УДК 519.71

Эффективный алгоритм распараллеливания сетей потоков работ с одним ресурсом

Е. А. Приходько
E-mail: ameliya9@mail.ru
Ярославский государственный университет им. П.Г. Демидова

Аннотация

Получен новый алгоритм, применимый к ресурсным workflow-сетям с одним ресурсом, основанный на методе регионов. Доказана корректность алгоритма и полиномиальная трудоёмкость относительно размера исходной сети. Приведены примеры его работы. Новый алгоритм позволяет оптимизировать ресурсные workflow-сети с одним ресурсом (с бесконечным числом состояний).

Ключевые слова: сеть Петри, workflow-сеть, ресурсная workflow-сеть, минимальный регион, метод регионов.

Введение

В последнее время всё более широкое распространение получают расширенные модели потоков работ, в которых помимо управления самим процессом моделируются какие-то дополнительные аспекты функционирования системы: потребление и производство ресурсов, взаимодействие с другими системами, вероятность/длительность работ и т. п. Один из важных случаев — потоки работ с ресурсами. Ресурсы, как правило, моделируются переменными (счётчиками), из-за чего система приобретает бесконечное множество состояний. А к таким схемам процессов уже не применимы алгоритмы, разработанные для обычных потоков работ (конечных автоматов). В частности, их нельзя распараллеливать при помощи метода регионов.

Цель данной работы — разработка алгоритма распараллеливания для схем потоков работ с ресурсами. Рассматривается частный случай — системы с одним счетчиком (одной ресурсной переменной). Этот класс достаточно важен, поскольку позволяет добавлять в модель какой-то ключевой ресурс, обладающий количеством: деньги, время, исполнителей, энергию и т. п. Идея нового метода распараллеливания была сформулирована ранее в работе [1]. В данной статье приводится формальное определение алгоритма, доказывается его корректность, а также рассматривается ряд свойств.

Мультимножества

Мультимножество — множество, в котором допускается повторение элементов. В последующих определениях X — непустое конечное множество.

Мультимножество M над множеством X — функция $M : X \to Nat$. Множество всех конечных мультимножеств над X обозначается $\mathcal{M}(X)$.

Пусть $M_1, M_2, M_3 \in \mathcal{M}(X)$, тогда операции и отношения над мультимножествами выглядят следующим образом:

- $M_1 = M_2 \Leftrightarrow \forall x \in X\ M_1(x) = M_2(x)$ — отношения равенства;
- $M_1 \subseteq M_2 \Leftrightarrow \forall x \leq X\ M_1(x) = M_2(x)$ — отношение включения;
- $M_1 \subset M_2 \Leftrightarrow M_1 \subseteq M_2 \wedge \exists x \in X\ M_1(x) \leq M_2(x)$ — отношение строгого включения;
- $M_1 = M_2 + M_3 \Leftrightarrow \forall x \in X\ M_1(x) = M_2(x) + M_3(x)$ — операция сложения;
- $M_1 = M_2 \cap M_3 \Leftrightarrow \forall x \in X\ M_1 = \min(M_2(x), M_3(x))$ — операция пересечения;
- $M_1 = M_2 - M_3 \Leftrightarrow \forall x \in X\ M_1(x) = M_2(x) \ominus M_3(x)$ — разность мультимножеств;
- $M_1 = kM_2, k \in Nat \Leftrightarrow \forall x \in X\ M_1(x) = kM_2(x)$ — операция умножения на скаляр.

Сети Петри

Обыкновенная сеть Петри — набор $N = (P, T, F)$, где

- P — конечное множество позиций;
- T — конечное множество переходов, $P \cap T = \varnothing$;
- $F : (P \times T) \cup (T \times P) \to Nat$ — функция инцидентности.

Графически сеть Петри изображается как двудольный ориентированный граф. Вершины-позиции изображаются кружками и характеризуют локальные состояния сети, вершины-переходы изображаются прямоугольниками и соответствуют действиям моделируемой системы. Дуги в графе (стрелки) соответствуют элементам F. В сетях Петри допустимы кратные дуги, которые обозначаются соответствующим количеством стрелок.

Разметка сети N — функция $M : P \to Nat$, сопоставляющая каждой позиции некоторое неотрицательное целое число.

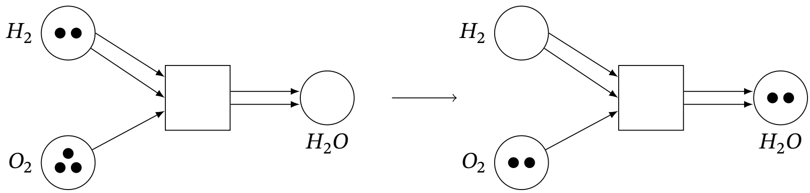

Рис. 1: Сеть Петри, моделирующая химическую реакцию

Для некоторой сети Петри $N = (P, T, F)$:

- Через $\bullet t$ и $t\bullet$, где $t \in T$ — переход, обозначим мультимножества его входных и выходных позиций, таких, что

$$\forall p \in P \bullet t(p) =_{\text{def}} F(p, t), t \bullet (p) =_{\text{def}} F(t, p).$$

- Переход $t \in T$ готов к срабатыванию при разметке M, если $\bullet t \subseteq M$.
- Готовый к срабатыванию переход t может сработать, порождая новую разметку $M' =_{\text{def}} M - \bullet t + t\bullet$.

Пусть (N, M_0) — маркированная сеть Петри, тогда разметка M назывется достижимой, если существует последовательность переходов $\sigma \in T^*$, переводящая сеть из начального состояния M_0 в состояние M.

Для сетей Петри определяется понятие ограниченности, связанное с ограничением количества фишек в позициях: позиция p_i называется k-ограниченной, если количество фишек в ней не может превышать целого числа k. Сеть Петри называется ограниченной, если все ее позиции ограничены.

Workflow-сети

Workflow-сеть [2] — сеть Петри $N = (P, T, F)$, в которой:

- в множестве P имеются две специальные позиции i и o, такие, что $\bullet i = o\bullet = \varnothing$ (отсутствуют переходы в позицию i и из позиции o);
- любой элемент множества $P \cup T$ лежит на пути из i в o.

Позиция i называется начальной, позиция o — финальной. Начальная разметка WF-сети всегда состоит из одной ресурсной фишки в позиции i.

Правильно построенные workflow-сети, которые гарантируют правильное завершение моделируемого процесса, должны удовлетворять свойству бездефектности (soundness), таким образом существует подкласс бездефектных WF-сетей.

WF-сеть $N = (P, T, F)$ называется бездефектной (sound), если:

- Для любого состояния M, достижимого из состояния i, существует последовательность срабатываний, переводящая состояние M в состояние o:

$$\forall M \in R(N, i)\ o \in R(N, M).$$

- Состояние o является единственным состоянием, которое достижимо из состояния i и содержит хотя бы одну фишку в позиции o:

$$o + M' \in R(N, i) \Rightarrow M' = \varnothing.$$

- В сети (N, i) нет мертвых переходов:

$$\forall t \in T \exists M \in R(N, i) : \bullet t \subseteq M.$$

Другими словами, в бездефектной сети финального состояния можно достичь из любого другого, при этом в финальном состоянии не остаётся «лишних» фишек. Так же в бездефектной сети не может быть переходов, которых нельзя достичь никоим образом.

Метод регионов

Для бездефектных workflow-сетей проблема распараллеливания может быть решена при помощи метода регионов. Регион — это подмножество состояний, в которых все переходы, помеченные одним и тем же событием e, имеют отношение «вход/выход». Это отношение станет отношением предшественника/преемника в сети.

Система переходов TS — набор (S, E, A, s_{in}), где:

- S— множество состояний;
- E— множество событий;
- A— множество переходов;
- s_{in}— начальное состояние.

Пусть $TS = (S, E, A, s_{in})$ — это система переходов. $S' \subseteq S$ — подмножество множества состояний, $e \in E$ — событие. Тогда события непересечения, входа и выхода выглядят следующим образом:

- $nocross(e, S') \equiv \exists (s_1, e, s_2) \in A : s_1 \in S' \Leftrightarrow s_2 \in S'$;
- $enter(e, S') \equiv \exists (s_1, e, s_2) \in A : s_1 \notin S' \wedge s_2 \in S'$;
- $exit(e, S') \equiv \exists (s_1, e, s_2) \in A : s_1 \in S' \wedge s_2 \notin S'$.

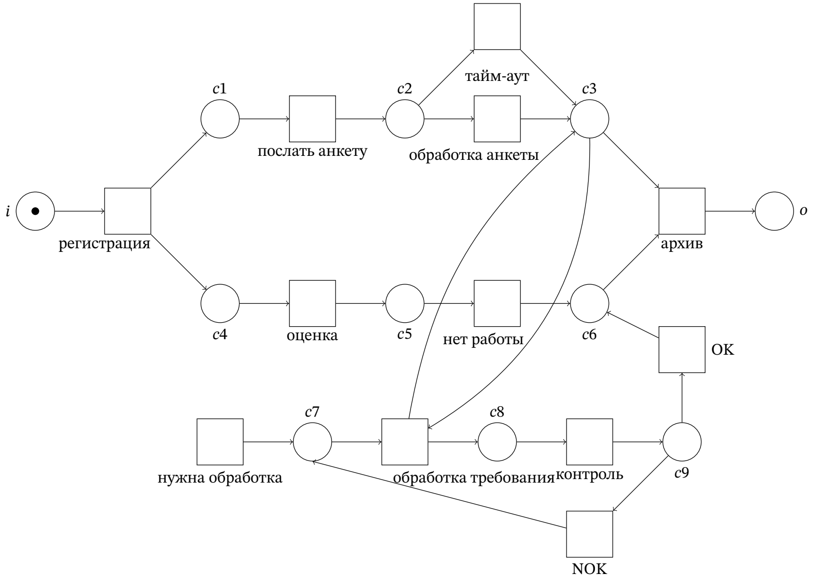

Рис. 2: Пример workflow-сети, отображающей бизнес-процесс обработки страховых требований

Множество состояний $S' \subseteq S$ в системе переходов $TS = (S, E, A, s_{in})$ называется регионом, если для каждого события $e \in E$ выполняются следующие два условия:

- $enter(e, S') \Rightarrow \neg nocross(e, S') \wedge \neg exit(e, S')$;
- $exit(e, S') \Rightarrow \neg nocross(e, S') \wedge \neg enter(e, S')$.

Пусть r и r' — регионы системы переходов. Регион r' называется субрегионом r, если $r \subset r'$. Регион r является минимальным регионом, если нет другого r' региона, который является субрегионом r.

Известно, что бездефектные workflow-сети являются ограниченными сетями Петри (то есть могут быть преобразованы в конечную систему переходов), поэтому для их распараллеливания можно предложить следующий достаточно очевидный алгоритм:

1. На вход алгоритма подаётся бездефектная WF-сеть.
2. Построение для данной сети эквивалентной ей системы переходов.
3. Применение к системе переходов метода регионов.

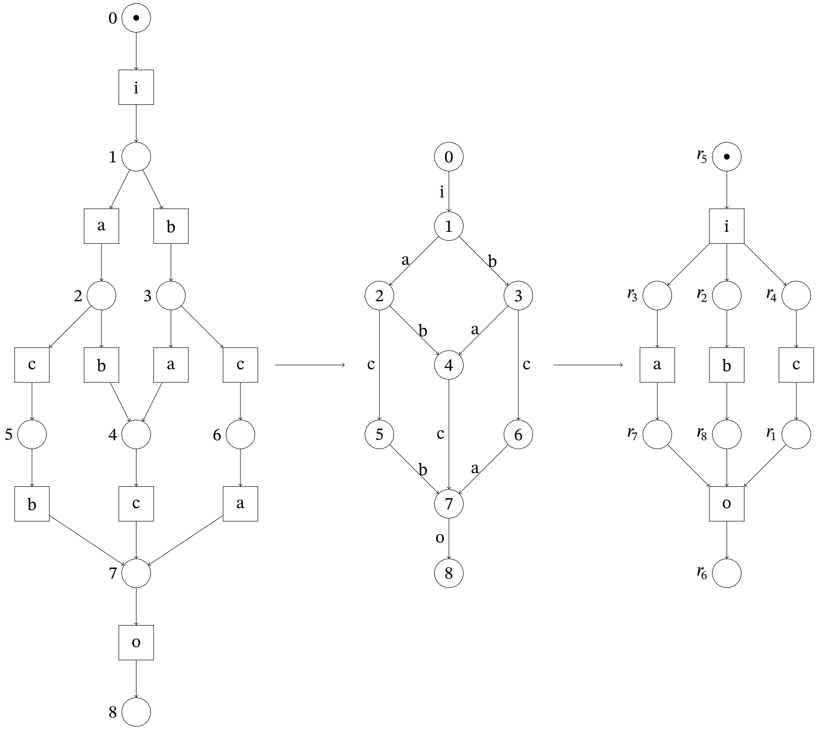

Рис. 3: Пример выполнения алгоритма

4. Преобразование новой системы переходов в WF-сеть.
5. На выходе алгоритма получается оптимизированная распараллеленная WF-сеть.

Схемы потоков работ с ресурсами

В ряде случаев обыкновенные схемы потоков работ недостаточно точно моделируют реальные системы. Требуется добавление тех или иных переменных величин: неограниченных счетчиков, стеков, строк и т. п. Однако при таком расширении синтаксиса множество состояний модели становится бесконечным и все эффективные методы её трансляции становятся неприменимыми. Анализ процессов с бесконечными пространствами состояний представляет собой гораздо более сложную задачу, требующую применения каких-то новых подходов и алгоритмов. При разработке workflow-процессов большое внимание уделяется управлению ресурсами. Ресурсы в данном случае понимаются в обобщённом смысле — это могут быть исполнители (люди или устройства), сырьё, финансы и т. д.

Сетью потоков работ с ресурсами (RWF-сетью) [3; 4] называется набор $N = (P_c, P_r, T, F_c, F_r)$, где

- $N_c = (P_c, T, F_c)$ — обыкновенная WF-сеть (называемая управляющей подсетью сети N, при этом элементы множеств P_c и F_c называются управляющими позициями и дугами соответственно);
- P_r — конечное множество ресурсных позиций, $P_c \cap P_r = \varnothing$;
- $F_r : (P_r \times T) \cup (T \times P_r) \to Nat$ — конечное множество ресурсных дуг.

Разметка RWF-сети распадается на две составляющие — управляющую и ресурсную часть. Мультимножество $c + r$, в котором $c \in M(P_c)$ и $r \in M(P_r)$, будем обозначать как $(c|r)$. Графически ресурсные позиции обозначаются овалами, ресурсные дуги — пунктирными стрелкам.

В RFW-сетях изменено понятие бездефектности. Размеченная RWF-сеть $(N, c \mid r)$ называется бездефектной, если $\forall s \in \mathcal{M}(P_r)$, $\forall M \in \mathcal{R}(N, c|r + s)$ выполняется:

- $\exists s' \in \mathcal{M}(P_r) : o|s' \in \mathcal{R}(N, M)$;
- $c'|r' \in \mathcal{R}(N, M) \Rightarrow c' = 0 \vee c' \cap o = 0$.

Таким образом процесс может правильно завершиться при любом развитии событий, а также не запрещено создание и уничтожение ресурсов.

Однако при таком расширении синтаксиса множество состояний модели становится бесконечным. Анализ процессов с бесконечными пространствами состояний представляет собой гораздо более сложную задачу, требующую применения каких-то новых подходов и алгоритмов. Также из-за этого к RWF-сетям нельзя применить метод регионов для оптимизации в чистом виде.

Алгоритм распараллеливания workflow-сетей с одним ресурсом

В общем случае, при наличии более чем одного ресурса, задача по оптимизации ресурсных workflow-сетей оказалось достаточно сложной, так как наличие нескольких ресурсных позиций усложня-

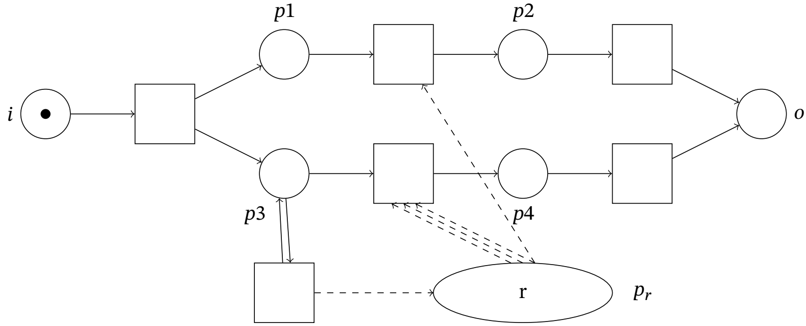

Рис. 4: Бездефектная RWF-сеть

ют анализ сети. Поэтому в данной работе рассматриваются WF-сети с одним ресурсом.

Уже существует работа [5], которая позволяет синтезировать сети Петри из односчётчиковых контуров с помощью метода регионов. Однако этот метод для workflow-сетей с одним ресурсом не подходит, так как односчётчиковые контуры представляют из себя сильно связные односчётчиковые сети. А в любых workflow-сетях есть начальное и финальное состояние, поэтому условие сильной связности нарушается.

В отличие от классического метода регионов, при анализе исходного автомата в качестве идентификатора действия предлагается использовать не букву алфавита действий (метку перехода), а комбинацию из этой буквы и целого числа — величины изменения счётчика (эффект перехода).

Алгоритм для распараллеливания RWF-сетей выглядит следующим образом (вход — WF-сеть с одним счётчиком, выход — распараллеленная WF-сеть с одним счётчиком):

1. Построим по данной WF-сети с одним ресурсом эквивалентную односчётчиковую сеть. По построению её диаграмма переходов будет содержать одну начальную и одну конечную вершины, а все промежуточные вершины будут лежать на путях из начальной в конечную.
2. Применим к односчётчиковой сети модифицированный метод регионов, рассматривая в качестве идентификатора перехода соответствующую пару (метка, эффект) [6; 7].

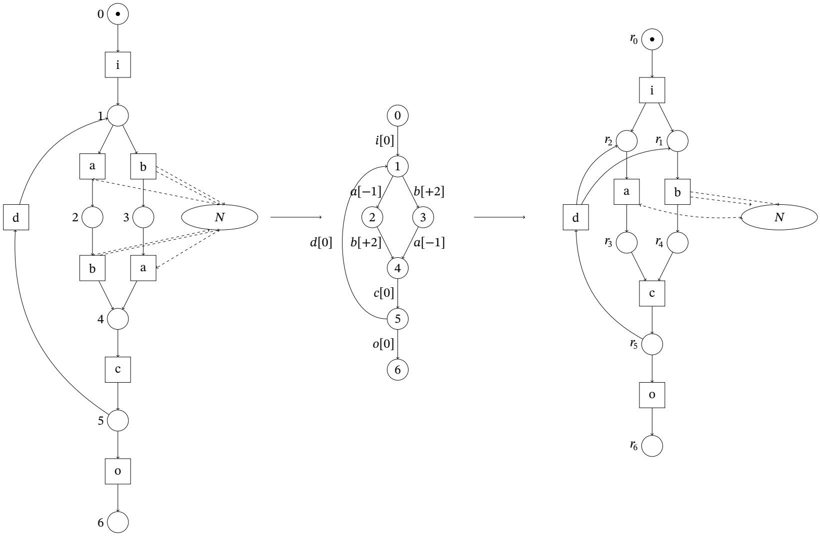

Рис. 5: Пример выполнения алгоритма

3. Преобразуем новую систему переходов в WF-сеть с одним ресурсом.

На примере на рис. 5 поведение полученной сети сохраняет свойства исходной: если начальное значение счётчика 0, то на первой итерации первым всегда будет срабатывать переход b; на последующих итерациях — в любом порядке. Если начальное значение счётчика отлично от нуля, то полученная сеть, как и раньше, сохраняет своё поведение, имея возможность выбрать любой переход.

Корректность алгоритма

Докажем, что распараллеленная алгоритмом RWF-сеть N' порождает язык (множество строк), в точности совпадающий с языком исходной RWF-сети N.

Пусть M — односчётчиковый автомат, построенный на первом этапе. Он по построению эквивалентен N, то есть порождает такой же язык. Соответственно, надо доказать, что язык, порождаемый N' совпадает с языком, порождаемым M.

Рассмотрим в качестве алфавита E не исходный алфавит символов (A, например $A = a, b, c$), а алфавит пар (символ, эффект), то есть $E = a[+1], b[-2], c[-1], d[0]$. Если рассматривать M как автомат над языком E (а не A), то он представляет собой обычный конечный автомат. Обозначим такой автомат как M_e. Фактически именно к нему применяется метод регионов. И результирующая сеть N' (если убрать из неё ресурсную позицию и инцидентные ей ресурсные дуги, а переходы пометить метками из E, а не метками из A) — эквивалентная M ограниченная сеть Петри над алфавитом E. Обозначим такую сеть как N'_e. Метод регионов корректен для конечных автоматов, следовательно, языки M_e и N'_e совпадают. Таким образом исключается из рассмотрения ресурс (или рассматривается как бесконечный), но добавляется эффект к метке перехода.

Очевидно, что язык автомата M является подмножеством языка, полученного из языка автомата M_e «стиранием» всех эффектов (например, цепочка $a[+1]b[-2]a[+1]$ рассматривается как aba). Из него исключены те строки, на которые не хватило ресурса (например, если начальный ресурс — 0, то строка $a[+1]b[-2]a[+1]$ не получится). Аналогично, и для сетей Петри — язык N' является подмножеством «очищенного от эффектов» языка N'_e. Ресурса так же может не хватить на какие-то цепочки.

Таким образом, единственное, что необходимо доказать, — что начальный ресурс одинаковым образом ограничивает срабатывания и в случае автомата, и в случае сети Петри. Это очевидно — ограничения ресурса действуют посредством сложения последовательно идущих друг за другом в цепочке эффектов. А цепочки в языках M_e и N'_e полностью совпадают, следовательно, и эффекты суммируются одинаковым образом. Таким образом язык N' совпадает с языком M.

Оценка трудоёмкости

Трудоёмкость первого и третьего шагов алгоритма линейна. Наиболее трудоёмким является второй шаг, в котором необходимо выделить минимальные регионы. Самый простой способ это сделать — перебрать все подмножества из множества состояний в системе переходов и проверить, являются ли они регионами, а затем из них выделить минимальные. В таком случае трудоёмкость будет $O(2^N)$, где N — число состояний. Однако, можно использовать алгоритм, основанный на понятии региона возбуждения.

Регион возбуждения события e, $ER(e)$ — это множество состояний, в которые включено e: $ER(e) = s|\exists s' : (s, e, s') \in A$. Сначала формируются регионы возбуждения для каждого события, а затем, если регион

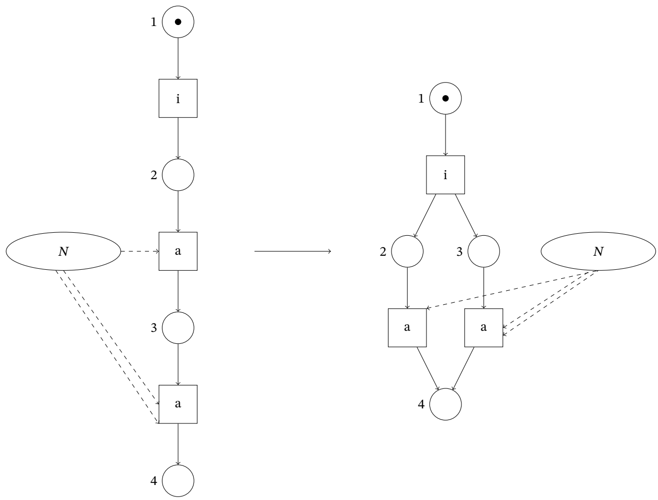

Рис. 6: Пример RWF-сети, для которой данный метод не эффективен

возбуждения нарушает условие региона, включаются события, чтобы соответствовать условию. Таким образом сложность алгоритма будет $O(N \cdot M)$, где N — количество различных событий, M — множество состояний.

Ограничения алгоритма

Важно заметить, что существуют такие RWF-сети, которые после применения алгоритма, преобразовываются в такие, которые распараллелены не максимально, или вообще могут не измениться. На рис. 6 представлены две эквивалентные RWF-сети, порождающие язык $\{iaa\}$ при значении счётчика больше 2. Однако данный алгоритм не сможет оптимизировать левую сеть до состояния правой, так как эффект у переходов разный. Данный пример подтверждает нетривиальность задачи и показывает возможные направления дальнейших исследований.

Заключение

В работе предложен алгоритм распараллеливания схем потоков работ с одномерными ресурсами. Показано, что новый метод позволяет повышать эффективность систем (степень параллелизма) при полном сохранении их поведения.

Список литературы

1. *Приходько Е. А.*, *Башкин В. А.* Распараллеливание схем потоков работ с одним ресурсом // Путь в науку: прикладная математика, информатика и информационные технологии : Сборник тезисов конференции. Ярославль : ЯрГУ, 2023. С. 61—63.
2. *Аалст В. ван дер*, *Хей К. ван*. Управление потоками работ: модели, методы и системы. М. : Физматлит, 2007. 316 с.
3. *Bashkin V. A.*, *Lomazova I. A.* Resource equivalence in workflow nets // Proceedings of Concurrency, Specification and Programming (CS&P-2006). Vol. 1. Berlin : Humboldt Universität, 2006. P. 80–91.
4. *Башкин В. А.*, *Ломазова И. А.* О разрешимости бездефектности для сетей потоков работ с неограниченным ресурсом // Моделирование и анализ информационных систем. Ярославль. 2013. Т. 20, № 4. С. 23—40.
5. *Лебедева Н. В.*, *Башкин В. А.* Об одном алгоритме синтеза сетей Петри из односчетчиковых контуров // Заметки по информатике и математике. 2020. Вып. 12. С. 171—179.
6. *Cortadella J.* [et al.]. Deriving Petri Nets from Finite Transition Systems // IEEE Transactions on Computers. 1998. Vol. 8, no. 47. P. 8596–882.
7. *Carmona J.*, *Cortadella J.*, *Kishinevsky M.* A Region-based Algorithm for Discovering Petri Nets from Event Logs // International Conference on Business Process Management. 2008. P. 358–373.