

DOI: 10.17516/1999-494X-0379

УДК 681.3.068

## The Direction of Development Parallel Programming of Software Tools Software Complexes Military Purpose

Michael A. Aksenov and Igor V. Lutikovov\*

*Military Academy of Aero-Space Defence  
named after the Marshal of Soviet Union G. K. Zhukov  
Tver, Russian Federation*

Received 23.11.2021, received in revised form 15.01.2022, accepted 06.02.2022

**Abstract.** The article discusses the issues of classification selection of preferred parallelization algorithms (with minimal execution time) implemented in parallel software development tools for multi-core (multiprocessor) computing systems with shared memory, based on the collected training statistical information of the parameters of the execution of cycles, previous program launches. The classification (decisive) rule for selecting the preferred algorithm in the learning process can be built only based on information from the training sample of previous runs, this information should not just be remembered, but should be generalized and transformed into an image most similar to the new recognizable image characterizing an unknown program cycle. To this end, the article presents three methods that allow, based on the mathematical apparatus of probability theory, to sequentially collect and transform training samples for each cyclic section of the program to the form of an image providing a functional relationship between the number of iterations of the current cycle and the preferred parallelization algorithm. The obtained dependence allows automatic selection (in real time) of the preferred parallelization algorithm based on classification selection by the input parameter (number of iterations) of the cycle in the generated project profile consisting of a set of tuples of preferred parallelization algorithms. The purpose of this article is to assess the impact of the execution time of parallelized cyclic sections of the target program, using the proposed method of automated selection of preferred algorithms, with multithreaded parallel execution of the program in multi-core (multiprocessor) PCs on the results of simulation of combat operations.

**Keywords:** software tool, modeling complex, parallelization algorithm, number of iterations, execution time.

Citation: Aksenov, M.A., Lutikovov, I. V. The direction of development parallel programming of software tools software complexes military purpose. J. Sib. Fed. Univ. Eng. & Technol., 2022, 15(1), 137–155. DOI: 10.17516/1999-494X-0379

## Направление развития параллельного программирования программных средств моделирующих комплексов военного назначения

**М. А. Аксенов, И. В. Лютиков**

*Военная академия воздушно-космической обороны  
им. Маршала Советского Союза Г. К. Жукова  
Российская Федерация, Тверь*

**Аннотация.** В статье рассмотрены вопросы классификационного выбора предпочтительных алгоритмов распараллеливания (с минимальным временем выполнения), реализованных в инструментальных средствах разработки параллельных программ для многоядерных (многoproцессорных) вычислительных систем с общей памятью, на основе собираемой обучающей статистической информации параметров выполнения циклов, предыдущих запусков программы. Классификационное (решающее) правило отбора предпочтительного алгоритма в процессе обучения может строиться только на основе информации из обучающей выборки предыдущих запусков. Эта информация должна не просто запоминаться, а обобщаться и преобразовываться в образ, наиболее похожий на новый распознаваемый образ, характеризующий неизвестный программный цикл. С этой целью в статье приведены три методики, позволяющие на основе математического аппарата теории вероятностей последовательно собирать и преобразовывать обучающие выборки под каждый циклический участок программы к виду образа (конечного профиля проекта), обеспечивающего функциональную связь между количеством итераций текущего цикла и предпочтительным алгоритмом распараллеливания. Полученная зависимость дает возможность производить автоматический выбор предпочтительного алгоритма распараллеливания на основе классификационного отбора по входному параметру (количества итераций) цикла в сформированном профиле проекта, состоящем из совокупности коротких предпочтительных алгоритмов распараллеливания. Целью данной статьи является оценка влияния времени выполнения распараллеленных циклических участков целевой программы с применением предлагаемого метода автоматизированного выбора предпочтительных алгоритмов при многопоточном параллельном выполнении программы в многоядерных (многoproцессорных) ПЭВМ на показатели результатов имитационного моделирования боевых действий.

**Ключевые слова:** программное средство, моделирующий комплекс, алгоритм распараллеливания, число итераций, время выполнения.

Цитирование: Аксенов, М. А. Направление развития параллельного программирования программных средств моделирующих комплексов военного назначения / М. А. Аксенов, И. В. Лютиков // Журн. Сиб. федер. ун-та. Техника и технологии, 2022, 15(1). С. 137–155. DOI: 10.17516/1999-494X-0379

### Введение

Эволюционное развитие средств воздушного нападения (СВН) приводит к повышению их боевых возможностей, базирующихся на тактико-технических характеристиках как количественного, так и качественного уровня, непрерывное совершенствование способов применения СВН – все это предъявляет сложные требования к системе противовоздушной обороны (ПВО) к отдельным ее элементам и взаимосвязям между ними, к появлению нового системного эффекта за счет эмерджентности. В основе тенденций развития средств ПВО и их технического

облика должны лежать не только особенности их боевого применения и, очевидно, тактико-технические характеристики средств СВН – «hard», но и программное обеспечение «soft», реализующее оптимальные методы и алгоритмы [9–20].

Одной из задач, возникающих при разработке программных средств моделирующих комплексов военного назначения (ПС МК ВН), является время выполнения готовой конечной программы. Существующие системы моделирования имеют ряд недостатков, одним из основных считается превышение нормативных сроков представления результатов имитационного моделирования [1].

В современных архитектурах ПЭВМ, поддерживающих параллелизм в многоядерных (многопроцессорных) вычислительных системах с общей памятью, наиболее распространены универсальные средства программирования – стандарты и технологии, предназначенные для создания современных параллельных программ, такие как языковые средства задания параллельности в программе, автоматические средства поиска параллельных вычислений и последующего представления их с использованием существующих библиотек поддержки параллельности [2].

Появилась тенденция объединения средств параллельного программирования [3] при использовании нескольких средств распараллеливания в одной разрабатываемой программе [4]. Сопоставительный анализ показывает, что разработанные ранее методы применения средств параллельного программирования не учитывают:

- случайный характер изменений параметров в структуре одних и тех же циклов программы по количеству итераций цикла и времени выполнения, что приводит к изменению критических по времени выполнения циклов и может стать причиной ошибочного выбора алгоритма распараллеливания;
- зависимость эффективности использования различных средств параллельного программирования от числа итераций цикла и от внутренней структуры цикла;
- разную зависимость временных затрат на выполнение одного и того же участка программного кода для разных средств распараллеливания при одних и тех же требованиях к аппаратно-программному обеспечению.

Основная идея состоит в создании системы параллельного программирования. С одной стороны, программист на традиционных языках программирования (C, C++) создает программу с параллельными вычислениями. Средствами эти параллельные вычисления должны быть распознаны. В случае проблем с анализом программист может добавить подсказки в программу, разрешив какой-либо конфликт, мешающий определению параллельности. Затем найденные параллельные вычисления автоматически представляются в параллельных терминах конкретной архитектуры. По сути, это среда разработки параллельных программ, в которой сбалансированы усилия человека и автоматических средств. В предложенном подходе основную роль по анализу, модернизации и инструментированию параллельного кода выполняет средство распараллеливания, а разработчик (программист) программами разрешает появляющиеся конфликты (если они, конечно, появляются), с которыми статический анализатор может не справиться. Для автоматического выбора предпочтительного алгоритма распараллеливания из доступного набора алгоритмов под каждый циклический участок текущей программы по минимальному времени выполнения цикла используются накопленные при тестировании

статистические данные в виде массива временных профилей программы и составленный на их основе профиль проекта, представляющий собой совокупность кортежей предпочтительных алгоритмов распараллеливания, функционально связанных с текущим числом итераций цикла. В результате по входным параметрам из выполняемой программы ПС МК ВН: номеру цикла и количеству итераций данного цикла с использованием составленного кортежа предпочтительных алгоритмов распараллеливания в реальном масштабе времени, определяется номер алгоритма распараллеливания, реализованного в средстве параллельного программирования, и автоматически подключается к процессу выполнения цикла с распараллеливанием выбранным средством.

### Основная часть

В статье описан подход, представляющий собой последовательность взаимосвязанных методик. Структурная схема предлагаемого подхода представлена на рис. 1. В рамках предлагаемого подхода были разработаны представленные ниже методики.

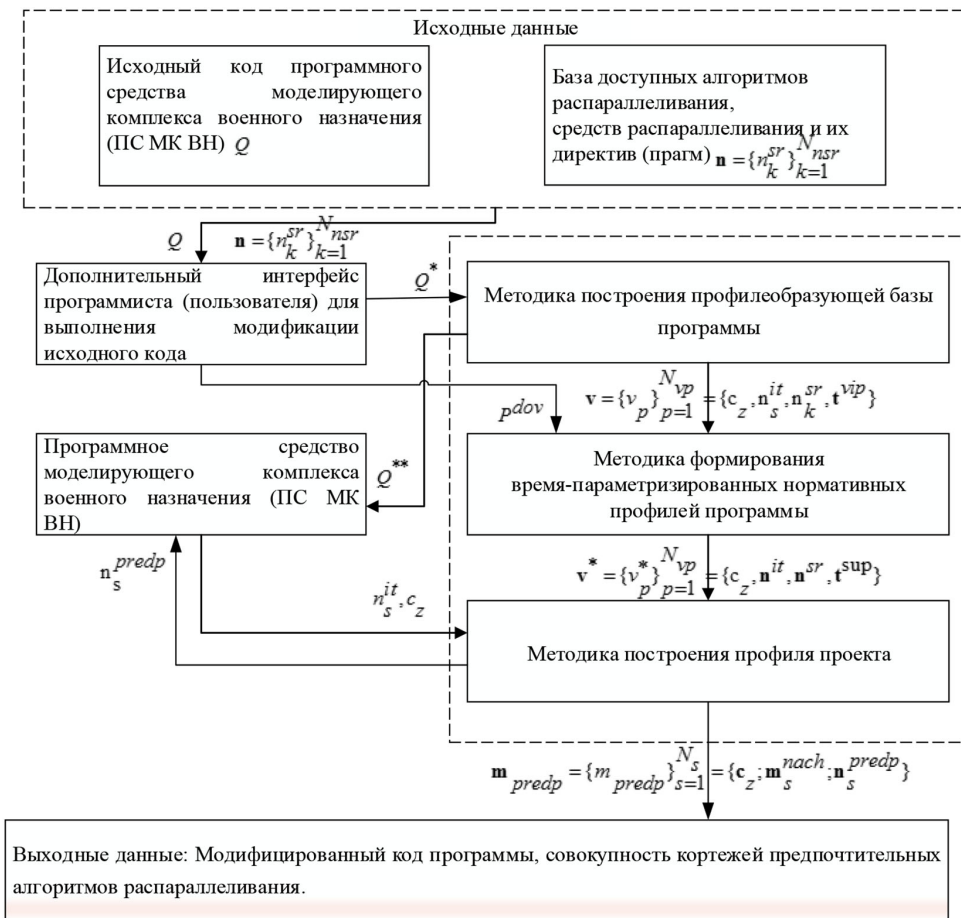


Рис. 1. Структурная схема построения программных средств моделирующих комплексов военного назначения

Fig. 1. Structural diagram of the construction of software modeling complexes for military purposes

## Методика построения профилеобразующей базы программы

Входные данные:

1.  $Q$  – исходный код программного средства моделирующего комплекса.

2.  $n = \{n_k^{sr}\}_{k=1}^{N_{mr}}$  – база доступных алгоритмов распараллеливания, средств распараллеливания и директив(прагм).

3. Интерфейс программиста для модификации и компиляции исходного кода.

Выходные данные:

$\{v_p\}_{p=1}^{N_{vp}} = c_z; n_s^{it}; n_k^{sr}; t_i^{vip}$  – профилеобразующая база программы в виде набора кортежей таблицы, где  $c_z$  – порядковый номер цикла программы;  $n^{it}$  – число итераций цикла;  $n^{sr}$  – порядковый номер средства распараллеливания;  $t_i^{vip}$  – массив времен выполнения цикла.

Схема алгоритма методики формирования профилеобразующей базы программы приведена на рис. 2. В соответствии с данной схемой выполняются следующие операции.

1.2. Производится разбор исходного кода программы. В исходном коде ПС МК ВН выявляются циклы, имеющие счетчик числа итераций (циклы со счетчиком), а также не имеющие в теле цикла команд принудительного выхода из цикла. Для разбора конструкции цикла проводится синтаксический анализ исходного кода программы, в ходе которого символы исходного кода группируются в лексемы (распознанные группы символов) в соответствии с набором токенов. Затем проводится синтаксический анализ конструкции цикла. Он выполняется в соответствии с деревом разбора. Изначально в исходном коде программы выделяются лексемы: параметр цикла, тело цикла. Выявление выполняется по последовательности символов, содержащихся в токенах. В сформированных лексемах в ходе лексического анализа идентифицируются входящие в них токены, определяются параметры токенов. В ходе разбора конструкции



Рис. 2. Схема алгоритма методики профилеобразующей базы программы

Fig. 2. Scheme of the algorithm of the methodology of the profile-forming base of the program

цикла выявляются также вложенные циклы по вхождению в состав лексемы тела цикла лексем параметров цикла.

3. Выполняется сквозная нумерация циклических участков программного кода.

4. Выполняется размещение меток начала и окончания циклического участка в коде.

5. Происходит клонирование циклических участков с добавлением прагм алгоритмов распараллеливания.

6,7. Полученный код программы компилируется в машинный код, и производится многократный запуск на тестовый прогон для получения первых, статистических данных по всем циклическим участкам с примененными алгоритмами распараллеливания и вариантам числа итераций. В результате тестовых запусков формируется профилеобразующая база программы, состоящая из набора кортежей  $\{v_p\}_{p=1}^{N_{vp}} = c_z; n_s^{it}; n_k^{sr}; t_i^{vip}$ , в которые входят номер цикла  $c$  количество итераций  $n^{it}$  в данном цикле, номер алгоритма распараллеливания  $n^{sr}$  и время выполнения для каждого из модифицированных циклических участков программы.

8. Формируется и сохраняется профилеобразующая база программы в виде набора кортежей таблицы.

### Методика формирования время-параметризованных нормативных профилей

Входные данные:

1.  $v = \{c_z; n^{it}; n^{sr}; t^{vip}\}$  – профилеобразующая база программы в виде набора кортежей таблицы, где  $c$  – порядковый номер цикла программы;  $n_s^{it}$  – число итераций цикла;  $n_k^{sr}$  – порядковый номер алгоритма распараллеливания;  $t^{vip}$  – время выполнения цикла.

2.  $P^{dov}$  – доверительная вероятность попадания времени выполнения цикла в заданный интервал.

Выходные данные:

$v^* = \{c; n_s^{it}; n^{sr}; t^{sup}\}$  – время-параметризованный нормативный профиль программы, где  $t^{sup}$  – значение времени выполнения цикла по верхней границе заданного доверительного интервала.

Схема алгоритма методики формирования время-параметризованных нормативных профилей приведена на рис. 3.

В соответствии с представленной схемой выполняются следующие операции.

1. Считывается профилеобразующая база программы в виде набора кортежей таблицы  $v = \{v_p\}_{p=1}^{N_{vp}} = c_z; n_s^{it}; n_k^{sr}; t_i^{vip}$  в которые входят номер цикла  $c$  количество итераций  $n^{it}$  в данном цикле, номер алгоритма распараллеливания  $n^{sr}$  и время выполнения для каждого из модифицированных циклических участков программы. Формируется выборка из профиля для одного значения: номера цикла  $c_z$ ; числа итераций  $n_s^{it}$  этого цикла; номера алгоритма распараллеливания  $n_k^{sr}$ . Соответственно, данная выборка содержит массив времен выполнения цикла  $\{t_i^{vip}\}_{i=1}^{N_i}$ .

2. В связи с тем, что функция распределения времени выполнения цикла неизвестна [5], из полученной профилеобразующей базы программы формируются эмпирические функции распределения времени выполнения каждого цикла следующим образом:

2.1. Элементы массива времен выполнения цикла  $\{t_i^{vip}\}_{i=1}^{N_i}$  ранжируются по возрастанию значения времени.

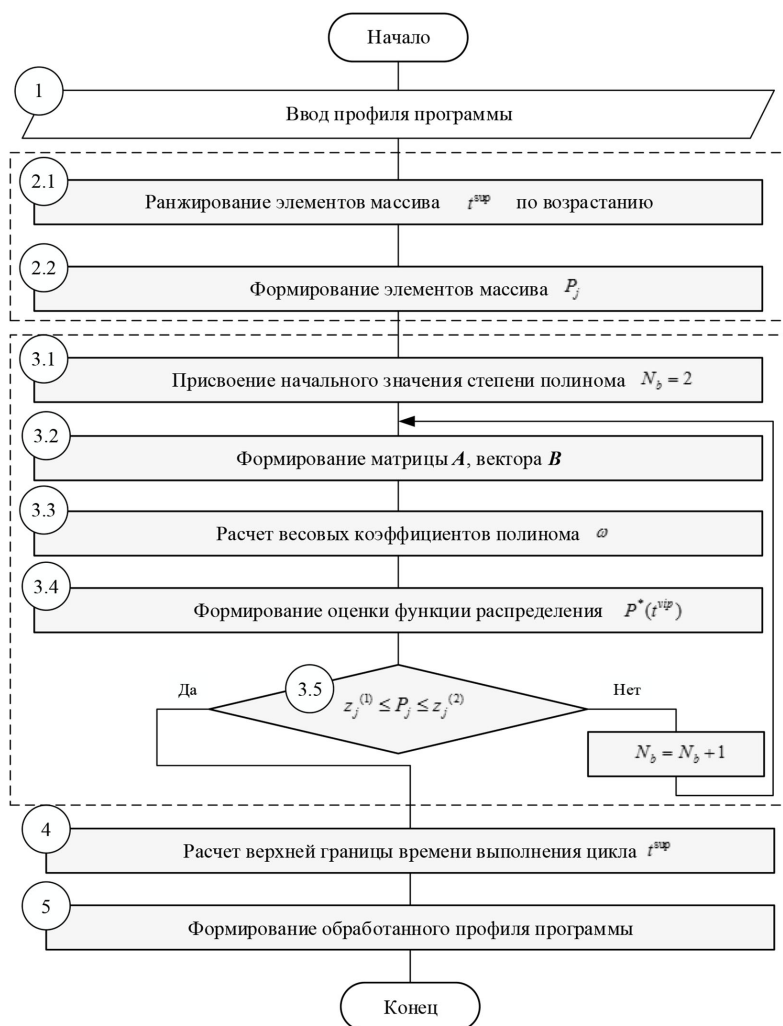


Рис. 3. Схема алгоритма методики формирования время-параметризованных нормативных профилей  
 Fig 3. The scheme of the algorithm of the methodology for the formation of parameterized normative profiles

2.2. Рассчитываются  $j$ -е значения функции распределения для ранжированного массива  $\{t_i^{vip}\}_{i=1}^{N_t}$  [6] по формуле

$$P_j = \frac{1}{N_t} \sum_{i=1}^{N_t} I_{ij} \quad (1)$$

где  $I_{ij}$  – индикаторная функция, принимающая значения:  $I_{ij} = \begin{cases} 1 & \text{при } t_i^{vip} \leq t_j^{vip} \\ 0 & \text{при } t_i^{vip} > t_j^{vip} \end{cases}$

3. Эмпирическая функция распределения аппроксимируется гладкой кривой. При выборе способа аппроксимации учитывается, что вид функции распределения времени выполнения цикла неизвестен. При этом плотность распределения может быть полимодальной, т. е. иметь несколько максимумов (при одном и том же числе итераций цикла, но при разных значениях исходных данных время выполнения цикла может быть различным).

Так как вид функции распределения неизвестен, то необходимо применять непараметрические методы аппроксимации. В [7] показано, что высокая точность аппроксимации непараметрическим методом как унимодальных, так и полимодальных распределений, заданных на конечных интервалах, обеспечивается применением полиномов Бернштейна. При этом оценка функции распределения описывается выражением

$$P^*(t^{pr}) = \sum_{k=1}^{N_b} \omega_k F(t^{pr}, k, N_b - k + 1) \quad (2)$$

где  $N_b$  – степень полинома Бернштейна;  $\omega_k$  – весовые коэффициенты полинома Бернштейна;

$t^{pr}$  – приведенное время выполнения цикла;  $F(t^{pr}, k, N_b - k + 1) = \frac{(N_b + 1)!}{k!(N_b - k + 1)!} \int_0^{t^{pr}} t^{k-1} (1-t)^{N_b-k} dt$  –

функция распределения бета-распределения;  $t^{pr}$  – приведенное время выполнения цикла, рассчитывается по формуле

$$t^{pr} = \frac{t_{\min}^{vip} - t_{\max}^{vip}}{t_{\max}^{vip} - t_{\min}^{vip}}, \quad (3)$$

где  $t_{\max}^{vip}$ ,  $t_{\min}^{vip}$  – соответственно максимальное и минимальное значение времени выполнения

цикла в массиве  $\{t_j^{vip}\}_{j=1}^{N_t}$ . Приведенное значение времени используется ввиду того, что функция

распределения бета-распределения определена на интервале  $[0;1]$ .

Расчет параметров оценочной функции  $P^*(t^{vip})$  выполняется в два этапа:

I) при заданной степени полинома  $N_b$  (начиная с минимального значения  $N_b = 2$ ) определяются значения весовых коэффициентов  $\omega_k$ ;

II) повторяется первый этап для набора значений  $N_b$  и выбирается величина  $N_b$ , при которой обеспечивается требуемая точность аппроксимации.

3.1. Присваивается начальное значение степени полинома Бернштейна  $N_b = 2$ .

3.2. Для определения значений весовых коэффициентов  $\omega_k$  используется метод минимальных квадратов. При этом применяется целевая функция  $W_{mnk}$ , минимизирующая разность квадратов между значениями эмпирической функции распределения  $P$  и оценкой функции распределения  $P(t^{pr})$ :

$$W_{mnk} = \sum_{j=1}^{N_t} (P_j - P_j^*)^2 = \sum_{j=1}^{N_t} \left( P_j - \sum_{k=1}^{N_b} \omega_k F(t_j^{pr}, k, N_b - k + 1) \right)^2 \quad (4)$$

Для нахождения значений  $\omega_k$  вычисляются частные производные  $W_{mnk}$  по  $\omega_k$  ( $k = \overline{1; N_b}$ )

и приравняются к нулю:  $\frac{\partial W_{mnk}}{\partial \omega_k} = 0$ . Частные производные образуют систему из  $N_b$  линейных

уравнений. После дифференцирования целевой функции  $W_{mnk}$  получаем:

$$\sum_{k=1}^{N_b} \omega_k \sum_{j=1}^{N_t} F(t_j^{pr}, k, N_b - k + 1) F(t_j^{pr}, m, N_b - m + 1) - \sum_{j=1}^{N_t} P_j F(t_j^{pr}, m, N_b - m + 1) = 0 \quad (5)$$



Обозначим:

$$A_{m,k} = \sum_{j=1}^{N_i} F(t_j^{pr}, k, N_b - k + 1) \cdot F(t_j^{pr}, m, N_b - m + 1),$$

$$B_m = \sum_{j=1}^{N_i} P_j \cdot F(t_j^{pr}, m, N_b - m + 1),$$

$$F(t_j^{pr}, k, N_b - k + 1) = \frac{(N_b + 1)!}{k!(N_b - k + 1)!} \int_0^{t_j^{pr}} t^{k-1} (1-t)^{N_b-k} dt,$$

$$F(t_j^{pr}, m, N_b - m + 1) = \frac{(N_b + 1)!}{m!(N_b - m + 1)!} \int_0^{t_j^{pr}} t^{m-1} (1-t)^{N_b-m} dt,$$

где  $\overline{k=1; N_b}$  – номера элементов строки вектора  $\omega$  и столбцов матрицы  $A$ ;  $\overline{m=1; N_b}$  – номера элементов строки вектора  $B$  и матрицы  $A$ . Тогда система уравнений записывается в матричном виде

$$A\omega - B = 0. \quad (6)$$

3.3. Рассчитываются весовые коэффициенты полинома  $\omega = \{\omega_k\}_{k=1}^{N_b}$  путем нахождения приближенного решения системы линейных уравнений (6) при ограничениях:  $\omega_k \geq 0$ ,  $\sum_{k=1}^{N_b} \omega_k = 1$ .

3.4. Определяется выражение для оценки функции распределения:

$$P^*(t^{pr}) = \sum_{k=1}^{N_b} \omega_k F(t^{pr}, k, N_b - k + 1), \quad (7)$$

$$F(t^{pr}, k, N_b - k + 1) = \frac{(N_b + 1)!}{k!(N_b - k + 1)!} \int_0^{t^{pr}} t^{k-1} (1-t)^{N_b-k} dt. \quad (8)$$

3.5. Для принятия решения о возможности использования аппроксимирующей функции рассчитывается доверительный интервал для каждого  $j$ -го значения эмпирической функции распределения:

$$z_j^{(1,2)} = \frac{2N_t P^*(t_j^{vip}) + \varepsilon^2}{2(N_t + \varepsilon^2)} \mp \frac{\sqrt{4N_t \varepsilon^2 P^*(t_j^{vip})(1 - P^*(t_j^{vip})) + \varepsilon^4}}{2(N_t + \varepsilon^2)}. \quad (9)$$

где  $\varepsilon$  – квантиль нормального распределения для заданной доверительной вероятности (например, для  $P^{dov} = 0.99$ , значение  $\varepsilon = 2,576$ ).

Аппроксимирующая функция принимается при выполнении для всех  $j$ -х значений эмпирической функции распределения условия:  $z_j^{(1)} \leq P_j \leq z_j^{(2)}$ . В противном случае степень полинома  $N_b$  увеличивается на единицу и выполняется повторный расчет весовых коэффициентов.

4. Расчет значения верхней границы  $t^{sup}$  времени выполнения цикла выполняется путем решения уравнения  $P^*(t^{sup}) = P^{dov}$ . Ввиду того, что отсутствуют аналитические способы решения данного уравнения, поиск  $t^{sup}$  выполняется численным методом – методом дихотомии.

При этом появляется возможность сравнить по данному уже детерминированному параметру  $t^{\text{sup}}$  разные средства распараллеливания цикла и произвести выбор с минимальным временем.

5. Далее с использованием рассчитанного значения  $t^{\text{sup}}$  формируется время-параметризованный нормативный профиль программы  $v^* = \{c; n_s^{it}; n_k^{sr}; t^{\text{sup}}\}$ , который представляет собой массив данных, состоящих из векторов параметров: количество итераций цикла  $n_s^{it}$ , номер алгоритма распараллеливания  $n_k^{sr}$ , верхняя граница времени выполнения  $t^{\text{sup}}$  и номера цикла.

### Методика построения профиля проекта

Рассматривается поиск оптимального средства распараллеливания для одного конкретного цикла программы (одного порядкового номера цикла).

Входные данные:

1. Время-параметризованный нормативный профиль программы  $v^* = \{c; n^{it}; n^{sr}; t^{\text{sup}}\}$ , состоящий из профилей каждого цикла программы  $v^* = \{v_p^*\}_{p=1}^{N_s}$ , где  $c$  – номер цикла;  $n^{it}$  – вектор числа итераций цикла;  $n^{sr}$  – вектор порядковых номеров алгоритмов распараллеливания;  $t^{\text{sup}}$  – вектор значений верхней границы времени выполнения цикла.

2.  $n_z$  – текущее число итераций цикла.

Выходные данные:

1.  $s = N_m : m_{predp} = \{c; m_s^{nach}; n_s^{predp}\}_{s=1}^{N_m}$  – профиль проекта.

2.  $n_{predp}^{sr z}$  – номер предпочтительного алгоритма распараллеливания для текущего числа итераций цикла.

Схема алгоритма методики построения профиля проекта приведена на рис. 5. В соответствии с данной блок-схемой выполняются следующие операции.

1. Считывается обработанный время-параметризованный нормативный профиль программы  $v^* = \{c; n^{it}; n^{sr}; t^{\text{sup}}\}$ .

2. Для исследуемого порядкового номера цикла  $c$  время-параметризованный нормативный профиль перерабатывается и представляется в виде матрицы:

$$v_c^* = \begin{pmatrix} n_1^{it} & n_2^{it} & \dots & n_s^{it} & \dots & n_{N_{it}}^{it} \\ t_{1,1}^{\text{sup}} & t_{1,2}^{\text{sup}} & \dots & t_{1,s}^{\text{sup}} & \dots & t_{1,N_{it}}^{\text{sup}} \\ t_{2,1}^{\text{sup}} & t_{2,2}^{\text{sup}} & \dots & t_{2,s}^{\text{sup}} & \dots & t_{2,N_{it}}^{\text{sup}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{q,1}^{\text{sup}} & t_{q,2}^{\text{sup}} & \dots & t_{q,s}^{\text{sup}} & \dots & t_{q,N_{it}}^{\text{sup}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{N_{sr},1}^{\text{sup}} & t_{N_{sr},2}^{\text{sup}} & \dots & t_{N_{sr},s}^{\text{sup}} & \dots & t_{N_{sr},N_{it}}^{\text{sup}} \end{pmatrix},$$

где  $s = \overline{1; N_{it}}$  – порядковый номер числа итераций цикла;  $q = \overline{1; N_{sr}}$  – порядковый номер алгоритма распараллеливания.

Массив ранжирован по возрастанию значений  $n_s^{it}$ . Первый столбец матрицы состоит из нулевых элементов. Элементами следующих строк являются верхние границы времен выполнения цикла с учетом разных номеров примененных алгоритмов распараллеливания.

3. Выполняется интерполяция и экстраполяция зависимости верхней границы времени выполнения цикла  $t^{\text{sup}}$  от числа итераций  $n_s^{it}$  для всех рассматриваемых алгоритмов распарал-

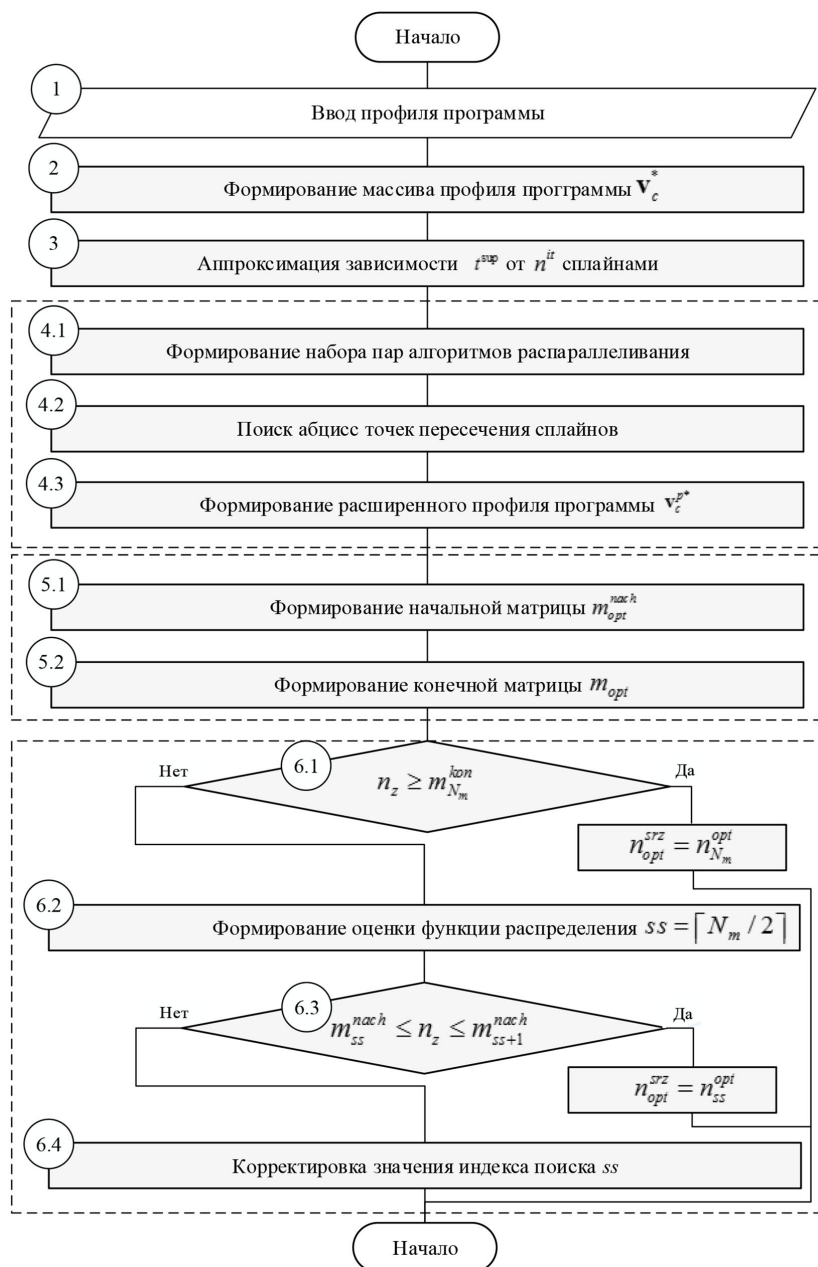
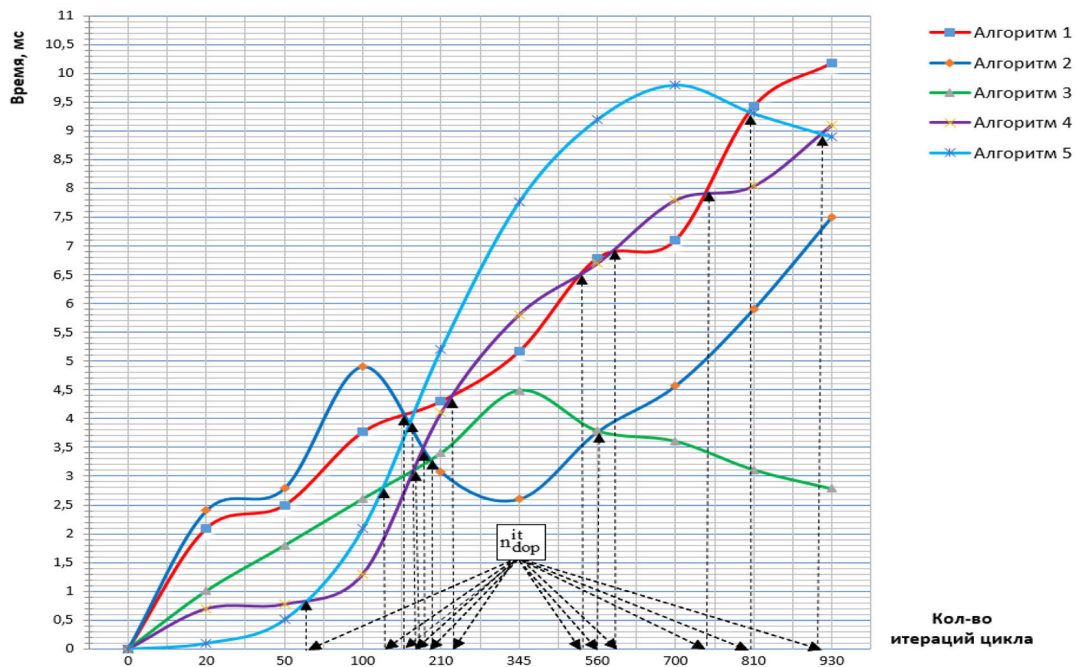


Рис. 4. Схема алгоритма методики построения профиля проекта

Fig. 4. Scheme of the algorithm of the project profile construction methodology

леливания. Интерполяция и экстраполяция время-параметризованного нормативного профиля программы  $v_c^*$  необходимы для расчета величины  $t^{\text{sup}}$  для любых значений  $n_s^{\text{it}}$  (не совпадающих со значениями  $n_s^{\text{it}}$ , содержащимися в профиле). При интерполяции и экстраполяции используется одинаковый подход, основанный на применении кубических сплайнов [8]. При использовании кубических сплайнов на каждом из отрезков  $[n_{s-1}^{\text{it}}, n_s^{\text{it}}]$  определяется аппроксимирующая функция для  $q$ -го алгоритма распараллеливания:

Рис. 5. Аппроксимация зависимости  $t^{\text{sup}}$  от  $n^{\text{it}}$ Fig. 5. Approximation of the dependence  $t^{\text{sup}}$  on  $n^{\text{it}}$ 

$$f_{q,s}(n^{\text{it}}) = a_{q,s} + b_{q,s}(n^{\text{it}} - n_s^{\text{it}}) + c_{q,s}(n^{\text{it}} - n_s^{\text{it}})^2 + d_{q,s}(n^{\text{it}} - n_s^{\text{it}})^3, \quad (10)$$

где  $n^{\text{it}}$  – текущее число итераций цикла;  $a_{q,s}$ ,  $b_{q,s}$ ,  $c_{q,s}$ ,  $d_{q,s}$  – коэффициенты сплайна для  $s$ -го числа итераций цикла  $n_s^{\text{it}}$   $q$ -го алгоритма распараллеливания;  $n_s^{\text{it}}$  –  $s$ -е число итераций цикла. Рассчитанные значения коэффициентов сплайна сохраняются в оперативной памяти для их дальнейшего использования на шаге 4.

В результате нахождения аппроксимирующих функций  $f_{q,s}(n^{\text{it}})$  зависимости  $t^{\text{sup}}$  от  $n^{\text{it}}$  сглаживаются гладкими кривыми. На рис. 5 приведен пример аппроксимации верхней границы времени выполнения цикла  $t^{\text{sup}}$  от числа итераций для пяти алгоритмов распараллеливания. Из него понятно, что могут возникать пересечения аппроксимирующих кривых разных алгоритмов распараллеливания.

4. Формируются дополнительные элементы массива профиля программы. Как видно на рис. 5, могут возникать пересечения аппроксимирующих кривых разных средств распараллеливания. Так как предпочтительный алгоритм выбирается в процессе штатной работы программы, то необходимо минимизировать время выбора. Для этого время-параметризованный нормативный профиль программы  $v^* = \{c; n_s^{\text{it}}; n^{\text{sr}}; t^{\text{sup}}\}$  преобразуется в кортеж предпочтительных алгоритмов распараллеливания  $\mathbf{m}_{\text{predp}} = \{c; \mathbf{m}^{\text{nach}}; \mathbf{n}^{\text{predp}}\}$ , где  $\mathbf{m}^{\text{nach}}$  – начальное число итераций цикла,  $\mathbf{n}^{\text{predp}}$  – номер предпочтительного алгоритма распараллеливания для этого цикла при определенном числе итераций. Выполнение такого преобразования требует нахождения участков  $[m_s^{\text{nach}}; m_{(s+1)}^{\text{nach}}]$ , на которых отсутствуют пересечения аппроксимирующих сплайнов разных алгоритмов распараллеливания. Поэтому в сформированный массив про-

филя программы необходимо добавить столбцы по каждому такому дополнительному значению числа итераций  $n_{dop}^{it}$ , возникающему в местах пересечения сплайнов. Данная операция выполняется путем формирования набора векторов-столбцов дополнительных значений числа итераций цикла. Для этого организуется цикл по числу итераций  $s = \overline{2; N_{it}^p}$ .

4.1. На каждом  $s$ -м шаге цикла для значений числа итераций  $n_{s-1}^{it}, n_s^{it}$  формируются наборы пар алгоритмов распараллеливания из  $q1$ -го и  $q2$ -го алгоритмов ( $q(2) = \overline{1; N_{sr}^p}$ ). Для каждого элемента этой пары из оперативной памяти считываются соответствующие значения коэффициентов сплайна  $a_{q1(2), s}, b_{q1(2), s}, c_{q1(2), s}, d_{q1(2), s}$ , рассчитанные ранее.

4.2. Так как в точках пересечения сплайнов значения аппроксимирующих функций  $f_{q1, s}(n^{it}), f_{q2, s}(n^{it})$  имеют одинаковую величину, то они приравняются друг к другу. В результате этого выполняется поиск действительных корней кубического уравнения относительно переменной  $n^{it}$  (абсцисс точек пересечения):

$$(a_{q1, s} - a_{q2, s}) + (b_{q1, s} - b_{q2, s})x_s + (c_{q1, s} - c_{q2, s})x_s^2 + (d_{q1, s} - d_{q2, s})x_s^3 = 0 \quad (11)$$

4.3. Далее анализируется попадание найденных корней  $n^{it}$  в рассматриваемый интервал  $[n_{s-1}^{it}, n_s^{it}]$  по выполнению условия:  $n_{s-1}^{it} < n^{it} < n_s^{it}$ . При этом для последнего значения  $s = N_{it}$  при проверке предыдущего условия принимается  $n_s^{it} = \infty$ , так как последний отрезок используется для экстраполяции зависимости  $t^{sup}$  от  $n^{it}$ . А в случае выполнения условия найденные действительные корни  $n^{it}$  фиксируются как дополнительное значение числа итераций  $n_{dop}^{it}$ . Для каждого значения  $n_{dop}^{it}$  рассчитываются величины  $t_{q, s}^{sup}$  по каждому  $q$ -му средству распараллеливания. Таким образом, формируются дополнительные столбцы матрицы  $v_c^*$ , которыми дополняется ранее полученная матрица. В результате дополнения формируется расширенная матрица профиля программы  $v_c^{p*}$ .

Столбцы этой матрицы ранжируются в порядке возрастания значений  $n_s^{it}$ .

$$v_c^{p*} = \begin{pmatrix} n_1^{it} & n_{dop1}^{it} & n_2^{it} & n_{dop2}^{it} & \dots & n_s^{it} & \dots & n_{N_{it}}^{it} \\ t_{1.1}^{sup} & t_{1,dop1}^{sup} & t_{1.2}^{sup} & t_{1,dop2}^{sup} & \dots & t_{1,s}^{sup} & \dots & t_{1,N_{it}}^{sup} \\ t_{2.1}^{sup} & t_{2,dop1}^{sup} & t_{2.2}^{sup} & t_{q,dop2}^{sup} & \dots & t_{2,s}^{sup} & \dots & t_{2,N_{it}}^{sup} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{q.1}^{sup} & t_{q,2,dop1}^{sup} & t_{q.2}^{sup} & t_{q,dop2}^{sup} & \dots & t_{q,s}^{sup} & \dots & t_{q,N_{it}}^{sup} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{N_{sr},1}^{sup} & t_{N_{sr},dop1}^{sup} & t_{N_{sr},2}^{sup} & t_{N_{cp},dop2}^{sup} & \dots & t_{N_{sr},s}^{sup} & \dots & t_{N_{sr},N_{it}}^{sup} \end{pmatrix}$$

5. Формируется массив оптимальных средств распараллеливания. Так как предпочтительный алгоритм распараллеливания выбирается в процессе штатной работы программы, то необходимо минимизировать время выбора. Для этого время-параметризованный нормативный профиль программы  $v^* = \{c; n_s^{it}; n^{sr}; t^{sup}\}$  преобразуется в профиль проекта, состоящий из совокупности кортежей предпочтительных алгоритмов распараллеливания. Для этого организуется цикл по расширенному числу итераций  $s = \overline{1; N_{it}^p} - 1$ .

5.1. На каждом  $s$ -м шаге цикла формируется значение числа итераций начала диапазона:

$$m_s^{nch} = n_s^{it} \quad (12)$$

Для интервала  $[m_s^{nach}, m_{s+1}^{nach}]$  определяется номер алгоритма распараллеливания, имеющий минимальное значение верхней границы времени выполнения цикла  $t_{q,s}^{sup}$ :

$$n_s^{predp} = \arg \min_q t_{q,s}^{sup} \quad (13)$$

Формируется начальная матрица предпочтительных алгоритмов распараллеливания:

$$m_{predp}^{nach} = \left\{ c; m_s^{nach}; n_s^{predp} \right\}_{s=1}^{N_m^p - 1} \quad (14)$$

5.2. Ввиду того, что некоторые смежные интервалы могут иметь одинаковые оптимальные средства распараллеливания, выполняется объединение таких смежных интервалов. В результате формируется конечная матрица оптимальных средств распараллеливания с числом элементов  $s = N_m$ :

$$m_{predp} = \left\{ c; m_s^{nach}; n_s^{predp} \right\}_{s=1}^{N_m} \quad (15)$$

Полученная матрица  $m_{predp}$  сохраняется в базе данных. При выполнении программы матрица  $m_{predp}$  загружается в оперативную память для ускорения доступа к данным.

В результате после объединения смежных интервалов с одинаковыми предпочтительными алгоритмами распараллеливания формируется профиль проекта с числом элементов  $s = N_m$ :  $m_{predp} = \{c; m_s^{nach}; n_s^{predp}\}_{s=1}^{N_m}$ . Полученный профиль  $m_{predp}$  сохраняется в базе данных. Конечный профиль проекта для указанного диапазона итераций цикла, который представляет собой совокупность кортежей, состоящих из трех параметров: номер цикла  $c_z$ , начальное количество итераций цикла  $m_z^{nach}$ , номер предпочтительного алгоритма распараллеливания  $n_s^{predp}$ . Ввиду того, что некоторые смежные интервалы могут иметь одинаковые предпочтительные алгоритмы распараллеливания, выполняется объединение таких смежных интервалов.

6. Для заданного текущего числа итераций цикла  $n_z$  определяется номер предпочтительного алгоритма распараллеливания. В ходе штатного выполнения программы определяется текущее число итераций цикла  $n_z$ . Из базы данных считывается профиль проекта  $m_{predp}$ . Далее методом дихотомии осуществляется поиск предпочтительного алгоритма распараллеливания  $n_s^{predp}$ . Применение метода дихотомии позволяет ускорить поиск предпочтительного алгоритма. Это особенно важно ввиду того, что поиск выполняется в ходе штатной работы программы в режиме реального времени. Такой подход позволяет найти оптимальное решение не более чем за  $\log_2 N_m$  шагов.

6.1. Проверяется выход значения  $n_z$  за пределы интервалов, имеющих в профиле проекта по выполнению условия

$$n_z \geq m_{N_m}^{kon} \quad (16)$$

Если условие выполняется, то присваивается  $n_{predp}^{srz} = n_{N_m}^{predp}$  и поиск заканчивается.

6.2. Присваивается начальное значение индекса  $m_{predp}$  для выполнения поиска  $ss = \lceil N_m / 2 \rceil$  ( $\lceil \cdot \rceil$  – операция нахождения большей целой части числа).

6.3. При выполнении условия:

$$m_{ss}^{nach} \leq n_z \leq m_{ss+1}^{nach} \quad (17)$$

присваивается  $n_{predp}^{srz} = n_{ss}^{predp}$  и поиск заканчивается.

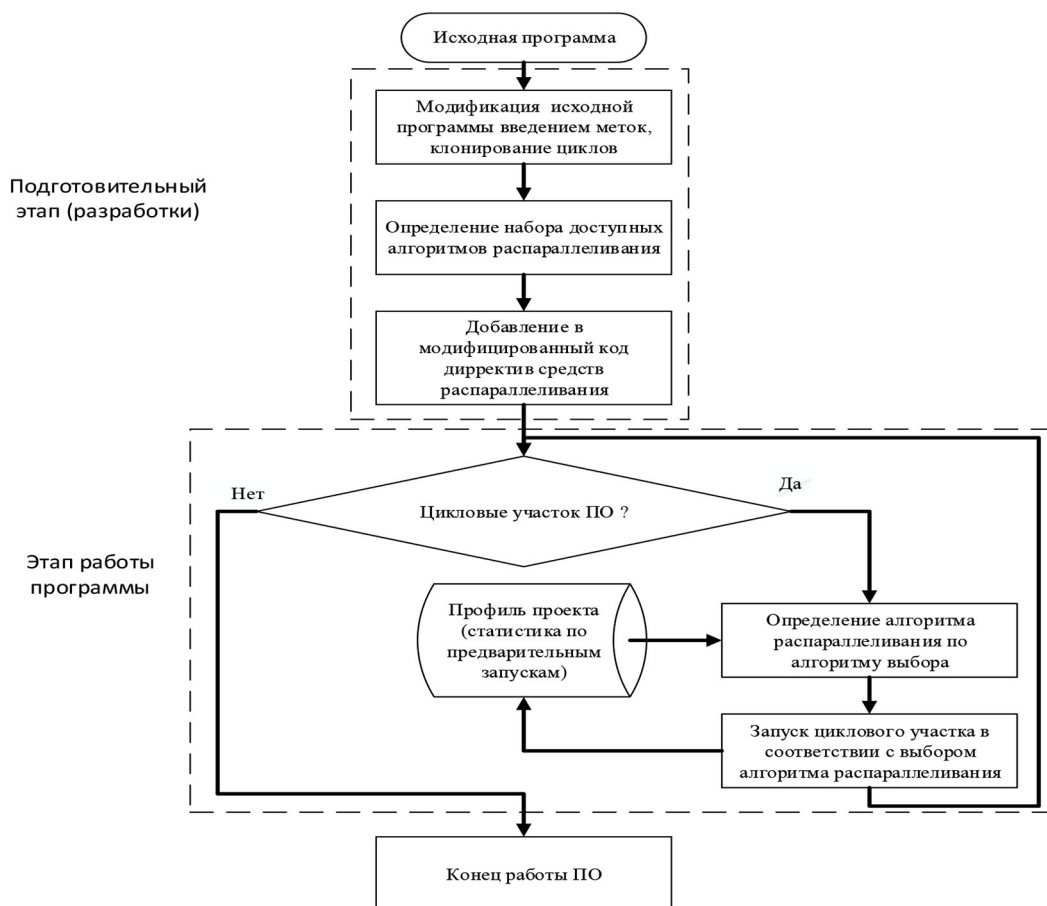


Рис. 6. Схема реализации предлагаемого направления

Fig. 6. Scheme of implementation of the proposed direction

6.4. Корректируется значение индекса поиска  $ss$ . При  $n_z < m_{ss}^{max}$  присваивается  $ss = \lceil ss/2 \rceil$ , иначе  $ss = \lceil (ss + N_m)/2 \rceil$ . Выполняется переход к шагу 6.3.

### Практическая реализация предлагаемого направления

На рис. 6 представлена схема возможной реализации предлагаемого направления. Тело программного кода увеличивается на определенное количество клонов циклов, равное количеству алгоритмов распараллеливания, реализованных в средствах распараллеливания, но при этом продолжительность исполнения программы остается прежней, так как в процессе исполнения все клоны не выполняются, выполняется только выбранный. Процесс управления выбором клона (с примененным алгоритмом распараллеливания) позволяет снизить временные затраты выполнения циклических участков программы и в целом всей программы. Это связано с тем, что временные затраты на выполнение одного и того же циклического участка с заданным количеством итераций при применении разных средств распараллеливания будут существенно отличаться. Поэтому оптимизация выбора ветки (клона) программы позволяет снизить временные затраты на выполнение цикла и программы в целом. Работа автоматизирована,

по входным параметрам циклического участка целевой программы (номера цикла и количества итераций в цикле) происходит выбор предпочтительного алгоритма распараллеливания.

Реализация предполагает два этапа. Первый этап, подготовительный, необходим для загрузки кода целевой программы, исходных данных и библиотек алгоритмов распараллеливания, а также выполнения тестовых запусков целевой программы и наполнения начальной статистики. Вторым этапом – это непосредственно работа с автоматическим выбором предпочтительного алгоритма распараллеливания под каждый циклический участок целевой программы.

### Заключение

Апробация предлагаемого направления и имитационные эксперименты были проведены на базе Военной академии воздушно-космической обороны (г. Тверь) с использованием многофункционального МК ВН «Небосвод 2.0», позволяющего моделировать боевые действия противоборствующих группировок в воздушно-космической сфере и оценивать их результаты в едином интерактивном пространственно-временном контуре управления. Программные средства комплекса разработаны в программно-технической среде *Microsoft Visual Studio 2015* (язык программирования C++, СУБД *PostgreSQL*).

Таким образом, представленное направление развития параллельного программирования, применяемое при разработке ПС МК ВН, позволяет комбинировать алгоритмы распараллеливания и при необходимости комплексно применять их для каждого распараллеливаемого цикла, что дает возможность сократить время выполнения распараллеленного ПС МК ВН, т. е. позволяет сократить время имитационного моделирования отражения удара противника с использованием программных средств моделирования автоматизированных систем военного назначения. Оно заключается в выборе предпочтительного алгоритма распараллеливания под каждый циклический участок программы с учетом времени их выполнения, что поможет минимизировать время выполнения ПС МК ВН и, как следствие, сократить время моделирования. Также установлена функциональная связь между количеством итераций текущего цикла и предпочтительным алгоритмом распараллеливания. Автоматический выбор предпочтительного алгоритма распараллеливания производится на основе классификационного отбора по входному параметру (количества итераций) цикла в сформированном профиле проекта, состоящем из совокупности кортежей предпочтительных алгоритмов распараллеливания. Данное направление может быть взято за основу и использовано при разработке ПС МК ВН, применяемых при моделировании боевых действий в воздушно-космической сфере.

### Список литературы / References

[1] Лясковский В.Л., Ризаев Р.Н., Допира Р.В., Кучеров Ю.С. Методика оценки степени влияния модели системы управления на эффективность применения частей и подразделений радиоэлектронной борьбы в моделирующих комплексах военного назначения, *Вопросы радиоэлектроники*, 2020, 10, 46–52. [Lyaskovskiy V.L., Rizaev R.N., Dopira R.V., Kucherov Yu. S. Methodology for assessing the influence of control system onefficiency of application of electronic warfare units in military modeling complexes. *Issues of radio electronics*, 2020, 10, 46–52 (in Russian)]



[2] Волконский В., Ермолицкий А., Ровинский Е. Развитие метода векторизации циклов при помощи оптимизирующего компилятора. *Высокопроизводительные вычислительные системы и микропроцессоры*, 2005, 8, 34–56. [Volkonsky V., Ermolitsky A., Rovinsky E. Development of the cycle vectorization method using an optimizing compiler. *High-performance computing systems and microprocessors*, 2005, 8, 34–56. (in Russian)]

[3] Биллиг В. А. *Параллельные вычисления и многопоточное программирование*. М.: НОУ «ИНТУИТ», 2016, 311 с. [Billig V. A. *Parallel computing and multithreaded programming*. Moscow, NOU «INTUIT», 2016, 311 p. (in Russian)]

[4] Андреев А. Е., Егунов В. А., Шаповалов О. В. *Технологии программирования многопроцессорных систем*. ВолгГТУ. Волгоград, 2015, 103 с. Andreev A. E., Egunov V. A., Sharovalov O. V. *Programming technologies of multiprocessor systems: textbook. manual*. Volgograd, VolgSTU, 2015, 103 p. (in Russian)]

[5] Вентцель Е. С. *Теория вероятностей*. М.: ЮСТИЦИЯ, 2018, 658 с. [Ventsel E. S. *Teoriya veroyatnostey [Probability theory]*. Moscow, Yustitsiya Publ, 2018, 658 p. (in Russian)]

[6] Выгодский М. Я. *Справочник по высшей математике*. М.: АСТ, 2019, 703 с. [Vygodsky M. Y. *Handbook of Higher Mathematics*. Moscow, AST Publishing House, 2019, 703 p. (in Russian)]

[7] Голик Ф. В. Аппроксимация эмпирических распределений вероятностей полиномами Бернштейна. *Журнал радиоэлектроники*, 2018, 7, 1–8. [Golik F. V. Approximation of empirical probability distributions by Bernstein polynomials. *Journal of Radio Electronics*, 2018, 7, 1–8 (in Russian)]

[8] Волков Е. А. *Приближение функций многочленами*. М.: Наука, 1987, 248 с. [Volkov E. A. *Approximation of functions by polynomials*. Moscow, Nauka, 1987, 248 p. (in Russian)]

[9] Богданов А. В., Горбунов С. А., Кучин А. А., Хадур А. А. Алгоритм наведения управляемой ракеты класса «воздух-воздух» с активной радиолокационной головкой самонаведения на вертолет при различном характере его полета. *Журнал Сибирского федерального университета. Техника и технологии*, 2020, 13(7), 829–842 [Bogdanov A. V., Gorbunov S. A., Kuchin A. A., Hadur A. A. The Guidance Algorithm of Air-to-Air Guided Missile with an Active Radar Target-Seeking Head to a Helicopter with Various Modes of its Flight, *Journal of Siberian Federal University. Engineering and technology*, 2020, 13(7), 829–842 (in Russian)]

[10] Анциферов А. А., Богданов А. В., Бондарев В. Н., Закомолдин Д. В., Иванов И. М., Кучин А. А., Беляев В. М., Голубенко В. А., Дедов М. В., Дрогалин В. В., Иванов Е. С., Шестаков Н. А., Лютиков И. В., Тяпкин В. Н. Назначение, состав, конструктивные особенности и характеристики наземной аппаратуры регистрации радиолокационных сигналов в сантиметровом диапазоне волн. *Журнал Сибирского федерального университета. Техника и технологии*, 2018, 11(3), 318–324 [Antsiferov A. A., Bogdanov A. V., Bondarev V. N., Zakomoldin D. V., Ivanov I. M., Kuchin A. A., Belyaev V. M., Golubenko V. A., Dedov M. V., Drogalin V. V., Ivanov E. S., Shestakov N. A., Lyutikov I. V., Tyapkin V. N. The Appointment, Composition, Design Features and Characteristics of the Ground Equipment for Registration Radar Signals in the Centimeter Wavelength Range, *Journal of Siberian Federal University. Engineering and technology*, 2018, 11(3), 318–324 (in Russian)]

[11] Богданов А. В., Кучин А. А., Федотов А. Ю., Филонов А. А., Скрынников А. А. Алгоритм совместного оценивания функционально связанных координат и состояния

групповой воздушной цели на основе линейной модели с марковской структурой. *Журнал Сибирского федерального университета. Техника и технологии*, 2019, 12(1), 41–63 [Bogdanov A. V., Kuchin A. A., Fedotov A. Yu., Filonov A. A., Skrynnikov A. A. Algorithm of Simultaneous Assessment of Functionally Related Aircraft Principal Axes and Clustered Air Target State Based on Linear Model with Markovian Structure, *Journal of Siberian Federal University. Engineering and technology*, 2019, 12(1), 41–63 (in Russian)]

[12] Богданов А. В., Закомолдин Д. В., Докучаев Я. С., Новичёнок В. А., Кочетов, И. В. Алгоритм идентификации уводящих по дальности и скорости помех в импульсно-доплеровских РЛС в интересах формирования достоверных оценок дальности и скорости при сопровождении воздушной цели. *Журнал Сибирского федерального университета. Техника и технологии*, 2019, 12(1), 30–40 [Bogdanov A. V., Zakomoldin D. V., Dokuchaev Y. S., Novichenok V. A., Kochetov I. V. Algorithm of Your Identification Quantities on the Distance and Speed of Interference in the Doppler Radar in the Interests of Forming the Reliable Estimates of the Distance and Speed at the Adaptation of the Air Objective, *Journal of Siberian Federal University. Engineering and technology*, 2019, 12(1), 30–40 (in Russian)]

[13] Замараев В. В., Кутузов А. С., Лютиков И. В., Мальцев Д. В. Метод построения траектории скрытного выхода летательного аппарата в заданную точку зоны обнаружения обзорного радиолокатора, основанный на анализе спектра доплеровских частот принимаемого сигнала. *Журнал Сибирского федерального университета. Техника и технологии*, 2021, 14(3), 281–291 [Zamaraev V. V., Kutuzov A. S., Lyutikov I. V., Malcev D. V. Method of Constructing a Line Secretive Exit the Aircraft at a Given Point of the Detection Area Surveillance Radar Based on the Spectrum Analysis of the Doppler Frequency of the Received Signal, *Journal of Siberian Federal University. Engineering and technology*, 2021, 14(3), 281–291 (in Russian)]

[14] Богданов А. В., Филонов А. А., Ковалев А. А., Кучин А. А. и др. *Методы самонаведения истребителей и ракет класса «воздух-воздух» на групповую воздушную цель: монография*. Красноярск: Сиб. федер. ун-т, 2014, 168 с. [Bogdanov A. V., Filonov A. A., Kovalev A. A., Kuchin A. A. *Homing methods of fighters and air-to-air missiles to a group air target: monograph*. Krasnoyarsk: Siberian Federal University, 2014, 168 p. (in Russian)]

[15] Анциферов А. А., Богданов А. В., Бондарев В. Н., Гарин Е. Н., Гончаров А. М., Кучин А. А., Лютиков И. В. Синтез оптимальных алгоритмов наведения летательных аппаратов на групповую воздушную цель, *Журнал Сибирского федерального университета. Техника и технологии*, 2017, 10(2), 169–182 [Antsiferov A. A., Bogdanov A. V., Bondarev V. N., Garin E. N., Goncharov A. M., Kuchin A. A., Lyutikov I. V. Synthesis of optimal algorithms for guiding aircraft to a group air target. *Journal of Siberian Federal University. Engineering and technology*, 2017, 10(2), 169–182 (in Russian)]

[16] Богданов А. В., Кучин А. А., Мещеряков В. Г., Филонов А. А., Лютиков И. В. *Методы наведения летательных аппаратов на групповую воздушную цель в интересах построения радиоэлектронных систем управления с элементами поддержки решений летчика*. М.: Радиотехника, 2013, 6, 65–69 [Bogdanov A. V., Kuchin A. A., Meshcheryakov V. G., Filonov A. A., Lyutikov I. V. *Methods for guiding aircraft to a group air target in the interests of building electronic control systems with elements of support for the pilot's decisions*. Moscow: Radiotekhnika, 2013, 6, 65–69 (in Russian)]

[17] Богданов А. В., Горбунов С. А., Кучин А. А., Шпортко С. А. Точностные и вероятностные характеристики алгоритма распознавания режима полета вертолета в радиолокационной системе воздушного базирования. *Журнал Сибирского федерального университета. Техника и технологии*, 2018, 11(3), 358–370 [Bogdanov A. V., Gorbunov S. A., Kuchin A. A., Shportko S. A. Accuracy and probabilistic characteristics of the helicopter flight mode recognition algorithm in an airborne radar system, *Journal of Siberian Federal University. Engineering and technology*, 2018, 11(3), 358–370 (in Russian)]

[18] Богданов А. В., Коротков С. С., Кучин А. А., Бондарев В. Н., Лютиков И. В. Концепция распознавания воздушных целей в авиационном радиолокационном комплексе. *Журнал Сибирского федерального университета. Техника и технологии*, 2016, 9(3), 319–331 [Bogdanov A. V., Korotkov S. S., Kuchin A. A., Bondarev V. N., Lyтиков I. V. The concept of recognition of air targets in the aviation radar complex, *Journal of Siberian Federal University. Engineering and technology*, 2016, 9(3), 319–331 (in Russian)]

[19] Жиронкин С. Б., Близнюк А. А., Кучин А. А. Помехоустойчивость запросного канала опознавания с широкополосными сигналами и корректирующими кодами в условиях импульсных шумовых и внутрисистемных помех. *Журнал Сибирского федерального университета. Техника и технологии*, 2019, 12(6), 673–682 [Zhironkin S. B., Bliznyuk A. A., Kuchin A. A. Jamming Resistance of the Inbound Channel of an Identification System with Broadband Signals and Error Control Codes in the Conditions of Pulse Noise and Intra-System Jamming, *Journal of Siberian Federal University. Engineering and technology*, 2019, 12(6), 673–682 (in Russian)]

[20] Анциферов А. А., Бедрицкий А. И., Богданов А. В., Кучин А. А., Лютиков И. В., Муравьева А. С., Филонов А. А. Пути повышения эффективности поражения групповых воздушных целей в противовоздушном бою. *Журнал Сибирского федерального университета. Техника и технологии*, 2021, 14(7), 763–775 [Antsiferov A. A., Bedritsky A. I., Bogdanov A. V., Kuchin A. A., Lyutikov I. V., Muravyeva A. S., Filonov A. A. Ways to improve the effectiveness of defeating group air targets in anti-aircraft combat, *Journal of Siberian Federal University. Engineering and technology*, 2021, 14(7), 763–775 (in Russian)]