



# МОДЕЛИ ДЕКОМПОЗИЦИИ И ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ КООРДИНАЦИИ ДВИЖЕНИЯ ПОДВИЖНЫХ СРЕДСТВ

УДК 004.9

**АРКАТОВ Денис Борисович**

ассистент кафедры Автоматизированных систем управления Национального технического университета «ХПИ».

**Научные интересы:** информационные технологии, математическое моделирование и оптимизация технологических процессов.

**e-mail:** denarkatov@gmail.com

## ВВЕДЕНИЕ

Автоматизация диспетчерского управления железнодорожного транспорта Украины является современным и актуальным вопросом. Особое место в этой проблеме занимает разработка алгоритмов координации подвижных средств (ПС), которые находятся в зоне железнодорожных диспетчерских пунктов. При разработке этих алгоритмов самым важным требованием является обеспечение безопасности и высокой эффективности перевозок всей совокупности поездов, а также принятие таких решений, которые были бы оптимальными с экономической точки зрения.

Применение параллельных вычислений является стратегическим направлением развития системы связи и передачи данных разрабатываемой автоматизированной системы координации движения подвижных средств [1, 2]. Это обстоятельство вызвано не только принципиальным ограничением максимально возможного быстродействия обычных последовательных персональных компьютеров, но и практически постоянной необходимостью осуществления принятия решений за достаточно малый интервал времени.

Организация параллельности вычислений при решении задачи координации и диспетчеризации движения, когда в один и тот же момент времени выполняется одновременно несколько операций обработки

данных, осуществляется, в основном, введением избыточности функциональных устройств (многопроцессорности). В этом случае можно достичь ускорения процесса решения вычислительной задачи, если осуществить разделение применяемого алгоритма на информационно независимые части и организовать выполнение каждой части вычислений на разных серверах автоматизированной системы. Подобный подход позволяет выполнять необходимые вычисления с меньшими затратами времени, и возможность получения максимального ускорения ограничивается только числом имеющихся серверов, находящихся в диспетчерских пунктах, и количеством "независимых" частей в выполняемых вычислениях.

Вопросы параллельных вычислений рассматривались в работах Гергея В. П. [3,4], Ивенса Д. [5], Воеводина В. В. [6], Бусленко Н. П. [7], однако в них задача распараллеливания не решается с точки зрения повышения эффективности, оптимизации вычислительной способности алгоритмов с целью получения результата за определенное время.

## ПОСТАНОВКА ПРОБЛЕМЫ

Для повышения эффективности и безопасности железнодорожного движения, а также выполнения временных ограничений необходимо разделение подвижных средств на отдельные группы, для каждой из

которых в параллельном режиме будет выполняться алгоритм координации. В [8] было представлено общее деление железнодорожной зоны Украины на отдельные диспетчерские зоны (ДЗ). Внутри каждой ДЗ осуществляется диспетчеризация тех ПС, которые осуществляют движение только внутри своей ДЗ. Если же подвижные средства двигаются между ДЗ, то координация движения осуществляется посредством диспетчера железнодорожной зоны Украины.

Между сервером приложений, осуществляющим решение задачи координации, осуществляется взаимосвязь с АРМом диспетчера-графиста, содержащего расписание движения, сервером базы данных контрольных точек, а также бортовыми интеллектуальными комплексами [2] ПС, передающими в реальном времени навигационную информацию (см. рис. 1).

Под навигационной информацией понимается набор координат и скорость движения ПС, полученные с помощью GPS/GPRS-модема. За счет координат осуществляется позиционирование ПС на специализированной навигационной карте [2], а текущая скорость движения дает возможность оценки временного интервала до прохода контрольных точек.

Под контрольной точкой следует понимать точку на навигационной карте, в момент прохода которой необходимо для конкретного ПС проверить условие бесконфликтности и, при необходимости выработать управляющее воздействие. Контрольные точки находятся на необходимом расстоянии от точек возможных конфликтов, что обеспечивает временной интервал для принятия решений.

**Целями данной статьи** являются моделирование параллельных вычислений для решения задачи координации движения ПС на основе взаимодействия комплекса алгоритмов, разработка параллельного алгоритма, а также оценка его эффективности.

### МОДЕЛЬ КООРДИНАЦИИ ДВИЖЕНИЯ ДИСПЕТЧЕРСКИХ ЗОН В ПАРАЛЛЕЛЬНОМ РЕЖИМЕ

Пусть в диспетчерской зоне управления находится некоторое множество поездов  $S_i = \{1, 2, \dots, j, \dots, N-1\}$ , где  $r$  соответствует очередности их появления в зоне управления,  $i$  – индекс рассматриваемой ДЗ. На навигационной карте задано множество контрольных точек

$P_i = (1, \dots, M_i)$ . Для каждого поезда из этого множества, с учетом оперативной информации, определяется расчетное время отправления и прибытий  $t^r$  ( $r = 1, \dots, N-1$ ). Упорядочим множество  $S_i$  по возрастанию расчетного времени  $t^r$  и обозначим его через  $S_i^* = \{1, 2, \dots, N-1\}$ . Очередность прибытия и отправления поездов множества  $S_i$  устанавливается соответственно увеличению расчетного времени для каждого поезда и соответствует последовательности элементов множества  $S_i^*$ . С учетом указанного выше, расчетное время для  $r$ -го поезда обозначим через  $t_j^r$ , где нижний индекс на очередность прибытий и отправления.

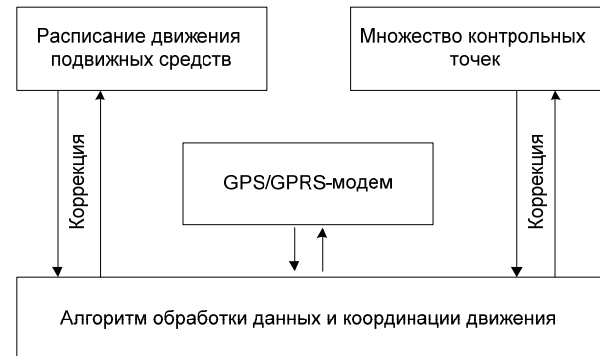


Рисунок 1 – Информационное взаимодействие в системе координации движения подвижных средств

Будем считать, что для множества поездов  $S_i$  определено действительное время прибытий и отправления  $t_j^r$  ( $r \in S$ ),  $j \in S_i^*$  с учетом безопасности железнодорожного движения на путях. При появлении  $N$ -го поезда с расчетным временем  $t_x^N$  определяем его очередность прибытия, то есть неизвестный индекс  $x$ , который определяется из условия выполнения неравенства:

$$t_{i-1}^r < t_x^N < t_i^{S_i} \quad (1)$$

Тогда  $x = i$ , а нижние индексы поездов, начиная с  $t_j^S$ , увеличиваются на единицу.

Потом осуществляется проверка бесконфликтности движения

$$t_i^N - t_{i-1}^r \geq \tau, t_{i+1}^{S_i^*} - t_i^N \geq \tau \quad (2)$$

где  $\tau$  – минимально безопасный интервал между поездами на железнодорожном пути, с которого осуществляется прибытие или отправление. Когда оба

неравенства выполняются, то действительное время  $N$ -го поезда будет равняться  $t_i^N = t_i^{N^*}$ . Если хотя бы одно из неравенств (2) не выполняется, то для  $N$ -го поезда условие безопасности не гарантируется и возникает конфликт. При устранении этого конфликта возможно невыполнение условия (2) для других поездов множества  $S_i$ . Все поезда, для которых это условие

нарушается, образуют подмножество  $S_{1i}^* \subset S_i^*$ . Мощность  $S_{1i}^*$  равняется количеству поездов, связанных между собой, и вычисляется следующим образом. Определяется количество поездов, которое удовлетворяет неравенствам

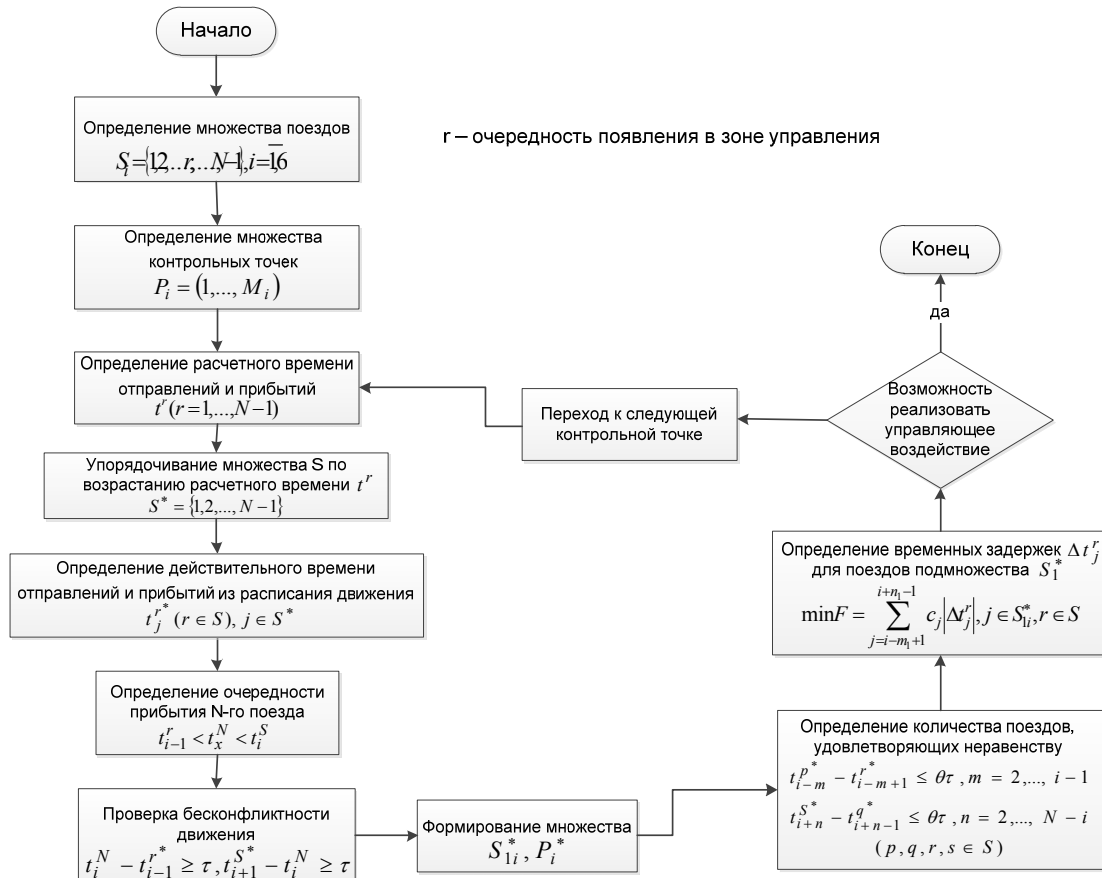


Рисунок 2 – Алгоритм координации движения в параллельном режиме для диспетчерских зон

$$\begin{aligned}
 t_{i-m}^{p*} - t_{i-m+1}^{r*} &\leq \theta\tau, m = 2, \dots, i-1 \\
 t_{i+n}^{s*} - t_{i+n-1}^{q*} &\leq \theta\tau, n = 2, \dots, N-i \\
 (p, q, r, s \in S_i)
 \end{aligned}
 \quad (3)$$

где  $\theta=2$ , поскольку все поезда, кроме  $p$ -го, разделены безопасным интервалом  $\tau$ ;

$t_{i-m}^p$  – время прибытия или отправления поезда, порядковый номер которого отличается от  $p$ -го на  $m$  единиц (аналогично следует понимать  $t_{i+n-1}^{q*}$ ).

Пусть для  $n_1$  и  $m_1$  неравенства (3) уже не выполняются, тогда мощность подмножества  $S_{1i}^*$  равня-

ется  $n_1 + m_1$  и  $S_{1i}^* = \{i - m, i - m + 1, \dots, i + n - 1\}$ . Блок-схема определения подмножества  $S_{1i}^*$  приведена в [4]. Для поездов этого подмножества снова решается задача определения действительного времени прибытия или отправления с учетом их безопасности движения на железнодорожных путях.

Эта задача состоит в следующем. Необходимо определить задержки  $\Delta t_j^r$  для поездов подмножества  $S_{1i}^*$ , которые минимизируют линейную форму

$$\min F = \sum_{j=i-m_1+1}^{i+n_1-1} c_j |\Delta t_j^r|, j \in S_{1i}^*, r \in S_i,$$

где  $c_j$  – весовые коэффициенты  $j$ -го поезда, которые учитывают себестоимость одного производственного часа для оптимального режима езды поезда. Безопасность движения на железнодорожных путях накладывает на  $\Delta t_j^r$  ограничения, которые имеют вид

$$t_{j+1}^r - \Delta t_{j+1}^r - t_j^s + \Delta t_j^s \geq \tau, j = i - m_1 + 1, \dots, i + n_1 - 1, \\ r, s \in S_i$$

где  $t_j^{s*} = t_j^s - \Delta t_j^s$  – действительное время прибытия  $s$ -го поезда с очередностью  $j \in S_{i_i}^*$ . Считается, что все  $\Delta t_j^r$  возможно устранить.

После формирования управляющих воздействий необходима проверка возможности их реализации подвижным объектом: если их применить невозможно до прохода точки конфликта, то необходимо сформировать новые управляющие воздействия с целью возврата подвижного средства в график до следующей контрольной точки.

Общая схема алгоритма координации движения ПС, который выполняется параллельно для всех групп поездов диспетчерских зон представлена на рис. 2.

### МОДЕЛЬ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ

При разработке параллельных алгоритмов решения сложных научно-технических задач принципиальным моментом является анализ эффективности использования параллелизма [5, 6], состоящий обычно в оценке получаемого ускорения процесса вычислений (сокращения времени решения задачи). Формирование подобных оценок ускорения может осуществляться применительно к выбранному вычислительному алгоритму (оценка эффективности распараллеливания конкретного алгоритма). Другой важный подход состоит в построении оценок максимально возможного ускорения процесса решения задачи конкретного типа

(оценка эффективности параллельного способа решения задачи).

Представим множество операций, выполняемых в разрабатываемом алгоритме решения задачи координации движения подвижных средств, и существующие между операциями информационные зависимости в виде ациклического ориентированного графа

$$G = (V, R),$$

где  $V$  – множество вершин графа, представляющих выполняемые операции алгоритма, а  $R$  – множество дуг графа (при этом дуга  $r = (i, j)$  принадлежит графу только в том случае, если операция  $j$  использует результат выполнения операции  $i$ ). На рис. 3 представлен граф алгоритма параллельного вычисления коррекции расписания движения ПС на основе полученной информации о местоположении подвижных средств и соблюдения условий бесконфликтности движения. В качестве исходных данных используется навигационная информация, которая передается на сервера баз данных, находящихся в каждой из диспетчерских зон. Для уменьшения временного интервала решения задачи координации расчеты соответствующих алгоритмов для множеств поездов, движущихся в различных диспетчерских зонах, должны выполняться одновременно. В случае движения подвижных средств через несколько диспетчерских зон результаты алгоритмов координации должны быть объединены для дальнейших расчетов и коррекции расписания движения.

В рассматриваемой вычислительной модели алгоритма вершины без входных дуг могут использоваться для задания операций получения навигационной информации, т.е. исходных данных, а вершины без выходных дуг – для операций получения результатов вычислений, т.е. необходимые корректировки расписания движения подвижных средств.

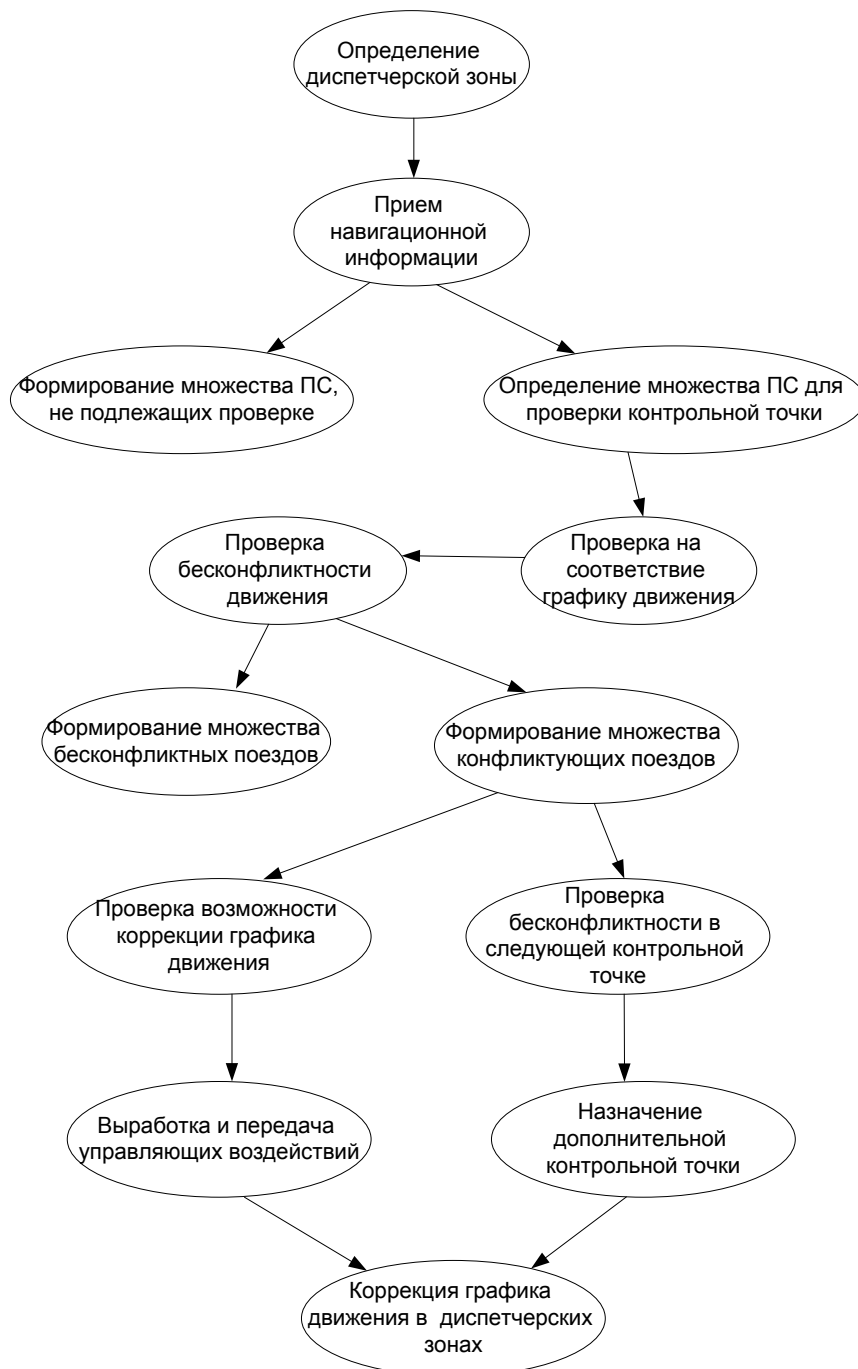


Рисунок 3 – Схема параллельного алгоритма коррекции расписания движения ПС

### ОЦЕНКА ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА

Пусть  $p$  – это количество серверов, используемых для выполнения алгоритма координации движения ПС. Тогда для параллельного выполнения вычисле-

ний необходимо задать множество (расписание выполнения операций)

$$H_p = \{(i, P_i, t_i) : i \in V\},$$

в котором для каждой операции  $i \in V$  указывается номер используемого для выполнения операции сервера диспетчерской зоны  $P_i$  и время начала выполнения операции  $t_i$ . Для того чтобы расписание было

реализуемым, необходимо выполнение следующих требований при задании множества  $H_p$  [3]:

1)  $\forall i, j \in V, t_i = t_j \Rightarrow P_i \neq P_j$ , т.е. один и тот же сервер не должен назначаться разным операциям в один и тот же момент;

2)  $\forall i, j \in R \Rightarrow t_i \geq t_j + 1$ , т.е. к назначаемому моменту выполнения операции все необходимые данные уже должны быть вычислены.

Вычислительная схема алгоритма  $G$  совместно с расписанием  $H_p$  может рассматриваться как модель параллельного алгоритма  $A_p(G, H_p)$ , исполняемого с использованием  $p$  серверов. Время выполнения параллельного алгоритма определяется максимальным значением времени, применяемым в расписании

$$T_p(G, H_p) = \max_{i \in V} (t_i + 1).$$

Для выбранного алгоритма координации движения подвижных средств необходимо использование расписания, обеспечивающего минимальное время выполнения алгоритма, что обеспечит актуальность сформированных управляющих воздействий и устранение конфликтных ситуаций

$$T_p(G) = \min_{H_p} T_p(G, H_p).$$

Уменьшение времени выполнения может быть обеспечено и путем подбора наилучшей вычислительной схемы

$$T_p = \min_G T_p G.$$

Оценки  $T_p(G, H_p)$ ,  $T_p G$  и  $T_p$  могут быть применены в качестве показателей времени выполнения параллельного алгоритма. Кроме того, для анализа максимально возможного параллелизма можно определить оценку наиболее быстрого исполнения алгоритма

$$T_\infty = \min_{p \geq 1} T_p$$

Оценку  $T_\infty$  можно рассматривать как минимально возможное время выполнения параллельного алгоритма при использовании неограниченного количества серверов (концепция вычислительной системы с бесконечным количеством серверов широко применяется при теоретическом анализе параллельных вычислений).

Оценка  $T_1$  определяет время выполнения алгоритма при использовании одного сервера и представляет, тем самым, время выполнения последовательного варианта алгоритма решения задачи координации. Построение подобной оценки является важной задачей при анализе параллельных алгоритмов, поскольку она необходима для определения эффекта использования параллелизма (ускорения времени решения задачи). Очевидно, что

$$T_1(G) = |\bar{V}|,$$

где  $|\bar{V}|$  есть количество вершин вычислительной схемы без вершин ввода. Важно отметить, что если при определении оценки ограничиться рассмотрением только одного выбранного алгоритма решения задачи и использовать величину

$$T_1 = \min_G T_1 G,$$

то получаемые при такой оценке показатели ускорения будут характеризовать эффективность распараллеливания выбранного алгоритма. Для оценки эффективности параллельного решения исследуемой вычислительной задачи время последовательного решения следует определять с учетом различных последовательных алгоритмов, т.е. использовать величину

$$T_1^* = \min T_1,$$

где операция минимума берется по множеству всех возможных последовательных алгоритмов решения данной задачи.

Ускорение, получаемое при использовании параллельного алгоритма для  $p$  серверов, по сравнению с последовательным вариантом выполнения вычислений определяется величиной

$$S_p(n) = \frac{T_1(n)}{T_p(n)},$$

т.е. как отношение времени последовательного решения множества задач на одном сервере к времени выполнения параллельного алгоритма (величина  $n$  применяется для параметризации вычислительной сложности решаемой задачи и может пониматься, например, как количество входных данных задачи).

Эффективность использования параллельным алгоритмом серверов при решении задачи координации определяется соотношением

$$E_p(n) = \frac{T_1(n)}{pT_p(n)} = \frac{S_p(n)}{p},$$

т.е. величина эффективности определяет среднюю долю времени выполнения алгоритма, в течение которой серверы реально задействованы для решения задачи.

При выборе надлежащего параллельного способа решения задачи может оказаться полезной оценка стоимости вычислений, определяемой как произведение времени параллельного решения задачи и числа используемых процессоров

$$C_p = pT_p.$$

В связи с этим можно определить понятие стоимости-оптимального параллельного алгоритма как метода, стоимость которого является пропорциональной времени выполнения наилучшего последовательного алгоритма.

В рассматриваемой задаче координации движения ПС ускорение за счет распараллеливания вычислений определяется следующим образом. Количество операций последовательного варианта алгоритма решения задачи  $K_{\text{послед}} = 72$ . Количество операций параллельного варианта алгоритма решения задачи  $K_{\text{пар}} = 12$ . Поскольку считается, что выполнение любых вычислительных операций является одинаковым и единичным, то  $T_1 = K_{\text{послед}} = 72$ , а  $T_p = K_{\text{пар}} = 12$ ,

поэтому показатели эффективности и ускорения алгоритма координации движения подвижных средств можно оценить как:

$$S_p(n) = \frac{T_1(n)}{T_p(n)} = \frac{72}{12} = 6.$$

$$E_p(n) = \frac{T_1(n)}{pT_p(n)} = \frac{S_p(n)}{p} = \frac{6}{6} = 1.$$

## ВЫВОДЫ

Таким образом, в данной статье получены следующие результаты:

- описана проблема распараллеливания вычислений подсистемы координации движения подвижных средств, а также принцип деления подвижных средств на группы для ускорения вычислений и выполнения временных ограничений;
- приведено описание алгоритма решения задачи координации с точки зрения его реализации в параллельных вычислениях;
- представлена модель, отражающая принцип разделения вычислительных процессов с целью их выполнения в параллельном режиме;
- осуществлена оценка эффективности применения параллельных вычислений в задаче координации движения подвижных средств.

## ЛИТЕРАТУРА:

1. Борушко Ю.М. АСУ «Навигация и управление» на основе спутниковых технологий для железнодорожного транспорта /Ю.М. Борушко, С.Б. Семенов, Н.Н. Титов //Спутниковые технологии и системы цифровой связи на службе железных дорог. – М.: ВНИИАС, 2007. – С.33-37.
2. Аркатов Д.Б. Постановка задачи «Разработка системы связи и передачи данных автоматизированной системы «Навигация и управление для железнодорожного транспорта Украины» /Д.Б. Аркатов, Ю.М. Борушко //Вісник Національного технічного університету «ХПІ». – 2012 – №29. – С.75-83.
3. Гергель В.П. Теория и практика параллельных вычислений. – М.: Интернет-университет информационных технологий, 2007. – 423 с.
4. Гергель В.П. Основы параллельных вычислений для многопроцессорных вычислительных систем. Учебное пособие /В.П. Гергель, Р.Г. Стронгин. – Нижний Новгород: Изд-во ННГУ им. Н.И. Лобачевского, 2003. – 184 с.
5. Ивенс Д. Системы параллельной обработки. – М.: Мир, 1985. – 416 с.
6. Воеводин В.В. Параллельные вычисления /В.В. Воеводин, Вл.В. Воеводин. – СПб.: БХВ-Петербург, 2002. – 608 с.
7. Бусленко Н.П. Моделирование сложных систем. – М.: Главная редакция физико-математической литературы из-ва «Наука», 1968. – 356 с.
8. Аркатов Д.Б. Задача координации движения подвижных средств //Системи обробки інформації. – 2012. – №7 (105). – С.40-43.
9. Аркатов Д.Б. Модели и методы автоматизации диспетчерского управления для железнодорожного транспорта Украины //Восточно-Европейский журнал передовых технологий. – 2013. – №1/10 (61). – С.61-63.