

Взаимодействие параллельных алгоритмов

В.С. Поляков, С.В. Поляков, О.А. Авдеюк, В.Ю. Наумов,

Е.С. Павлова, Ю.П. Муха

Волгоградский государственный технический университет

Аннотация: В статье рассматривается актуальная задача построения алгоритма, основанного на взаимодействии параллельно работающих компонентов. В работе приводятся элементы разработанной авторами методики параллельного функционирования алгоритмов на базе теории графов. Показано, что построение алгоритма с параллельным функционированием компонентов невозможно без учёта времени, для чего было введено понятие временной оси, а последовательный алгоритм представлен в виде двудольного графа путём его доопределения. Показанные в статье методы представления параллельных алгоритмов, реализованные в виде дуальных графов или матрично-предикатном виде, дают возможность использовать их при описании сложных управляющих и измерительных систем. Сделан вывод, что преимущество подобного представления параллельного функционирования алгоритмов заключается в том, что становится возможным работать с параллельными иерархическими структурами, отслеживая изменения в каждой из них при помощи вводимых временных осей.

Ключевые слова: алгоритм, параллельная обработка данных, алгоритмизация параллельной работы объектов, последовательные алгоритмы, предикативный блок, временная ось, дуальный граф, взаимодействие параллельных алгоритмов, граф-схема алгоритма, функционально-предикатный модуль.

Сложность построения алгоритмов процессов, осуществляемых многокомпонентными, многокритериальными объектами, заключается в параллельном функционировании компонентов. При использовании последовательных алгоритмов основная проблема состоит в том, что при работе с ними необходимо отслеживать множество процессов, которые в реальном времени происходят параллельно. В связи с этим такие алгоритмы имеют свои особенности: движение алгоритмизируемой системы во времени определяется временной координатой; обеспечение синхронной работы объектов, из которых состоит рассматриваемая система, определяется структурой алгоритма.

В настоящий момент известны четыре основных принципа регламентации событий, среди которых особо стоит выделить Принцип параллельной работы объектов (объектный принцип). Объектный принцип –

это по своей сути алгоритмизация параллельной работы объектов, составляющих рассматриваемую систему, и, естественно, без учёта времени описать такую систему невозможно.

Взаимодействие параллельных алгоритмов является одним из самых сложных процессов при описании параллельного функционирования технических объектов. Часто за параллельный алгоритм принимается такой, «который может быть реализован по частям на множестве различных вычислительных устройств с последующим объединением полученных результатов и получением корректного результата» [1-5,10,11]. В этом случае распараллеливается вычислительный процесс. В данной работе задача ставится таким образом, чтобы было сразу видно, когда и в какой момент времени выполняются различные части общего алгоритма, как они взаимодействуют между собой.

Рассмотрение параллельного функционирования алгоритмов невозможно без учёта времени, для чего вводим понятие временной оси, как направленного отрезка, на котором отмечаются моменты времени или моменты событий выполнения операций алгоритма. В отношении рассматриваемых многокомпонентных, многокритериальных объектов, содержащих параллельно функционирующие компоненты временная ось может быть: реальной, абстрактной или комбинированной.

Реальная временная ось – ось, измерения в которой ведутся непосредственно по времени. То есть каждому действию или операции соответствует какой-то момент времени, непосредственно в который операция имеет начало. Производится корректировка либо самой процедуры, либо времени, отведенном на ее выполнение. Ось удобна, когда дело ведется с системой, в которой время играет значимую роль.

Абстрактная временная ось – ось, измерения на которой ведутся событийно. То есть каждой процедуре предшествует событие, откликом на

которое является запуск процедуры. Событием может быть как внешнее, так и внутреннее воздействия. Абстрактная шкала удобна в применении там, где время не играет существенной роли и на выполнение операции можно отвести большое количество времени.

Комбинированная временная ось – ось, на которой в соответствии с требованиями отсчет ведется как по времени, так и по событиям. Совмещение является несомненным плюсом временной оси, т.к. существуют системы, которые по времени ограничены только внешне, а выполняться должны по командам. Также имеет место совмещения нескольких осей на одной, являющейся главной.

Представление последовательностного алгоритма в виде двудольного графа путём его доопределения. Рассмотрим произвольную ГСА (рис. 1).

Здесь: $A = \{ A_0, A_1, A_2, A_3, A_4, A_k \}$ – вершины, определяющие выполнение отдельных операций, будем называть функциональными блоками;

$\mu = \{ \mu_1^1, \mu_2^1, \mu_3^1, \mu_1^2, \mu_2^2, \mu_3^2, \mu_1^3, \mu_1^4 \}$ – вершины, определяющие логику (порядок) выполнения алгоритма, будем называть функциональными или логическими блоками.

Работа с алгоритмами, заданными в виде (рис. 2), осложняется следующими недостатками:

- переход от выполнения одной операции к выполнению другой в некоторых случаях ничем не обозначен, например, вершины $A_0 - A_1$, здесь подразумевается, что после выполнения операции A_0 следует переход к выполнению операции A_1 , хотя такой переход ничем не фиксируется;

- условия, определяющие порядок выполнения алгоритма, часто задаются несколькими логическими функциями, что усложняет рассмотрение алгоритма.

Рассмотрим связанную пару из функционального – Д (рис. 2а) и предикативного – Л (рис. 2б) блоков:

Д – блок имеет один выход и может иметь несколько входов;

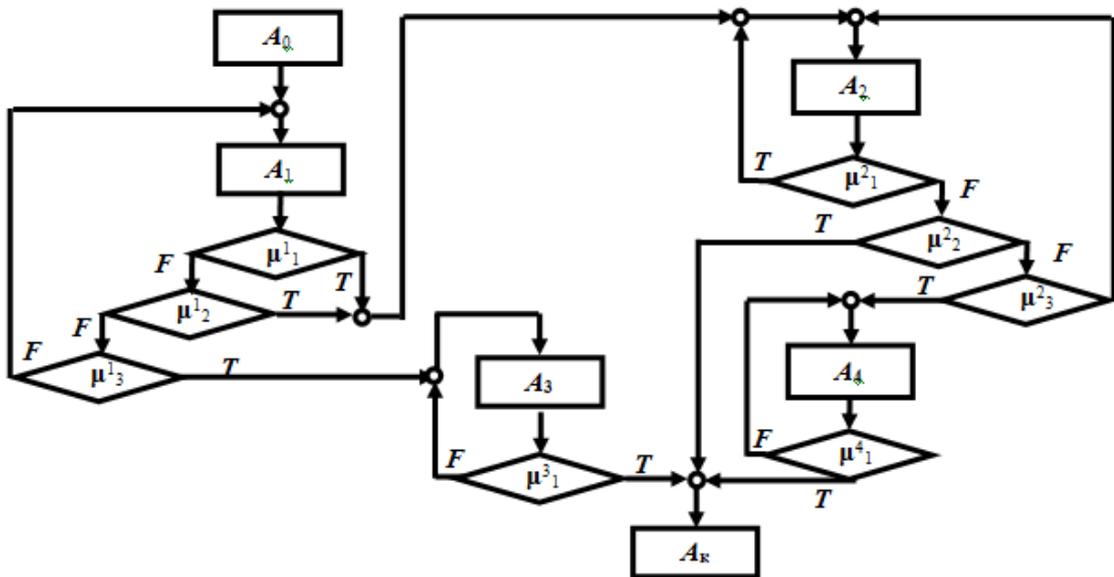


Рис. 1. Граф-схема алгоритма А

Л – блок имеет один вход и может иметь несколько выходов.

Связанную пару (рис. 2в) будем называть функционально-предикатным модулем – М (рис. 2г).

Для построения алгоритма в модульном виде проведём ряд предварительных операций. Зафиксируем окончание выполнения каждого функционального блока соответствующим предикативным блоком.

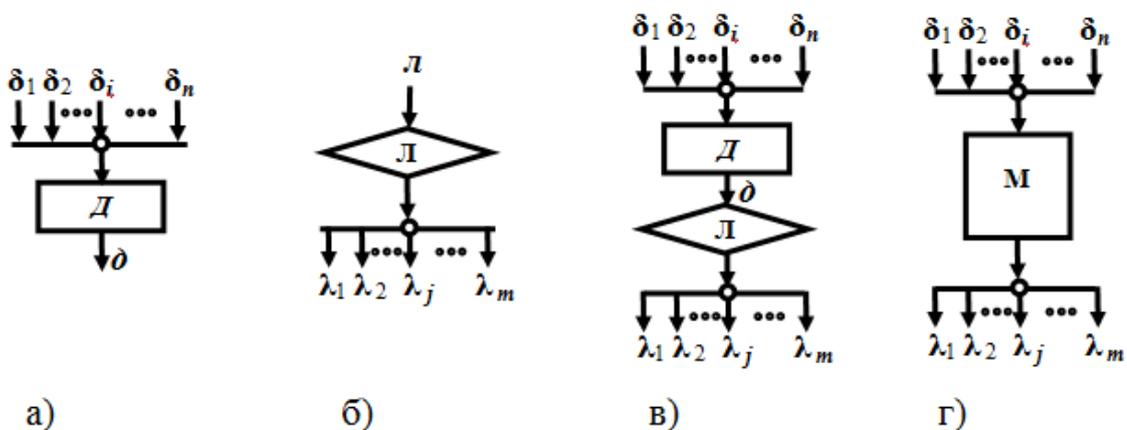


Рис. 2. Блоки алгоритмов: а) функциональный; б) предикативный; в) связанная пара; г) функционально-предикатный модуль

Если переход от выполнения одной операции к выполнению другой ведётся напрямую, иначе говоря, если между функциональными блоками отсутствует предикативный блок, то, введя предикативную вершину α_i^0 , можно представить модуль как на рис 3а. А если после функционального блока существует один или несколько предикативных блоков, но отсутствует фиксация окончания работы функционального блока, то необходимо ввести дополнительно предикативную вершину α_i^0 . Объединим все предикативные вершины в один многозначный предикативный блок и получим модуль как на рис. 3б.

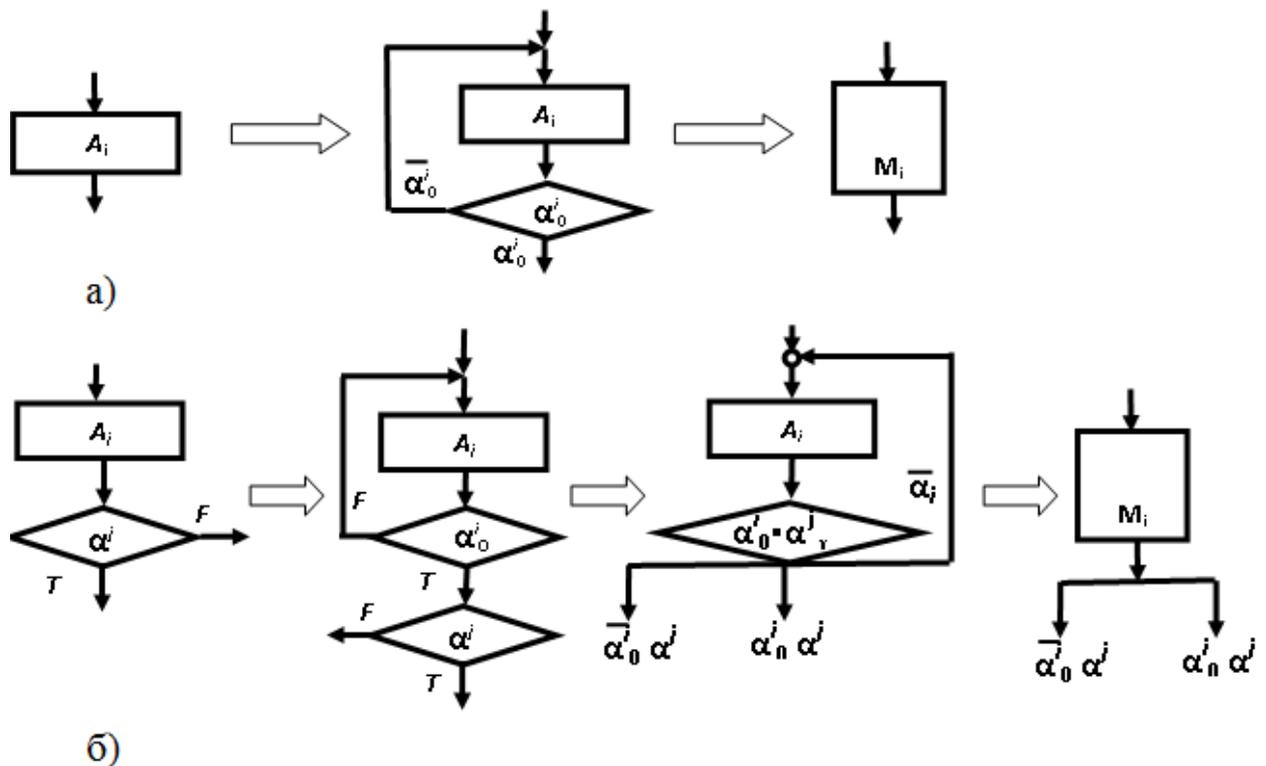


Рис.3. Предикативный блок: а) представление модуля; б) получение модуля

Используя вышеприведённые принципы, «доопределим» функциональные блоки алгоритма, представленного на рис. 1. В результате

получим алгоритм (рис. 4а), дуальный граф которого представлен на рисунке 4б, а на рисунке 4в показано выполнение на оси времени t .

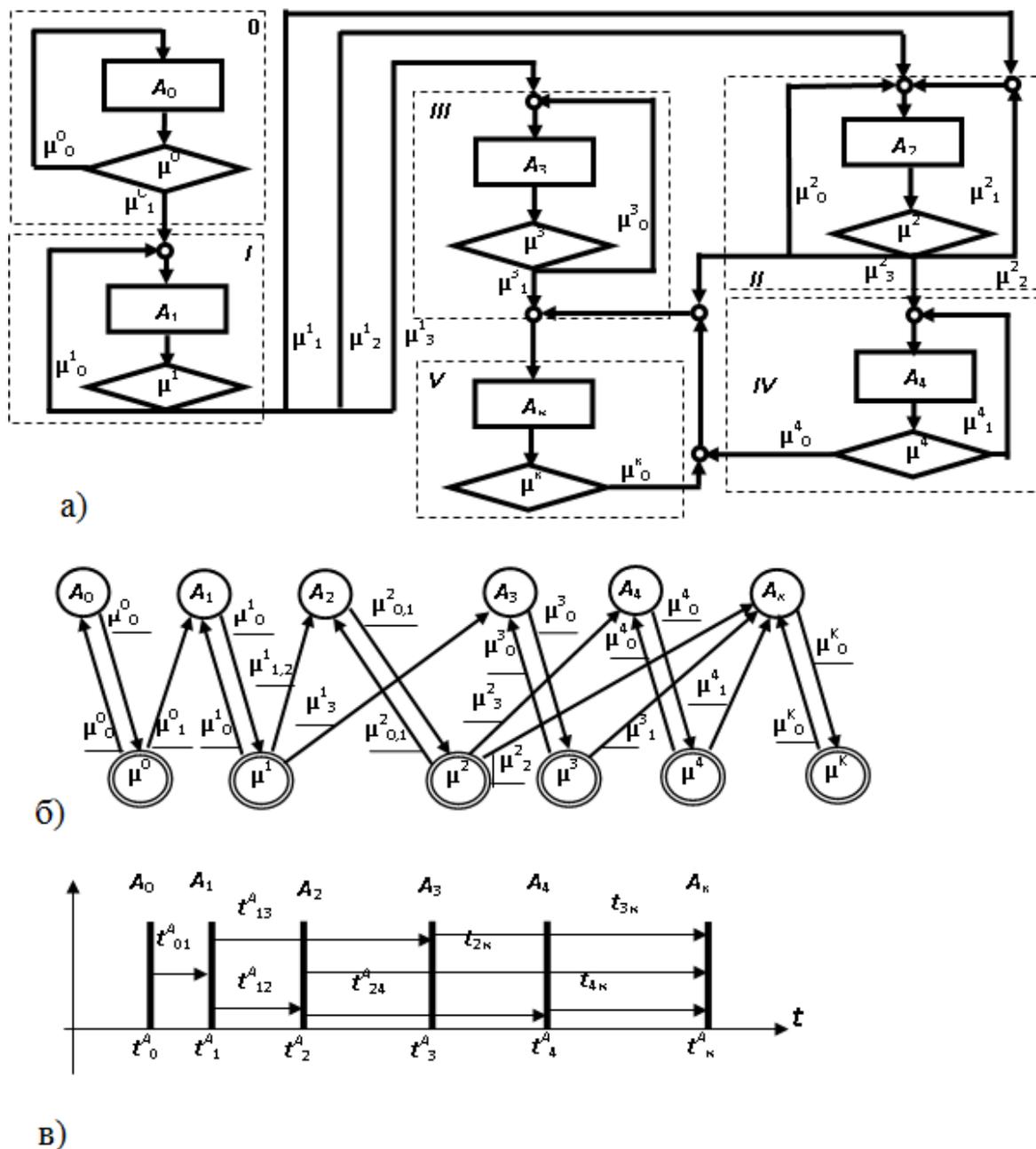


Рис. 4. Нормализованный алгоритм А: а) графическое представление; б) дуальный граф; в) выполнение во времени

Здесь:

A_0, A_1, \dots, A_k – операторы действия алгоритма А;

$t_{A_0}, t_{A_1}, \dots, t_{A_k}$ – время начала оператора действия алгоритма А;

$tA01, tA12, \dots, tA4k$ – время функционирования оператора действия алгоритма А, причём $tA01 = tA1 - tA0, tA12 = tA2 - tA1$ и т.д.

Отметим также, что матрично-предикатное представление возможно двумя способами: функционально-предикативным и модульным [6-9].

На рис. 5 рассмотрен один из подходов представления параллельно функционирующих алгоритмов, для чего был задан второй алгоритм В (рис.5а), его дуальный граф (рис. 5б), а также его выполнение во времени (рис. 5в).

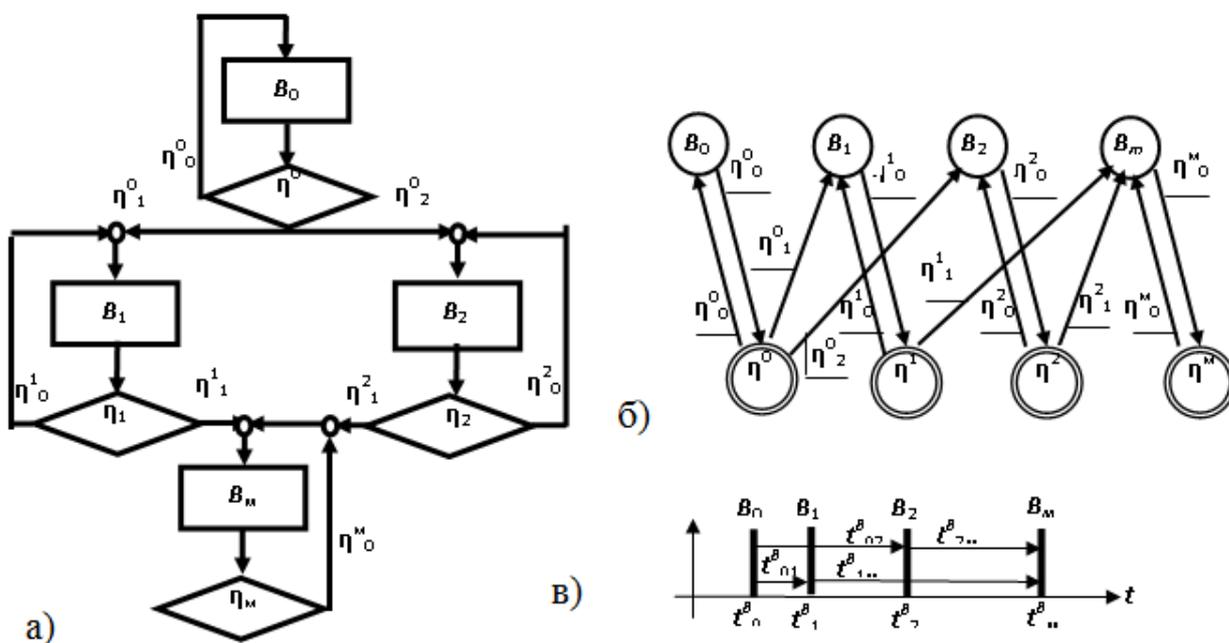


Рис.5. Алгоритм В: а) графическое представление; б) дуальный граф; в) выполнение во времени.

Матрично-предикатное представление алгоритма В можно представить, как и в предыдущем случае, двумя способами: функционально-предикативным и модульным. Возможные различные варианты параллельного взаимодействия, представлены на рисунке 6. Наиболее простым с алгоритмической точки зрения является параллельное взаимодействие двух равнозначных алгоритмов А и В. Стоит также заметить, что при параллельном взаимодействии алгоритмов важно определить какой алгоритм является основным, а какой – вспомогательным. Кроме того

необходимо чётко указать точки начала и окончания параллельного взаимодействия алгоритмов.

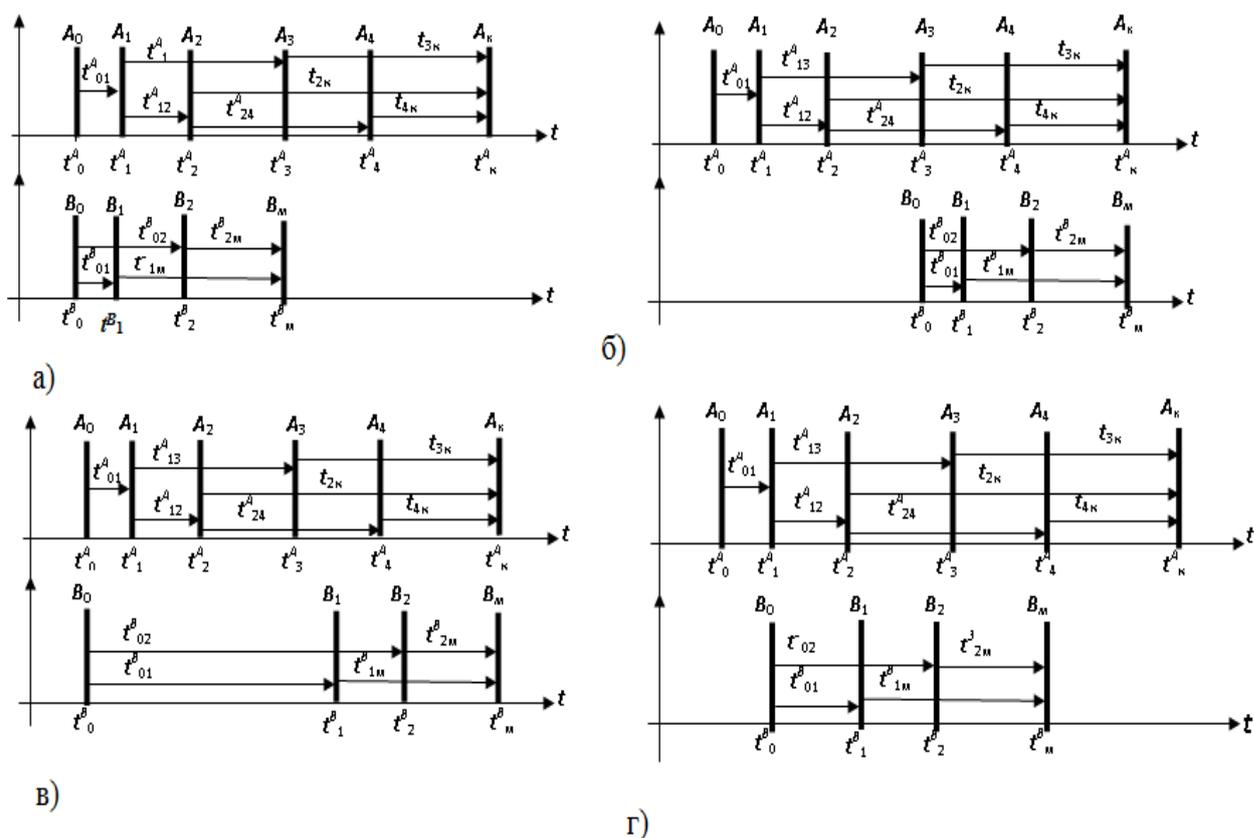


Рис. 6. Параллельное выполнение алгоритмов А и В во времени; а) начала выполнения алгоритмов А и В совпадают; б) окончания выполнения алгоритмов А и В совпадают; в) начало и окончание выполнения алгоритмов А и В совпадают; г) начало и окончание выполнения алгоритма В находится в зоне выполнения алгоритма А, а выполнение операций алгоритма В (кроме начала и конца) происходит от момента начала операции А2 до момента окончания операции А4 алгоритма А.

Таким образом, показанные в статье методы представления параллельных алгоритмов, реализованные в виде дуальных графов или матрично-предикатном виде, дают возможность использовать их при

описании сложных управляющих и измерительных систем. Преимущество подобного представления параллельного функционирования алгоритмов заключается в том, что становится возможным работать с параллельными иерархическими структурами, отслеживая изменения в каждой из них при помощи вводимых временных осей.

Литература

1. Альфред В., Хопкрофт Д, Ульман Д. Структуры данных и алгоритмы. М. : Издательский дом "Вильямс", 2000. 384 с.
2. Астанин С.В., Драгныш Н.В., Жуковская Н.К. Вложенные метаграфы как модели сложных объектов // Инженерный вестник Дона, 2012, №4. URL: ivdon.ru/magazine/archive/n4p2y2012/1434.
3. Макконелл Дж. Основы современных алгоритмов. М: Техносфера, 2004. 368 с.
4. Гудман С., Хидетниемеи С. Введение в разработку и анализ алгоритмов. М.: Мир, 1981. 368 с.
5. Воеводин В.В., Воеводин В.Вл. Параллельные вычисления. СПб.:БХВ-Петербург,2002. 608 с.
6. Поляков В.С., Поляков С.В., Федченков П.В. Построение формального описания технологического процесса в матрично-предикатной форме // Известия ВолгГТУ. Серия "Прогрессивные технологии в машиностроении". Вып. 9 : межвуз. сб. науч. ст. Волгоград: ВолгГТУ, 2013. № 7 (110). С. 105-108.
7. Поляков В.С., Поляков С. В. Запись алгоритма матрицей инцидентора // Инновации на основе информационных и коммуникационных технологий. Инфо 2014: матер. XI междунар.научн.-практ. Конф.(г. Сочи, 1–10 окт. 2014) /Национальный исследовательский ун-т «Высшая школа экономики»[и др.]. М., 2014. С. 149-152.

8. Поляков В.С., Поляков С.В. Использование нагруженных матриц инцидентора (операторов) для моделирования сложных систем // Контроль. Диагностика. 2013. № 3. С. 57-62.

9. Поляков В.С., Поляков С.В., Нефедьев А.И. Математическая модель подвески подвижной части электроизмерительного прибора.// Инженерный вестник Дона. 2013. №3. URL: ivdon.ru/magazine/archive/n3y2013/1788.

10. Voevodin V.V. Information structure of sequential programs // Russ. J. of Num. An. and Math. Modelling. 1995. V.10. №3. pp. 279–286.

11. Voevodin V.V. Mathematical foundations of parallel computing. World Scientific Publishing Co., Series in computer science. 1992. V.33. 343 p.

References

1. Al'fred V., Hopkroft D., Ul'man D. Struktury dannyh i algoritmy [Data structures and algorithms]. M. : Izdatel'skij dom "Vil'yame", 2000. 384 p.

2. Astanin S.V., Dragnysh N.V., Zhukovskaja N.K. Inženernyj vestnik Dona (Rus), 2012, №4. URL: ivdon.ru/magazine/archive/n4p2y2012/1434.

3. Makkonell Dzh. Osnovy sovremennyh algoritmov [Fundamentals of modern algorithms]. M: Tekhnosfera, 2004. 368 p.

4. Gudman S., Hidetniemi S. Vvedenie v razrabotku i analiz algoritmov [Introduction to the development and analysis of algorithms]. M.: Mir, 1981. 368 p.

5. Voevodin V.V., Voevodin V.Vl. Parallelnye vychisleniya [Parallel computing]. SPb.:BHV-Peterburg, 2002. 608 p.

6. Polyakov V.S., Polyakov S.V., Fedchenkov P.V. Izvestiya VolgGTU. Seriya "Progressivnye tekhnologii v mashinostroenii". Vyp. 9 : mezhvuz. sb. nauch. st. Volgograd: VolgGTU, 2013, № 7 (110). С. 105-108.

7. Polyakov V.S., Polyakov S. V. Innovacii na osnove informacionnyh i kommunikacionnyh tekhnologij. (Prod. Info 2014: mater. XI mezhdunar.nauchn.-



prakt. Konf.(g. Sochi, 1–10 okt. 2014)). Moscow: Nacional'nyj issledovatel'skij un-
t «Vysshaya shkola ehkonomiki, 2014. pp. 149-152.

8. Polyakov V.S., Polyakov S.V. Kontrol'. Diagnostika. 2013. № 3.
pp. 57-62.

9. Poljakov V.S., Poljakov S.V., Nefed'ev A.I. Inzhenernyj vestnik Dona
(Rus), 2013. №3. URL: ivdon.ru/magazine/archive/n3y2013/1788.

10. Voevodin V.V. Russ. J. of Num. An. and Math. Modelling, 1995.
V.10. №3. pp. 279–286.

11. Voevodin V.V. Mathematical foundations of parallel computing.
World Scientific Publishing Co., Series in computer science, 1992. V.33. 343 p.