: the number of logical threads, the size of the batch of subtasks for computation, the number of the subtasks. The results will be used in when the creating of a subsystem for managing computational threads of the general problem, and can be used in other applied problems together with a portable software in the software of the routes computation threads manager.

Keywords: robot, algorithm, multithreading, performance, optimization, heterogeneous, thread, suboptimal

¹ Работа выполнена при финансовой поддержке РФФИ, грант № 18-29-03250 мк (The reported study was funded by RFBR according to the research project № 18-2903250 mk).

Графическая аннотация (Graphical annotation)

		Распараллеливание вычислений (Parallel computing)	Вычисленные пути (Calculated routes)
Группа роботов (Robots group)	Построение путей (Path finding)		
План построения путей (Route calculated planning)	Оптимизация вычислений (Calculation optimization)		Вычисленные пути (Route)

Введение. Многопоточный метод позволяет максимально использовать имеющиеся вычислительные ресурсы, так как специальным образом настроенное ПО позволяет оптимизировать и диспетчеризовать многопоточные вычисления. По результатам исследования [2] сформулирована постановка задачи о поиске субоптимальной конфигурации настроек менеджера потоков, так как влияние ключевых параметров, таких как количество задач в пакете или количество логических потоков, оказывает значительное влияние на время вычисления путей. Ручной подбор параметров трудоемок и не гарантирует выбор наилучшей конфигурации. В целом представленные исследования являются расширением работ по части автоматического управления менеджером потоков. Это приводит к тому, что ручная настройка параметров инструмента весьма вероятно приведет к неэффективному использованию, а в некоторых случаях может замедлить расчет до уровня однопоточного выполнения. Поэтому автоматическое вычисление субоптимальной конфигурации настроек менеджера памяти в контексте прикладной задачи [1] на заранее определенной вычислительной системе также является важной научно-технической задачей, а работа в целом является расширением работ по автоматическому управлению менеджером потоков.

Связанные работы. Многопоточность в качестве фактора быстродействия оценивается как фиксированный фактор [3]. В рассматриваемых задачах сортировки такой подход уместен, так как разбиение по потокам происходит по сути в сплошной среде задач. Весь набор подзадач получается однотипным и оптимальным вариантом является простой одновременный запуск количества задач по количеству потоков. Исследование различных многопоточных модификаций алгоритма A* [4] привели к заключению о высокой эффективности вычисления каждого отдельного пути в отдельном потоке и низкой эффективности распараллеливания многопоточного расчета параметров внутри одного маршрута. Дальнейшие предложенные модификации A* связаны с исключительно адаптацией алгоритма для задачи и одновременного расчета нескольких путей. Идея комбинации многопоточного подхода с оптимизацией фрагментации путей в статье не рассмотрена, однако имеет шансы на улучшение времени выполнения расчетов.

Интересные варианты применения многопоточного подхода связаны с генетическим алгоритмом [5, 13]. Этот алгоритм не параллелизуется, так как имеет итеративную основу и четкую последовательность действий. Однако нюансы реализации позволяют выделить в отдельные потоки запись популяции в БД и графическое построение текущей итерации с целью наглядности. По сути время расчета на программном комплексе сведено ко времени расчета базовой частью алгоритма.

Непосредственное взаимодействие на низком уровне с вычислительной средой рассматривается как отдельная задача [6]. Один из важных выводов: отсутствие средств исследования динамических характеристик создаваемых пользователями алгоритмов.

На скорость вычисления может также влиять непосредственная реализация алгоритма [7]. По сути, используемые многими готовые решения в виде контейнеров и встроенных алгоритмов могут содержать избыточную информацию для контекста задачи и вызывать дополнительные расходы ресурсов вычислительной системы.

Возвращаясь к математическим аспектам, следует уделить внимание критериям оптимизации [8]. В исследовании, которому посвящена статья используется динамическая модель. Однако подбор критериев позволяет утверждать о возможности развития текущей статической модели до динамической с более обширным подбором критериев.

Рассмотрение задачи с точки зрения теории управления также позволяет понять многие аспекты целевой среды [9]. Разбиение задачи на синтез и обеспечение устойчивости относительно точки пространства состояний с использованием эволюционных методов позволяет получить высокоуровневые функции управления, что в широком смысле соответствует определению стратегий из первой части исследования [1]. Подобный подход показывает отличный результат на синтетических тестах, однако требует обширного рассмотрения на реальных данных для решения об использовании его в целевой задаче.

Идея исследования местности с воздуха известная, и различные варианты действий предлагались и ранее [10]. В целом, гетерогенная группа роботов позволяет разделить задачи и на начальном этапе создать подобие пайплайна, чтобы затем развивать модель в сторону динамической. Однако в постановке задачи обозначен только один дрон, что не позволяет использовать параллельные вычисления для группы.

Разделение задач на группы рассматривается с разных ракурсов. Интересный вариант предложен с использованием облачных вычислений [11]. Идея заключается в декомпозиции общей задачи использованием специальных способов программирования алгоритмов при решении. Линейное распределение, роевое решение и синтез решений представляют достаточно возможностей для подобного подхода. По результатам статьи успешно апробированы разработанные алгоритмы. Однако вызывает вопросы использование виртуальных машин в облачном сервисе. Требуется уточнение характеристик системы в части производительности.

Также по технической части продолжают исследования в области использования ГП в прикладных задачах [12]. Известно, что производительность GPU в некоторых задачах значительно превышает производительность CPU. Алгоритм нахождения пути несколько отличается от используемого в настоящем исследовании, однако реализованный параллельный алгоритм на программно-аппаратной архитектуре CUDA показывает значительное улучшение производительности.

Технологический аспект в некотором смысле подвергался более углубленному анализу [14]. В статье рассматривалась непосредственно физическая часть многоядерного процессора. По части архитектуры изыскания проводились примерно с момента, как стало понятно, что при современной технологии создания чипов очень скоро будет достигнут предел по рабочей частоте, поэтому необходимо масштабировать другую характеристику – количество агентов.

Наиболее близкой к глобальной идее менеджера потоков можно считать DDMT (Data Driven MultiThreading) [15]. Идея подхода заключается в адаптации внутренних алгоритмов работы ядра под приходящие на расчет данные. Однако целью работы являлось улучшение производительности оптимизацией нагрузки на кешах различных уровней.

Стратегия оптимизации, непосредственно связанная с многопоточностью, также стала частью созданной SensorOS [16, 17]. Специальная операционная система создана для многоагентных задач, где критичен аспект времени. Идея заключалась в согласованном функционировании множества агентов с SensorOS на борту.

Одним из следующих шагов настоящего исследования станет переход к реальным данным. В подобном контексте потребуется предобработка для получения данных для алгоритма поиска пути. Предложенный в статье динамический вариант с использованием триангуляции Делоне и модифицированным алгоритмом A^* [18] может значительно упростить задачу перехода от синтетических данных к реальным.

В продолжение темы об «уходе» от теоретических моделей следует вспомнить аспект многоагентной [19] системы. Вопрос эмуляции централизованного контроля, на первый взгляд, несет неочевидные преимущества. Но если рассмотреть общую схему функционирования системы из настоящего исследования, то возникает вопрос об аспекте принятия решений. В условиях и решении явно не указано отсутствие центрального узла, так как этот вариант постановки планируется рассматривать в дальнейшем. Тем не менее, материал статьи позволяет заранее получить большее понимание возможностей реализации, и, на этом основании скорректировать алгоритм поиска путей.

Немаловажным аспектом при переходе к динамической системе является своевременное обнаружение препятствий на пути наземного робота. На данном этапе эта задача «переложена» на дроны, однако существуют задачи, где они неприменимы. Исследование местности внутри заброшенного завода, например, не позволяет эффективно использовать воздушные единицы без специального дорогостоящего оборудования из-за препятствий в том числе и сверху. Предложенная в статье модель визуального обнаружения препятствий [20] позволяет с использованием обычной видеокамеры с некоторой точностью оцифровывать препятствия и добавлять их в алгоритм поиска пути, как обработанные данные. Эффективность функционирования в сложных условиях при этом «остается открытым» и требует натурных испытаний.

Возвращаясь к вопросам, собственно, алгоритма, необходимо рассмотреть вариант кооперативного алгоритма A^* -ICA*. Идея заключается в наличии штрафных функций за углы поворота и пересечения путей других агентов. Алгоритм показал хорошую производительность. Однако в контексте настоящего исследования требуется учитывать решение задачи построения покрытия, что создает дополнительные условия и может негативно влиять на штрафные функции. Возможным решением является попеременное использование алгоритмов A^* и ICA*. При этом A^* будет использоваться на траектории построения покрытия, когда активен радар, а ICA* при перемещении между этими траекториями.

Таким образом, в контексте текущего исследования построение алгоритмов и разработке программного модуля, связывающего алгоритмическую часть решения задачи с операционной системой и многопоточными вычислениями, является востребованным и важным с точки зрения оптимизации использования ресурсов вычислительной системы.

Постановка задачи предметной области. Дана карта M и гетерогенная группа роботов R . Для карты M могут быть сделаны экспликация $M_g = \{G, Y, R\}$ для наземного робота, где G – зоны свободные для передвижения, Y – опасные зоны, к которым относятся большие углы продольного/поперечного наклона, R – закрытые для перемещения зоны, как то, вода, лава или непроходимые заросли, и экспликация $M_f = \{H\}$ для летающих роботов, где H – карта минимальных высот полета. Реализация карты сохраняется в масштабе достаточном для составления точного пути наземным роботом, при этом вначале выполнения задачи карта не известна.

Группа роботов $R = \{F, G\}$ состоит из: $F = \{f_1, \dots, f_n\}$ – подгруппа из n летающих роботов, а $G = \{g_1\}$ – подгруппа из одного наземного робота. Робот $f = \{C, v, r, h_{max}\}$ имеет радиус обзора карты r , среднюю скорость полета v , затраты энергии в единицу времени C и максимальную высоту полета h_{max} , обусловленную внешними факторами: погода или мощность радиопередатчиков. Летающий робот составляет актуальную карту крупного масштаба. Робот $g = \{C, r, E, \alpha, \beta\}$ радиус обзора карты r , затраты энергии на единицу пройденного расстояния, зависящие от угла въезда δ $C = C(\delta)$, максимальный запас энергии E , максимальный угол поперечного крена – угол сваливания α , максимальный угол въезда β .

В рамках задачи требуется построить такой маршрут от точки A до точки B , чтобы минимизировать суммарные затраты энергии группы и не превысить углы сваливания и въезда для наземного робота.

В этом случае общая постановка сводится к задаче двухкритериальной оптимизации:

$$\min_{\underline{x}} \{f_1(\underline{x}), f_2(\underline{x})\},$$

$$\underline{x} \in S,$$

где $f_i: R^n \rightarrow R$ – это целевые функции, отвечающие за расход энергии в полёте и допустимости движения наземного робота по углам. Векторы решений $\underline{x} = (x_1, x_2, \dots, x_n)^T$ принадлежат непустой области определения S .

В случаях многокритериальной оптимизации критерии удовлетворенности для различных целевых функций, как правило, конфликтуют, потому задача оптимизации часто имеет дополнительные условия в виде приоритетов критериев.

Постановка задачи оптимизации. В ходе решения общей задачи возникает подзадача многократного вычисления путей для каждого робота группы. Зная промежуточные точки, путь можно поделить на фрагменты, которые могут быть выполнены параллельно. При централизованном расчете путей в системе реального времени возникает вопрос о времени расчета участка или всего пути целиком для каждого конкретного робота. Для первой итерации в наших интересах как можно раньше запустить конкретный дрон. На местности могут быть зоны, недоступные для полетов, поэтому считается, что дрон начинает движение, только когда путь полностью рассчитан. В этом случае можем сформулировать критерий субоптимальности выбранной стратегии проведения расчета. Для n роботов имеем задачу оптимизации:

$$\min_{\text{strat}} \sum_{i=1}^n T_{s_i},$$

$$\underline{x} \in S,$$

где T_{s_i} – время старта(S) или время окончания расчета полного пути i -го робота; strat – выбранная стратегия расчета пути.

С учетом сказанного на предыдущих этапах исследования стратегия характеризуется следующими пунктами:

- количество логических потоков в целевой системе расчета (h);

- размер пакета задач для расчета через менеджер потоков (p);
- количество разбиений пути на подпути (s).

Метод распределения задач по потокам. Кроме перечисленного выше, имеется один важный аспект – порядок попадания задач для расчета в менеджер потоков (o). Предположительно, влияние этого параметра будет наименьшим на заданном этапе с учетом невозможности запуска робота без полного расчета. Рассмотрение аспекта позволит сравнить результат в дальнейшем, когда возможность начать движение по неполному пути станет доступна.

В этом случае можно уточнить формулу:

$$\min_{\{h,p,s\}} \sum_{i=1}^n T_{s_i}$$

Такое распределение позволит учесть большее количество параметров математической модели и параметров вычислительной системы.

Исследование влияния стратегии расчета на готовность пути. Основной идеей эксперимента является обнаружение зависимостей времени готовности путей от числа разбиений пути на подпути, количества логических потоков, количества задач в пакете. Вначале имеется предположение о том, что наибольшее влияние на сумму из формулы окажет количество логических потоков, как наиболее явный ограничитель. В предыдущей части [2] проверено влияние превышения количества логических потоков над количеством потоков доступных ОС, и результат показал нецелесообразность подобного превышения, так как не давал роста производительности. Однако этот вариант также необходимо рассмотреть для наглядности.

Первый этап эксперимента предполагал вычисление на стандартном наборе данных (табл. 1) с варьируемым количеством логических потоков. Для «чистоты эксперимента» при расчете количество наземных роботов взято равным нулю. С учетом принципа работы менеджера потоков, количество задач в пакете должно быть большим либо равным количеству потоков. В противном случае часть логических потоков не будет задействована и общая производительностькратно упадет.

Таблица 1 – Параметры эксперимента 1

Параметр или константа	Значение
Размер карты, точек*точек	258x258
Число летающих роботов, шт.	64
Число наземных роботов, шт.	0
Длина пути, точек	100
Количество логических потоков, шт.	8:2:20
Количество задач в пакете, шт.	20
Количество физических потоков, шт.	8(16 w/ Hyper-Threading)

Согласно полученным результатам, наименьшее время выполнения получено при наименьшем количестве потоков, что обуславливается совпадением заданного числа с количеством логических ядер процессора. При этом, если увеличивать количество логических потоков, то общая загрузка ЦП не изменяется, но при этом добавляются временные затраты на системные прерывания и функционирование дополнительных логических потоков менеджера. Зависимости числа и времени вычисления пакетов от числа потоков приведены на рисунках 1 и 2, соответственно.

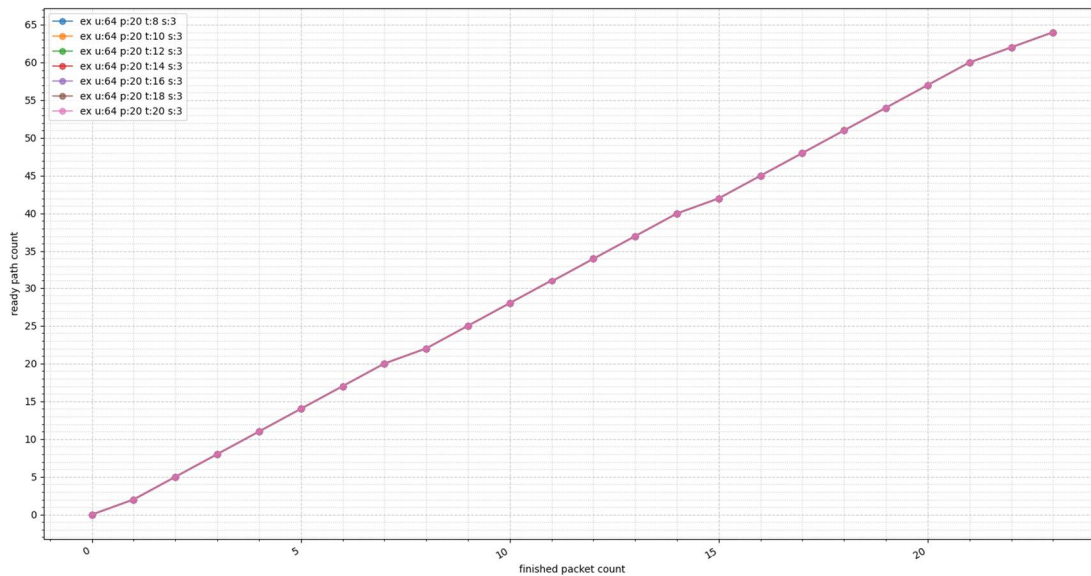


Рисунок 1 – Количество вычисленных пакетов для переменного количества потоков

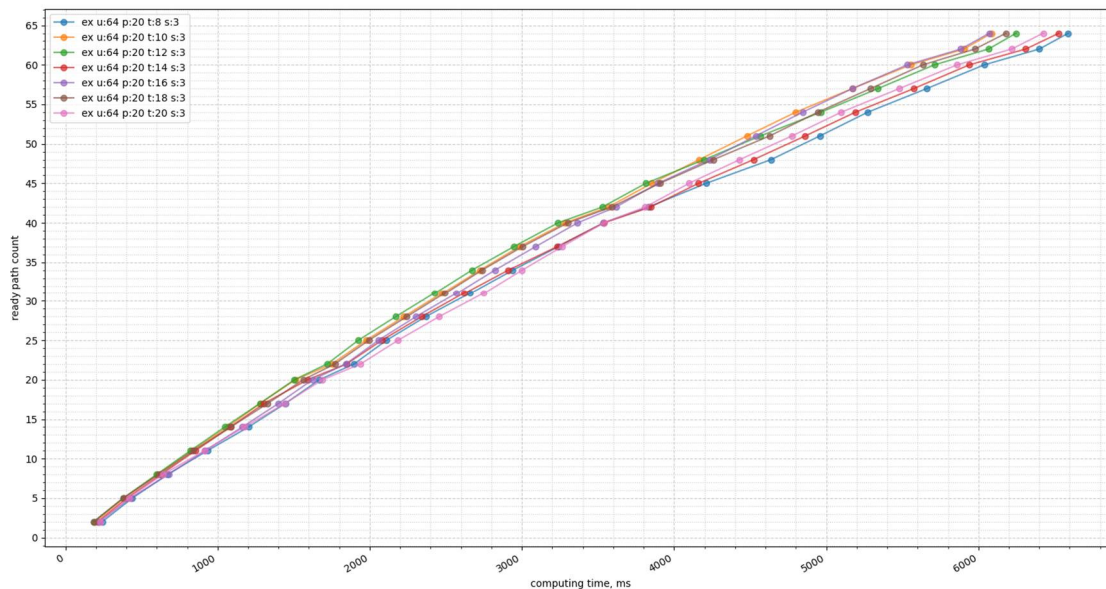


Рисунок 2 – Время расчета для переменного количества потоков

Второй этап заключается в рассмотрении времени вычисления для ограниченного количества логических потоков, которое меньше, чем количество физических потоков доступных ОС (табл. 2). Основной смысл заключается в проверке гипотезы о субоптимальном количестве логических потоков, равном количеству «честных» или физических ядер ЦП. Возможно, технология Hyper-Threading оказывает негативное влияние на однотипные математически операции.

Таблица 2 – Параметры эксперимента 2

Параметр или константа	Значение
Размер карты, точек*точек	258x258
Число летающих роботов, шт.	64
Число наземных роботов, шт.	0
Длина пути, точек	100
Количество логических потоков, шт.	2:2:8
Количество задач в пакете, шт.	8
Количество физических потоков, шт.	8(16 w/ Hyper-Threading)

Согласно полученным результатам, наихудшее время вычисления получается, когда потоков слишком мало. Однако нас в большей степени интересует другие два графика, которые показывают, что при использовании количества логических потоков, равным количеству логических ядер получаем преимущество по времени вычисления. При этом в отличие от экспериментов в предыдущей части исследования [2] выигрыш по времени составляет значительно меньше, чем два раза, так как для данного эксперимента считаем время готовности к движению, а не время математического расчета. Зависимости числа и времени вычисления пакетов для переменного числа потоков приведены на рисунках 3 и 4, соответственно.

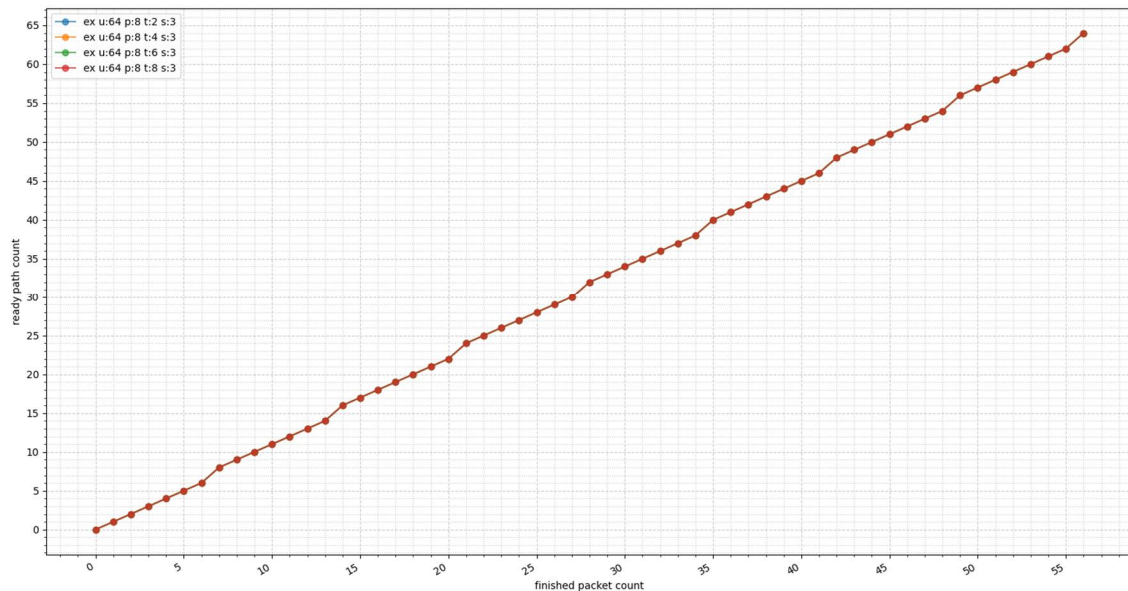


Рисунок 3 – Количество вычисленных пакетов для переменного количества потоков

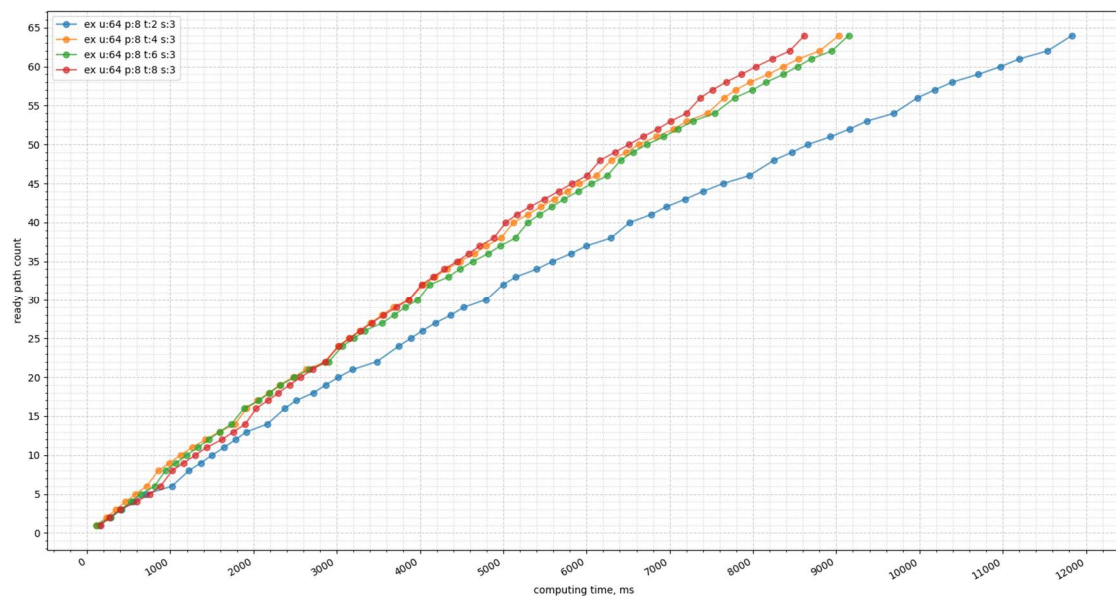


Рисунок 4 – Время расчета для переменного количества потоков

Зафиксируем субоптимальное значение количества логических потоков и рассмотрим варьируемое количество задач в пакете (табл. 3). Диапазон значений на данном этапе содержит как меньшее, так и большее количество задач относительно количества потоков.

Таблица 3 – Параметры эксперимента 3

Параметр или константа	Значение
Размер карты, точек*точек	258x258
Число летающих роботов, шт.	64
Число наземных роботов, шт.	0
Длина пути, точек	100
Количество логических потоков, шт.	8
Количество задач в пакете, шт.	2:2:16
Количество физических потоков, шт.	8(16 w/ Hyper-Threading)

В отличие от предыдущих этапов заметно различие в количестве рассчитанных пакетов на один готовый путь. Наиболее быстрый вариант расчета предполагает использование пакета максимального объема. Вначале имелось предположение, что в некоторых случаях возможна перегрузка менеджера потоков из-за большого количества задач в пакете. Однако на практике такой случай, если и возможен, то не в нашей области применения. Поэтому исследовать этот момент лучше в контексте нескольких задач различных предметных областей. Зависимости числа и времени вычисления пакетов для переменного числа задач приведены на рисунках 5 и 6 соответственно.

Рассмотрим далее влияние количества разбиений на полное время готовности пути (табл. 4). В текущем контексте разбиение кратно увеличивает количество подзадач поиска пути, но сокращает ожидаемую длину пути, а значит и время расчета каждого конкретного участка.

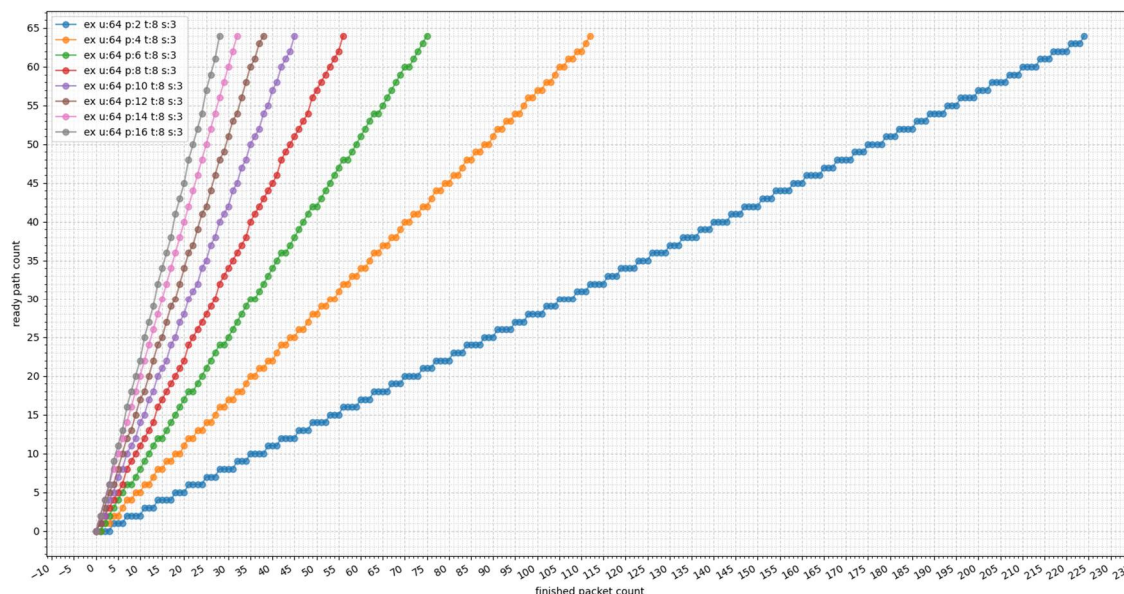


Рисунок 5 – Количество вычисленных пакетов для переменного количества задач

Таблица 4 – Параметры эксперимента 4

Параметр или константа	Значение
Размер карты, точек*точек	258x258
Число летающих роботов, шт.	64
Число наземных роботов, шт.	0
Длина пути, точек	100
Количество логических потоков, шт.	8
Количество задач в пакете, шт.	8
Количество физических потоков, шт.	8(16 w/ Hyper-Threading)
Количество разбиений пути, шт.	0:1:8

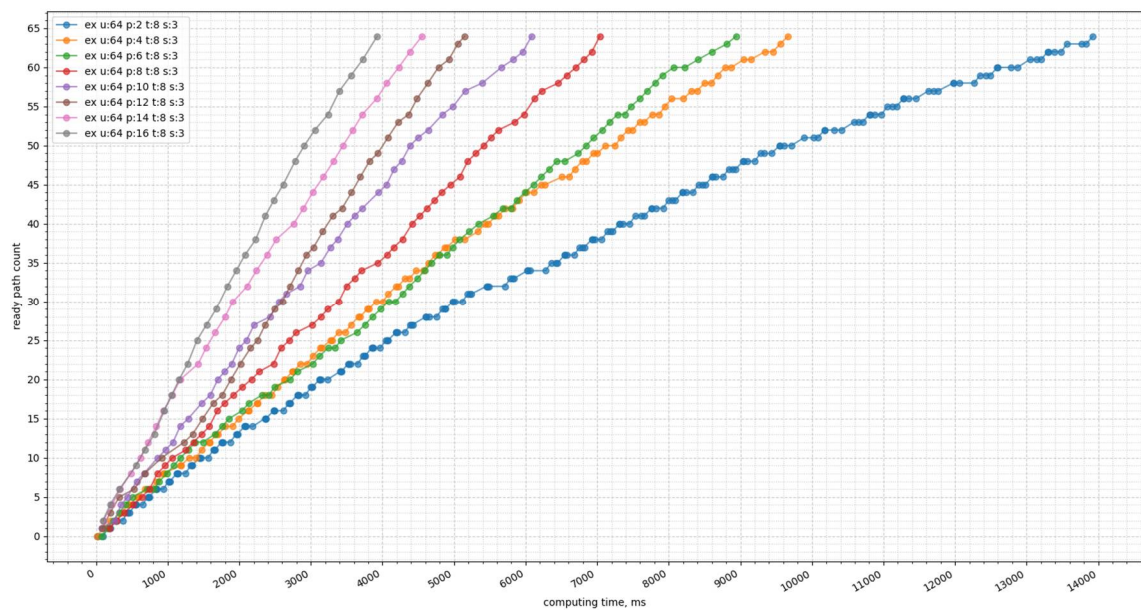


Рисунок 6 – Время расчета для переменного количества задач

Еще при рассмотрении стратегий обследования местности возник вопрос о количестве разбиений искомого пути летающих роботов. В соответствии с исходной постановкой задачи для каждого дрона может быть установлено любое количество контрольных точек. При этом в случае расчета на малых участках уменьшается размер матрицы покрытия для метода Ли, что предполагает уменьшение времени расчета. Однако согласно полученным результатам, наименьшее время расчета достигнуто при отсутствии разбиений. Следует заметить, что эксперимент проводился на модельной карте без препятствий и полученный результат по времени имеет сравнимый порядок. Следовательно, на реальных данных возможны более сложные случаи, и выбор субоптимального количества разбиений вариативен в зависимости от локации. Также для создания покрытия местности дронами с некоторым радиусом обзора маршрут должен иметь контрольные точки. Таким образом, субоптимальным количеством разбиений является минимально необходимое для выбранной стратегией исследования местности. Зависимости числа и времени вычисления пакетов при различном размере разбиений приведены на рисунках 5 и 6 соответственно.

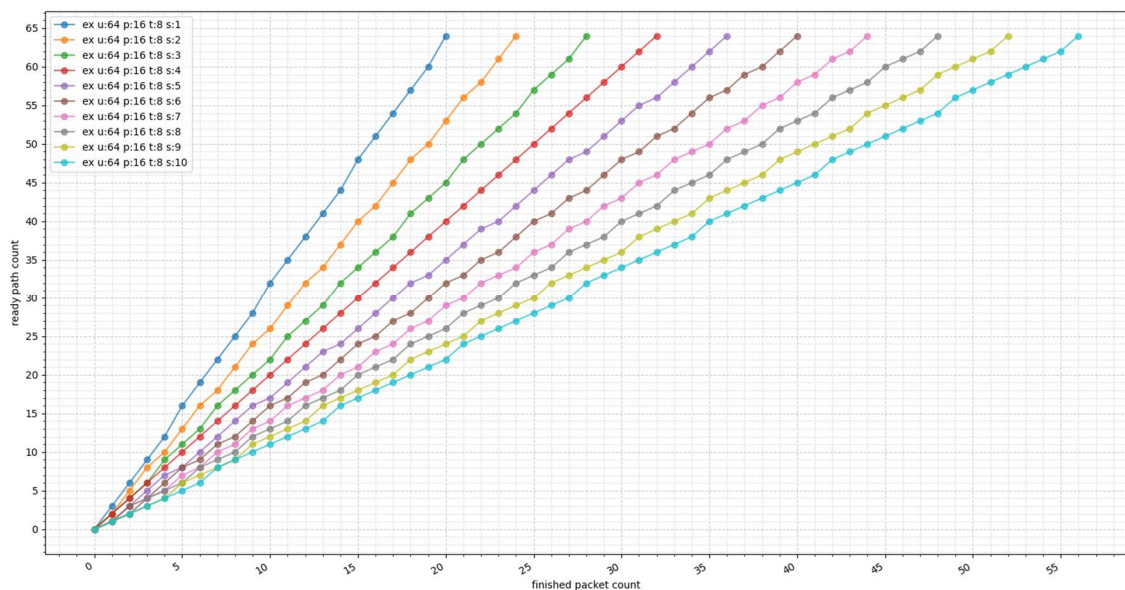


Рисунок 7 – Количество вычисленных пакетов для переменного количества разбиений

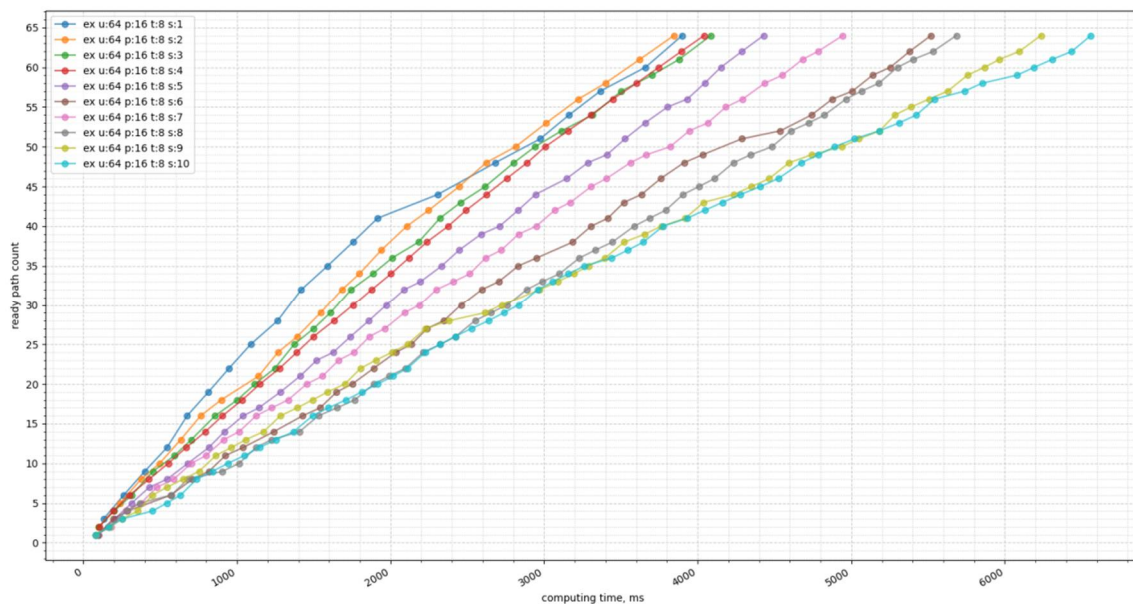


Рисунок 8 – Время расчета для переменного количества разбиений

Заключение. В соответствии с результатами проведенных экспериментов предложена методика конфигурации параметров для менеджера потоков с целью получения субоптимальных параметров, позволяющих как можно раньше запустить дроны для исследования местности. Эта конфигурация ориентирована исключительно на практическое применение, так как отвечает на вопрос «что сделать, чтобы запустить дронов как можно быстрее». В контексте общего исследования результат является смежным между статической и динамической моделью, что позволит в дальнейшем перейти к коррекциям путей «на лету» в изменяющихся внешних условиях с готовым механизмом параллелизации.

Также необходимо заметить, что исходный код менеджера потоков возможно дополнить модулем конфигурации и сделать полностью отжуждаемым, т.е. независимым от задачи. Подобная конструкция представляет самооптимизирующееся решение «из коробки» и позволит свести к минимуму время внедрения в сторонние программные комплексы.

Библиографический список

1. Попов, С. Г. Исследование алгоритма обследования местности гетерогенной группой автономных мобильных роботов / С. Г. Попов, А. С. Крашенинников, М. В. Чуватов // Робототехника и техническая кибернетика. – Санкт-Петербург : ЦНИИ РТК, 2019. – Т. 7, № 4. – С. 278–290.
2. Попов, С. Г. Исследование алгоритма многопоточного поиска путей гетерогенной группы автономных мобильных роботов / С. Г. Попов, А. С. Крашенинников, А. С. Тучков, А. И. Осадчий // Информация и космос. – 2020. – № 4. – С. 57–65.
3. Пашенко, Д. В. Многопоточность как фактор, влияющий на быстродействие и производительность параллельной вычислительной системы при выполнении операций сортировки данных / Д. В. Пашенко, А. И. Мартышкин, А. Н. Перекусихина // XXI век: итоги прошлого и проблемы настоящего плюс. – 2019. – Т. 8, № 3. – С. 23–29.
4. Койбаш, А. А. Модификация алгоритма A* для прогнозирования траектории движения подвижного объекта в распределенных вычислительных системах / А. А. Койбаш, Т. В. Завадская, С. В. Кривошеев // Проблемы искусственного интеллекта. – 2018. – № 3 (10).
5. Подвальный, С. Л. Разработка специального программного обеспечения решения транспортных задач модифицированным генетическим алгоритмом с использованием многопоточности / С. Л. Подвальный, Д. А. Вдовин // Вестник Воронежского государственного технического университета. – 2020. – Т. 16, № 4.
6. Зюбин, В. Е. Распределение вычислительных ресурсов в средах с многопоточной реализацией гиперавтомата / В. Е. Зюбин, А. Д. Петухов // Идентификация систем и задачи управления : материалы III Международ. конф. SICPRO. – 2004. – Т. 4. – С. 446–463.
7. Дергачев, С. А. Об одном вопросе реализации алгоритма планирования траектории A / С. А. Дергачев, К. С. Яковлев // Беспилотные транспортные средства с элементами искусственного интеллекта (БТС-ИИ-2019) : Пятый Всероссийский научно-практический семинар. – 2019. – С. 66–76.
8. Лавренов, Р. О. Планирование маршрута для беспилотного наземного робота с учетом множества критериев оптимизации / Р. О. Лавренов, И. М. Афанасьев, Е. А. Магид // Беспилотные транспортные средства с элементами искусственного интеллекта. – 2016. – С. 10–20.

9. Дивеев, А. И. Эволюционные методы вычислений для синтеза управления группой роботов и поиска оптимальных траекторий их движения / А. И. Дивеев, Е. Ю. Шмалько // Cloud of science. – 2017. – Т. 4, № 3.
10. Зенкевич, С. Л. Управление группой мобильных роботов с помощью сопровождающего дрона / С. Л. Зенкевич, А. В. Назарова, Х. Цзяньвень // Робототехника и техническая кибернетика. – 2019. – Т. 7, № 3. – С. 208–214.
11. Даринцев, О. В. Декомпозиция задач в группе роботов с использованием технологий облачных вычислений / О. В. Даринцев, А. Б. Мигранов // Труды Института механики им. П. П. Мавлютова УНЦ РАН. – 2017. – Т. 12, № 1. – С. 83–88.
12. Агафонов, А. А. Исследование эффективности вычисления надежного кратчайшего пути с использованием GPU / А. А. Агафонов, А. И. Максимов, А. А. Бородинов // Информационные технологии и нанотехнологии (ИТНТ-2020). – 2020. – С. 732–738.
13. Янчин, И. А. Параллелизация генетических алгоритмов при поиске пути / И. А. Янчин, О. Н. Петров // Международная конференция по мягким вычислениям и измерениям / Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный электротехнический университет ЛЭТИ им. В. И. Ульянова (Ленина)», 2016. – Т. 1, № 1–3. – С. 520–523.
14. Tullsen, D. M. Simultaneous multithreading: Maximizing on-chip parallelism / D. M., Tullsen, S. J. Eggers, H. M. Levy // Proceedings of the 22nd annual international symposium on Computer architecture. – 1995. – P. 392–403.
15. Roth, A. Speculative data-driven multithreading / A. Roth, G. S. Sohi // Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture. – IEEE, 2001. – P. 37–48.
16. Kim, H. Multithreading optimization techniques for sensor network operating systems / H. Kim, H. Cha // European Conference on Wireless Sensor Networks. – Springer, Berlin, Heidelberg, 2007. – P. 293–308.
17. Kuorilehto, M. Sensors: A new operating system for time critical wsn applications / M. Kuorilehto et al. // International Workshop on Embedded Computer Systems. – Springer, Berlin, Heidelberg, 2007. – P. 431–442.
18. Liu, Z. A Dynamic Fusion Pathfinding Algorithm Using Delaunay Triangulation and Improved A-Star for Mobile Robots / Z. Liu et al. // IEEE Access. – 2021. – Vol. 9. – P. 20602–20621.
19. Chudý, J. Emulating Centralized Control in Multi-Agent Pathfinding Using Decentralized Swarm of Reflex-Based Robots / J. Chudý, N. Popov, P. Surynek // 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). – IEEE, 2020. – P. 3998–4005.
20. Abiyev, R. H. et al. Robot pathfinding using vision based obstacle detection / R. H. Abiyev et al. // 2017 3rd IEEE International Conference on Cybernetics (CYBCONF). – IEEE, 2017. – P. 1–6.
21. Liu Y., Chen M., Huang H. Multi-agent Pathfinding Based on Improved Cooperative A* in Kiva System // 2019 5th International Conference on Control, Automation and Robotics (ICCAR). – IEEE, 2019. – P. 633–638.

References

1. Popov, S. G., Krasheninnikov, A. S., Chuvatov, M. V. Issledovaniye algoritma obsledovaniya mestnosti geterogennoy gruppy avtonomnykh mobilnykh robotov [The study of the survey algorithm of the heterogeneous group of autonomous mobile robots]. *Robototekhnika i tekhnicheskaya kibernetika* [Robotics and Technical Cybernetics]. Saint Petersburg, TsNII RTK, 2019, vol. 7, no. 4, pp. 278–290.
2. Popov, S. G., Krasheninnikov, A. S., Tuchkov, A. S., Osadchuy, A. I. Issledovanie algoritma mnogopotchnogo poiska putey geterogennoy gruppy avtonomnykh mobilnykh robotov [Study of the algorithm for multi-thread path search of a heterogeneous group of autonomous mobile robots]. *Informatsiya i kosmos* [Information and Space], 2020, № 4, pp. 57–65.
3. Paschenko, D. V., Martushkin, A. I., Perekusikhina, A. N. Mnogopotchnost kak faktor, vliyayushchiy na bystrodeystviye i proizvoditelnost parallelnoy vychislitelnoy sistemy pri vypolnenii operatsiy sortirovki dannykh [Multi-threading as a factor affecting the speed and performance of a parallel computing system when performing data sorting operations]. *XXI vek: itogi proshlogo i problemy nastoyashchego plyus* [XXI century: The results of the past and the problems of the present plus], 2019, vol. 8, no. 3, pp. 23–29.
4. Koybash, A. A., Zavadskaya, T. V., Krivosheev, S. V. Modifikatsiya algoritma A* dlya prognozirovaniya trayektorii dvizheniya podvizhnogo obyekt v raspredelennykh vychislitelnykh sistemakh [Modification of the A* algorithm for predicting the trajectory of a moving object in distributed computing system]. *Problemy iskusstvennogo intellekta* [AI Problems], 2018, vol. 3 (10).
5. Podvalny, S. L., Vdovin, D. A. Razrabotka spetsialnogo programmnoy obespecheniya resheniya transportnykh zadach modifitsirovannym geneticheskim algoritmom s ispolzovaniem mnogopotchnosti [Development of special softawre for solving transportation problems with a modified genetic algorithm using multi-threading]. *Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta* [Bulletin of Voronezh State Technical University], 2020, vol. 16, no. 4.
6. Zyubin, V. E., Petukhov, A. D. Raspredeleniye vychislitelnykh resursov v sredakh s mnogopotchnoy realizatsiyey giperavtomata [Distribution of computing resources in environments with multi-threaded implementation of a hyperautomat]. *Identifikatsiya sistem i zadachi upravleniya : materialy III Mezhdunarodnoy konferentsii SICPRO* [System identification and control tasks : proceedings of the III International conference SICPRO], 2004, vol. 4, pp. 446–463.
7. Dergachev, S. A., Yakovlev, K. S. Ob odnom voprose realizatsii algoritma planirovaniya trayektorii A* [On one issue of the implementation of the trajectory planning algorithm A*]. *Bespilotnyye transportnyye sredstva s elementami iskusstvennogo intellekta : Pyaty Vserossiyskiy nauchno-prakticheskiy seminar (BTS-AI-2019)* [Unmanned Vehicles with Artificial Intelligence Elements Fifth All-Russian Scientific and Practical Seminar], 2019, pp. 66–76.

8. Lavrenov, R. O., Afanasev, I. M., Magid, E. A. Planirovaniye marshruta dlya bespilotnogo nazemnogo robota s uchetoм mnozhestva kriteriyev optimizatsii [Route planning for an unmanned ground robot taking into account a variety of optimization criteria]. *Bespilotnyye transportnyye sredstva s elementami iskusstvennogo intellekta* [Unmanned vehicles with artificial intelligence elements], 2016, pp. 10–20.
9. Diveev, A. I., Shamlko, E. U., Evolyutsionnyye metody vychisleniy dlya sinteza upravleniya gruppy robotov i poiska optimalnykh trayektoriy ikh dvizheniya [Evolutionary computational methods for the synthesis of control of a group of robots and the search for optimal trajectories of their movement]. *Cloud of science*, 2017, vol. 4, no. 3.
10. Zenkevich, S. L., Nazarova, A. V., Jianwen, H., Tszyanven, Kh. Upravleniye gruppy mobilnykh robotov s pomoshchyu soprovozhdayushchego drona [Controlling a group of mobile robots using an escort drone]. *Robototekhnika i tekhnicheskaya kibernetika* [Robotics and technical cybernetics], 2019, vol. 7, no. 3, pp. 208–214.
11. Darintsev, O. V., Migranov, A. B. Dekompozitsiya zadach v gruppe robotov s ispolzovaniyem tekhnologiy oblachnykh vychisleniy [Decomposition of tasks in a group of robots using cloud computing technologies]. *Trudy Instituta mekhaniki im. RR Mavlyutova UNTs RAN* [Proceedings of the Institute of Mechanics. RR Mavlyutova UC RAS], 2017, vol. 12, no. 1, pp. 83–88.
12. Agafonov, A. A., Maksimov, A. I., Borodinov, A. A. Issledovaniye effektivnosti vychisleniya nadezhnogo krachayshego puti s ispolzovaniyem GPU [Investigation the Efficiency of Computing a Reliable Shortest Path Using a GPU]. *Informatsionnyye tekhnologii i nanotekhnologii IITHT-2020* [Information technology and nanotechnology (IITHT-2020)], 2020, pp. 732–738.
13. Yanchin, I. A., Petrov, O. N. Parallelizatsiya geneticheskikh algoritmov pri poiske puti [Parallelization of genetic algorithms in pathfinding]. *Mezhdunarodnaya konferentsiya po myagkim vychisleniyam i izmereniyam. Federalnoye gosudarstvennoye avtonomnoye obrazovatelnoye uchrezhdeniye vysshego obrazovaniya "Sankt-Peterburgskiy gosudarstvennyy elektrotekhnicheskiy universitet LETI im. VI Ulyanova (Lenina)"* [International Conference on Soft Computing and Measurements. Federal State Autonomous Educational Institution of Higher Education "V.I. Ulyanov (Lenin) Saint-Petersburg State Electrotechnical University LETI"], 2016, vol. 1, no. 1–3, pp. 520–523.
14. Tullsen, D. M., Eggers, S. J., Levy, H. M. Simultaneous multithreading: Maximizing on-chip parallelism. *Proceedings of the 22nd annual international symposium on Computer architecture*, 1995, pp. 392–403.
15. Roth, A., Sohi, G. S. Speculative data-driven multithreading. *Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture*. IEEE, 2001, pp. 37–48.
16. Kim, H., Cha, H. Multithreading optimization techniques for sensor network operating systems. *European Conference on Wireless Sensor Networks*. Springer, Berlin, Heidelberg, 2007, pp. 293–308.
17. Kuorilehto, M. et al. Sensoros: A new operating system for time critical wsn applications. *International Workshop on Embedded Computer Systems*. Springer, Berlin, Heidelberg, 2007, pp. 431–442.
18. Liu, Z. et al. A Dynamic Fusion Pathfinding Algorithm Using Delaunay Triangulation and Improved A-Star for Mobile Robots. *IEEE Access*, 2021, vol. 9, pp. 20602–20621.
19. Chudý, J., Popov, N., Surynek, P. Emulating Centralized Control in Multi-Agent Pathfinding Using Decentralized Swarm of Reflex-Based Robots. *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 3998–4005.
20. Abiyev, R. H. et al. Robot pathfinding using vision based obstacle detection. *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*. IEEE, 2017, pp. 1–6.
21. Liu, Y., Chen, M., Huang, H. Multi-agent Pathfinding Based on Improved Cooperative A* in Kiva System. *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2019, pp. 633–638.