

УДК 519.6

Лента достижимости J-сети, моделирующей применение алгоритма оптимизации последовательности отбора к одной задаче

А. П. Димитриев¹, Т. А. Лавина¹, Р. И. Баженов², Т. Н. Копышева¹¹ЧГУ им. И.Н. Ульянова (г. Чебоксары, Российская Федерация)²Приамурский государственный университет имени Шолом-Алейхема (г. Биробиджан, Российская Федерация)

Аннотация. Распределение учебной нагрузки на кафедре ранее было formalизовано как задача комбинаторной дискретной оптимизации, и для её решения эффективным является применение алгоритма оптимизации последовательности отбора. Этот алгоритм разработан авторами в более ранних работах и использует, в частности, принципы алгоритма имитации отжига. Разработана блок-схема для данного алгоритма с целью наглядного представления работы такого алгоритма, необходимого для дальнейшего понимания излагаемого материала. Для моделирования динамики алгоритма разработана математическая модель на основе одной из разновидностей цветных сетей Петри – J-сети. Детально описана логика работы этой модели. Построена лента достижимости J-сети, содержащая 269 маркировок, часть из которых, представляющая характерные особенности, приводится в статье. Для сокращения размера ленты достижимости приняты некоторые допущения. Отбраковка недостижимых маркировок производится путем дополнительного анализа наборов неравенств – результатов сравнения значений целевой функции. В связи со значительным количеством анализируемых неравенств разработано программное средство для решения систем неравенств, алгоритм работы которого обладает полиномиальной временной сложностью. Проведен анализ ленты достижимости, который показывает корректность работы алгоритма оптимизации. Научная новизна: впервые построена лента достижимости для J-сети.

Ключевые слова: лента достижимости, оптимизация последовательности отбора, дискретная оптимизация, учебная нагрузка, J-сеть, сеть Петри

Для цитирования: Димитриев А. П., Лавина Т. А., Баженов Р. И., Копышева Т. Н. Лента достижимости J-сети, моделирующей применение алгоритма оптимизации последовательности отбора к одной задаче // Журнал Средневолжского математического общества. 2025. Т. 27, № 4. С. 517–538. DOI: 10.15507/2079-6900.27.202504.517-538

Об авторах:

Димитриев Александр Петрович, к. т. н., доцент кафедры компьютерных технологий, ФГБОУ ВО «ЧГУ им. И. Н. Ульянова» (428015, Россия, г. Чебоксары, Московский проспект, д. 15), ORCID: <http://orcid.org/0000-0002-7345-9790>, dimitrie1@yandex.ru

Лавина Татьяна Ароновна, д. пед. наук, заведующий кафедрой компьютерных технологий, ФГБОУ ВО «ЧГУ им. И. Н. Ульянова» (428015, Россия, г. Чебоксары, Московский проспект, д. 15), ORCID: <http://orcid.org/0000-0002-7622-2246>, tlavina@mail.ru



Баженов Руслан Иванович, к. пед. наук, заведующий кафедрой информационных систем, математики и правовой информатики, ФГБОУ ВО Приамурский государственный университет имени Шолом-Алейхема (679015, Россия, г. Биробиджан, ул. Широкая, д. 70а), ORCID: <http://orcid.org/0000-0003-2668-1142>, r-i-bazhenov@yandex.ru

Копышева Татьяна Николаевна, к. ф.-м. н., заведующий кафедрой математического и аппаратного обеспечения информационных систем, ФГБОУ ВО «ЧГУ им. И. Н. Ульянова» (428015, Россия, г. Чебоксары, Московский проспект, д. 15), ORCID: <http://orcid.org/0000-0003-3392-1431>, tn_pavlova@mail.ru

Original article

MSC2020 90-10

Reachability tape of J-net simulating the application of selection sequence optimization algorithm to single problem

A. P. Dimitriev¹, T. A. Lavina¹, R. I. Bazhenov², T. N. Kopysheva¹

¹*Chuvash State University (Cheboksary, Russian Federation)*

²*Sholom-Aleichem Priamursky State University (Birobidzhan, Russian Federation)*

Abstract. In Russian universities, the problem of distributing the departmental educational load is annually resolved. This problem belongs to the class of combinatorial discrete optimization problems. To solve some problems of this class, it is effective to use the selection sequence optimization algorithm developed by the authors earlier. This article provides a flowchart of this algorithm for the purpose of its visual presentation, which is aimed at further understanding of the material presented. Modeling the dynamics of the algorithm is carried out using a mathematical model built on the basis of one of the varieties of colored Petri nets - that is, on the J-net. The operating logic of this model is described in detail. A reachability tape has been built for this J-net. It contains 269 markings, some of which, representing the main nuances, are presented in the paper in table form. The reachability tape has a significant size even with a minimal non-trivial amount of simulated data, so unattainable markings are rejected by additional analysis of some inequalities sets. Due to the difficulty of contemplative analysis of inequalities' systems, a software tool has been developed for solving these systems, whose algorithm has polynomial time complexity. Analysis of the reachability tape shows the correct operation of the optimization algorithm. Scientific novelty of the work is that for the first time, a reachability tape for the J-net has been constructed.

Keywords: reachability tape, selection sequence optimization, discrete optimization, educational load, J-net, Petri net

For citation: A. P. Dimitriev, T. A. Lavina, R. I. Bazhenov, T. N. Kopysheva. Reachability tape of J-net simulating the application of selection sequence optimization algorithm to single problem. *Zhurnal Srednevolzhskogo matematicheskogo obshchestva*. 27:4(2025), 517–538. DOI: [10.15507/2079-6900.27.202504.517-538](https://doi.org/10.15507/2079-6900.27.202504.517-538)

About the authors:

Alexander P. Dimitriev, Ph.D. (Engineering), Associate professor, Department of Computer Technologies, Chuvash State University (15 Moskovsky av., Cheboksary 428015, Russia), ORCID: <http://orcid.org/0000-0002-7345-9790>, dimitrie1@yandex.ru

Tatyana A. Lavina, Dr. Sci. (Pedagogy), Head of Department of Computer Technologies, Chuvash State University (15 Moskovsky av., Cheboksary 428015, Russia), ORCID: <http://orcid.org/0000-0002-7622-2246>, tlavina@mail.ru

Ruslan I. Bazhenov, Ph. D. (Pedagogy), Head Department of Information Systems, Mathematics and Legal Informatics, Sholom-Aleichem Priamursky State University (70a Shirokaya St., Birobidzhan 679015, Russia), ORCID: <http://orcid.org/0000-0003-2668-1142>, r-i-bazhenov@yandex.ru

Tatyana N. Kopysheva, Ph. D. (Phys.-Math.), Head of Department of Mathematical and Hardware Support for Information Systems, Chuvash State University (15 Moskovsky av., Cheboksary 428015, Russia), ORCID: <http://orcid.org/0000-0003-3392-1431>, tn_pavlova@mail.ru

1. Введение

Одним из методов моделирования динамики систем является использование сетей Петри. Исследователи применяют такой подход в самых разных областях и, в частности, при анализе организации учебного процесса в образовательных учреждениях. Так, в [1] с помощью цветных сетей Петри моделируется учебный план на уровне колледжа. В [2] дан обзор того, как цветные сети Петри могут использоваться для моделирования процессов высшего образования, а также на их основе разработана модель административного процесса на кафедре, модель курса и модель учебной дисциплины. В [3] на основе сетей Петри сформулирована модель, описывающая динамику обучения студентов университета и взаимодействия с учебным планом. В статье [4] предлагается обучение формальным методам с использованием цветных сетей Петри в Технологическом университете Суранари (Таиланд). В [5] на основе сетей Петри построена модель управления гражданским и политическим образованием в колледжах и университетах в Китае. В [6] сети Петри использованы при изучении применения преподавателями университета панели управления аналитикой обучения для онлайн-сотрудничества. В [7] с помощью сетей Петри смоделирован бизнес-процесс языкового института, преподающего английский, немецкий и французский языки, рабочий процесс модульного курса электронного обучения, учебный план и др.

Анализ поведения систем с помощью сетей Петри может осуществляться различными способами: запуск программной реализации Петри-машины, построение дерева достижимых состояний, составление ленты достижимости (ЛД) [8] и др. ЛД представляет дерево достижимости в табличном виде, делая его менее громоздким. В статье рассматривается построение и анализ ЛД для одного из расширений цветных сетей – J-сети, которые представляют собой подкласс синхронных цветных сетей Петри [9].

В качестве предметной области, которую моделирует J-сети, в работе используеться оптимизация распределения учебной нагрузки научно-педагогических работников (НПР) кафедры при помощи одного из эффективных оптимизационных алгоритмов – алгоритма оптимизации последовательности отбора (АОПО) [10]. Это актуально, поскольку в вузах России кафедрами распределение учебной нагрузки педагогических работников производится ежегодно и является нетривиальной задачей. Нагрузка измеряется в академических часах и для отдельных педагогов, например, в ФГБОУ ВО «ЧГУ

им. И.Н. Ульянова», может распределяться на учебный год по четырем видам внутривузовских документов, называемых «Индивидуальный план» (ИП): штатная (основная) нагрузка, нагрузка по совместительству, почасовая нагрузка, прочая нагрузка, количество часов в которых обозначим соответственно S_1, S_2, S_3, S_4 .

Когда закончено распределение нагрузки между разными преподавателями, в тех случаях, когда у некоторого преподавателя значение нагрузки превышает максимальную допустимую величину нагрузки по основному месту работы (у разных преподавателей эта величина разная), возникает задача распределения строк нагрузки между разными ИП одного и того же преподавателя, которая и рассматривается в статье. Необходимо соблюдать не только верхний предел в 900 часов за ставку, установленный Минобрнауки России (приказ № 1601 от 22 декабря 2014 г.), но и требования внутривузовских нормативных документов по планированию количества академических часов. Получение такого распределения – задача комбинаторной дискретной оптимизации. Её решению посвящено сравнительно немного работ, имеющих математическое обоснование. Г.П. Виноградовым и Н.В. Кирсановой [11] разработана математическая модель, но ничего не сообщается о программной реализации. В работе Д.А. Ивахненко [12] также детально изложена математическая модель, но нет реализации, а также не упоминаются прочая нагрузка и совместительство. В то же время существуют публикации с близкой темой, такие как статья С. Н. Султановой и С. В. Тархова [13], где, однако, рассмотрено распределение нагрузки между разными преподавателями, а не между ИП одного НПР. В работе авторов [10] разработаны не только теоретические предпосылки для получения оптимального распределения, но и программный продукт. Однако указанная работа является скорее прикладной, нежели теоретической. Для её теоретического обоснования актуальна разработка математического инструментария, позволяющего моделировать динамику такой дискретной системы, как АОПО.

Для решения представленной задачи комбинаторной оптимизации в статье [14] применялся генетический алгоритм типа Genitor [15].

В [10] для этого применён алгоритм оптимизации последовательности отбора [16], что более эффективно. В качестве последовательности отбора (этот термин при использовании в общем значении обозначим σ) будет использоваться σ элементов нагрузки для назначения вида ИП. АОПО реализует алгоритм имитации отжига [17], но не является его разновидностью.

Анализ работы АОПО при распределении учебной нагрузки можно произвести посредством J-сети с помощью ЛД, показывающей достижимые состояния этой сети.

Целью исследования является построение ленты достижимости для J-сети, моделирующей работу алгоритма оптимизации последовательности отбора при распределении учебной нагрузки.

2. Задача оптимизации распределения учебной нагрузки и алгоритм её приближенного решения

Для формализации постановки задачи, в соответствии с нормами в [14] выделены ограничения и критерии оптимальности следующим образом (здесь приводятся только с краткими пояснениями).

Вначале для каждого НПР задаются величины минимальной и максимальной допустимой нагрузки по четырём видам ИП, обозначаемые соответственно E_i и X_i : $E_i \leq S_i \leq X_i$, $i = 1, \dots, 4$. Для этого учитываются нормы значений по должностям,

доля ставки и работа на других кафедрах следующим образом. Предполагается, что задача распределения нагрузки на другой кафедре уже решена и предоставлены значения часов, которые там ведёт данный преподаватель, в виде константы y . Если он там ничего не ведет, $y = 0$.

Обозначим как R_i , $i = \{1; 2\}$ доли ставки по первым двум видам ИП: $0 < R_1 \leq 1$; $0 < R_2 < 0,5$. Обычно преподаватель не ведёт нагрузку на другой кафедре, входящую в S_1 и S_2 , поэтому $X_1 = R_1 \times 900$, $X_2 = R_2 \times 900$. Однако такая нагрузка может входить в S_3 , для которой Учебно-методическим управлением установлено максимальное значение 300 часов. Поэтому $X_3 = 300 - y$. Установлено также $X_4 = 200$.

E_i определяется так: $E_i = R_i \times T$, $i = \{1; 2\}$, где T – значение минимальной нормы для должности данного НПР, например для доцента $T = 860$. Установлено $E_4 = 10$, а $E_3 = 10$, если $S_4 = 0$, и $E_3 = 299$, если $S_4 > 0$.

Правила заполнения ИП для некоторого НПР следующие.

1. Если $S_1 > X_1$, то избыток переводится в S_2 или в S_3 .
2. Должно выполняться $E_2 \leq S_2 \leq X_2$. Если для этого часов недостаточно, избыточные (см. п. 1) часы направляются только в S_3 , а если много, то в S_3 передаётся соответствующий избыток.
3. Должно выполняться неравенство $S_3 \leq 300$. Поэтому избыток S_3 переходит в S_4 . Но если $S_4 > 0$, то должно быть $S_3 \geq 299$.
4. Должно быть $S_4 \leq 200$. Избыток надо распределить между другими НПР, его нельзя оформлять в S_3 (см. предыдущее правило).

5. Нагрузку рекомендуется распределять примерно одинаково по двум полугодиям, т.е. отношения S_{i1}/S_{i2} рекомендуется иметь в пределах от $1/N_i$ до N_i , где N_i – задаваемые значения; $N_i \geq 1$, S_{i1} , S_{i2} – количество часов по S_i соответственно в первом и втором полугодиях.

Всё это унифицированно представлено в виде следующих правил для вычисления значения целевой функции F (исходно $F = 0$):

1. $S_i < E_i \rightarrow F := F + T_i \times (E_i - S_i)$, где $i = 1, \dots, 3$, T_i – задаваемые штрафные коэффициенты для выхода нагрузки за допустимые пределы.
2. $S_i > X_i \rightarrow F := F + T_i \times (S_i - X_i)$, $i = 1, \dots, 4$.
3. $S_1 + S_2 < E_1 + E_2 \rightarrow F := F + E_1 + E_2 - S_1 - S_2$.
4. $S_1 + S_2 > X_1 + X_2 \rightarrow F := F + S_1 + S_2 - X_1 - X_2$.
5. $(S_{i1} > 0) \cap (((S_{i2}/S_{i1}) < 1/N_i) \cup ((S_{i2}/S_{i1}) > N_i)) \rightarrow F := F + |S_{i1} - S_{i2}|/D_i$, где $i = 1, \dots, 4$, D_i – задаваемые штрафные коэффициенты для неравномерности распределения нагрузки по полугодиям; $D_i \geq 1$.
6. $(S_{i1} = 0) \cap (S_{i2} > 0) \rightarrow F := F + S_{i2}/D_i$, где $i = 1, \dots, 4$.

На этой основе сформулирована целевая функция F для задачи условной дискретной оптимизации:

$$F(X) \rightarrow \min, \quad (2.1)$$

$$X \in R^n,$$

$$g_j(X) \leq 0, j = 1, \dots, p \text{ (неравенства), } p = 10,$$

$$h_k(X) = 0, k = 1, \dots, q \text{ (равенства), } q = 12,$$

$$X = (x_1, x_2, \dots, x_n);$$

x_i принадлежат дискретным или непрерывным множествам:

$x_1 = O_1$, $x_2 = O_2$, \dots , $x_C = O_C$; O_i – номер разновидности ИП для i -го элемента нагрузки: $O_i \in \{0, 1, 2, 3\}$, $i = 1, \dots, C$;

$x_{n-4} = S_1; x_{n-3} = S_2; x_{n-2} = S_3; x_{n-1} = S_4; S_i \in R$ (R – множество действительных чисел), $i = 1, \dots, 4$;

$x_n = y; y \in R$ (R – множество действительных чисел);

$n = C+5$; C – число элементов нагрузки, т.е. строк ИП, из которых состоит нагрузка; $C \in Z$ (Z – множество целых чисел);

при областных ограничениях $g_j(X) \leq 0$ либо $g_j(X) < 0$, $j = 1, \dots, 10$:

$$g_1(X) = C - 41 \leq 0; \quad g_2(X) = 1 - C < 0;$$

$$g_3(X) = y - 300 \leq 0; \quad g_4(X) = -y \leq 0;$$

$$g_5(X) = S_i - X_i \leq 0, \quad g_6(X) = E_i - S_i \leq 0, \\ i = 1, \dots, 4; S_i \in R; X_i \in R; \quad i = 1, \dots, 4; E_i \in R; S_i \in R;$$

$$g_7(X) = R_1 - 1 \leq 0, \quad g_8(X) = -R_1 < 0, \\ R_1 \in R; \quad R_1 \in R;$$

$$g_9(X) = R_2 - 0,5 \leq 0, \quad g_{10}(X) = -R_2 < 0, \\ R_2 \in R; \quad R_2 \in R;$$

и функциональных ограничениях $h_k(X) < 0$, $k = 1, \dots, 12$:

$$h_1(X) = X_1 - R_1 \times 900 = 0, R_1 \in R, \quad h_2(X) = X_2 - R_2 \times 900 = 0, R_2 \in R,$$

$$h_3(X) = X_3 - 300 + y = 0; \quad h_4(X) = X_4 - 200 = 0;$$

$$h_5(X) = E_i - R_i \times T = 0, \quad h_6(X) = T - 860 = 0 \\ i = \{1; 2\}; \quad (\text{только для доцента});$$

$$h_7(X) = T - 890 = 0 \quad h_8(X) = T - 800 = 0 \\ (\text{только для старшего преподавателя и ассистента}); \quad (\text{только для профессора});$$

$$h_9(X) = T - 765 = 0 \quad h_{10}(X) = E_3 - 10 = 0 \text{ при } S_4 = 0; \\ (\text{только для заведующего кафедрой});$$

$$h_{11}(X) = E_3 - 299 = 0 \text{ при } S_4 > 0; \quad h_{12}(X) = E_4 - 10 = 0.$$

АОПО предложен авторами [10]. Он решает задачу минимизации (2.1). Этот алгоритм представлен в виде блок-схемы (см. Рис. 2.1, 2.2, 2.3) иллюстративного характера для наглядного представления его работы и внесения определённости при дальнейшем изложении.

В применённом алгоритме АОПО используются переменные: C – число элементов нагрузки; O – массив видов ИП из C элементов; S – массив, представляющий собой σ . Приведём этот алгоритм.

Параметры алгоритма, используемые по умолчанию:

- исходное число итераций $It = 500$;

- увеличение числа итераций за шаг изменения температуры $d = 280$;

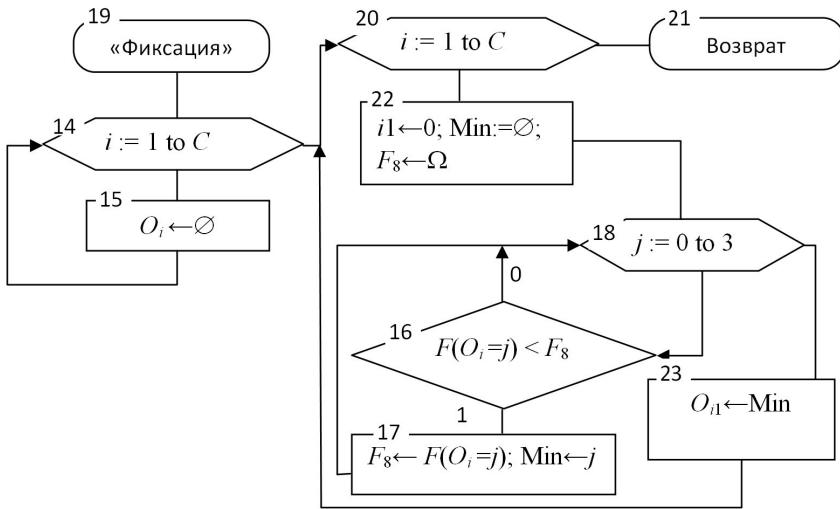


Рис. 2.1. Блок-схема алгоритма для модуля построения близкого к оптимальному решению

Fig. 2.1. Algorithm flowchart for the module for constructing a solution close to the optimal one

- число объектов последовательности, обменивающихся положением $X = 1$;
- исходная температура $t = 5$;
- делитель общего числа итераций (для определения числа шагов) $D_S = 25$.

Алгоритм следующий.

1. Считывание параметров: d, X, t, It, D_S .
2. Присвоить $t_1 \leftarrow t; t_2 \leftarrow t; it_1 \leftarrow 0; it_2 \leftarrow 0$.
3. Вычислить $Z \leftarrow (It \times (11 - t) + (d \times (11 - t)^2) \text{ div } 2) \text{ div } D_S$.
4. Если не установлен флагок «Продолжение» (см. ниже), то получить близкий к оптимальному вариант, сохраняемый в массиве O , и начальную последовательность, сохраняемую в массиве s из C элементов. Для этого выполнять C раз следующее:
 - 4.1. Присвоить номерам ИП $O_i \leftarrow -1$, где $i = 1, \dots, C$. Примечание: значение -1 означает, что вид ИП «недопустим»; тогда при вычислении F полагается, что соответствующей нагрузки нет.
 - 4.2. Цикл k от 1 до C :
 - 4.2.1. Цикл i от 1 до C : Если $O_i = -1$, то:
 - 4.2.1.1. Элементу O_i присваивать значения от 0 до 3, при этом вычисляя разность значений F после и до присвоения, и запоминая в Min тот вид, при котором разность минимальна по всем i , а в i_1 – соответствующее i .

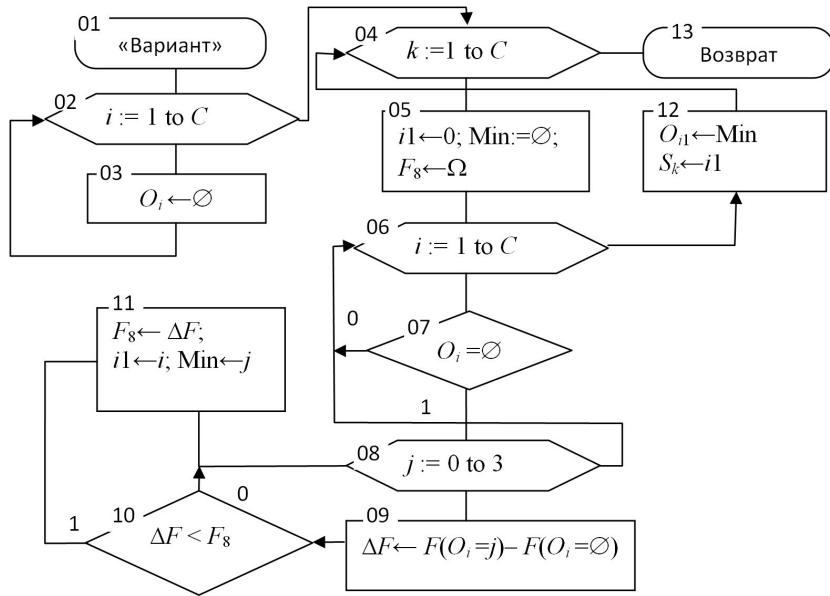


Рис. 2.2. Блок-схема алгоритма для модуля фиксации значений видов нагрузки с минимальным F

Fig. 2.2. Algorithm flowchart for the module for recording values of load types with minimum F

4.2.1.2. Присвоить $O_i \leftarrow -1$.

4.2.2. Зафиксировать Min в O_{i1} и присвоить $s_k \leftarrow i_1$.

5. Скопировать массив s_1 из s .

6. Вычислить исходное значение F : $min_2 \leftarrow F(O)$.

7. Повторять следующие действия:

7.1. Присвоить $fr \leftarrow 0$.

7.2. Пока $fr < It$, выполнять следующие действия.

7.2.1. Присвоить $fr \leftarrow fr + 1$.

7.2.2. Скопировать массив s_2 из s .

7.2.3. Совершить X раз обмен двух случайно выбранных элементов массива s .

7.2.4. Присвоить $O_i \leftarrow -1$, где $i = 1, \dots, C$.

7.2.5. Последовательно каждому элементу массива O присваивать значения от 0 до 3, при этом вычисляя F , и зафиксировать то значение, при котором F минимальна.

7.2.6. Присвоить $min_8 \leftarrow F(O)$; $it_2 \leftarrow it_2 + 1$.

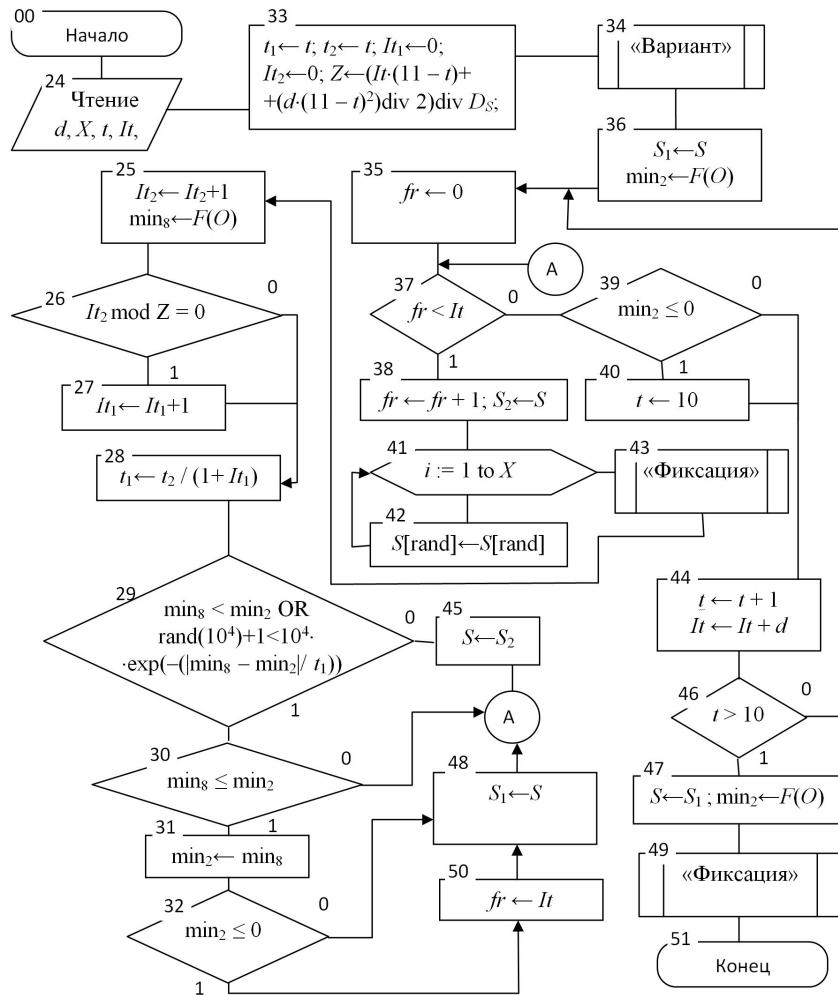


Рис. 2.3. Блок-схема алгоритма (основной модуль)

Fig. 2.3. Algorithm flowchart (the main module)

7.2.7. Если $it_2 \bmod Z = 0$, то $it_1 \leftarrow it_1 + 1$.

7.2.8. Присвоить $t_1 \leftarrow t_2 / (1 + it_1)$.

7.2.9. Если $\min_8 \leq \min_2$ или $(\text{random}(10000) + 1 < 10000 \times e^{-|\min_8 - \min_2|/t_1})$, то:

7.2.9.1. Если $\min_8 \leq \min_2$, то:

7.2.9.1.1. Присвоить $\min_2 \leftarrow \min_8$.

7.2.9.1.2. Если $\min_2 \leq 0$, то присвоить $fr \leftarrow It$.

7.2.9.1.3. Скопировать массив s_1 из s .

7.2.10. В противном случае (см. п. 7.2.9.) скопировать массив s из s_2 .

- 7.3. Если $\min_2 \leq 0$, то присвоить $t \leftarrow 10$.
- 7.4. Присвоить $t \leftarrow t + 1$; $It \leftarrow It + d$.
- 7.5. Если $t > 10$, то перестать повторять и перейти к следующему пункту.
8. Скопировать массив s из s_1 .
9. Присвоить $O_i \leftarrow -1$, где $i = 1, \dots, C$.
10. Последовательно каждому элементу массива O присваивать значения от 0 до 3, при этом вычисляя F , и зафиксировать то значение, при котором F минимальна. Конец.

В результате вычислений в массиве O будет находиться искомое распределение номеров элементов нагрузки по видам ИП.

3. Математическая модель работы алгоритма оптимизации последовательности отбора

Для представления работы алгоритма АОПО разработана следующая модель на основе J-сети (Рис. 3.1).

На Рис. 3.1 наличие стрелок на обеих сторонах дуг (в том числе прямолинейных) означает, что имеется одна дуга в одну сторону и одна дуга в обратную сторону. Число в прямоугольнике над дугой – это, как и в обычных сетях Петри, кратность дуги. Рассмотрим особенности остальных составляющих данной модели и правила её функционирования.

В отличие от обычных сетей Петри, все позиции, кроме P_A, P_B, P_I , являются макропозициями (для краткости далее – позициями), содержащими места локализации маркеров (называемых также фишками или метками) – таблицы, состоящие из одной, трех или четырех ячеек, изображённых в виде квадратов, с информационными маркерами (для краткости далее – маркерами). Это позволяет рассматривать данную модель как J-сеть. В данном случае синхронность, присущая J-сетям и подразумевающая срабатывание более одного перехода за такт [18], не учитывается, так как в ней нет надобности: переходов всего два, и одновременно они срабатывать не могут. Для полного соответствия концепции можно представить, что переходы состоят из подсетей, включающих по нескольку переходов, которые выполняют описываемые ниже действия, и всё это за один такт.

В позициях P_1, \dots, P_C в верхней части квадрата – номер маркера, представляющий собой также номер вида нагрузки (от 0 до 3). В нижней части квадрата либо g_k – количество часов для k -го элемента нагрузки («несущий маркер»), $k = 1, \dots, C$, либо 0, что означает, что выбор вида нагрузки не сделан для данного элемента нагрузки («нуль-маркер»). Несущие маркеры могут быть просто несущими (временно) либо на временно-постоянной основе (ВПО).

Процесс работы сети состоит из двух этапов. На первом этапе производится получение близкого к оптимальному варианта распределения нагрузки (см. Рис. 2.1), а на втором – оптимизация полученного варианта.

Состояние сети в любой момент τ описывается маркировкой $M_\tau = \{m(P_0), \dots, m(P_R)\}$, представляющей собой набор разметок для каждой позиции P_0, \dots, P_R , где аргументом отображения $m(x)$ является название x позиции сети,

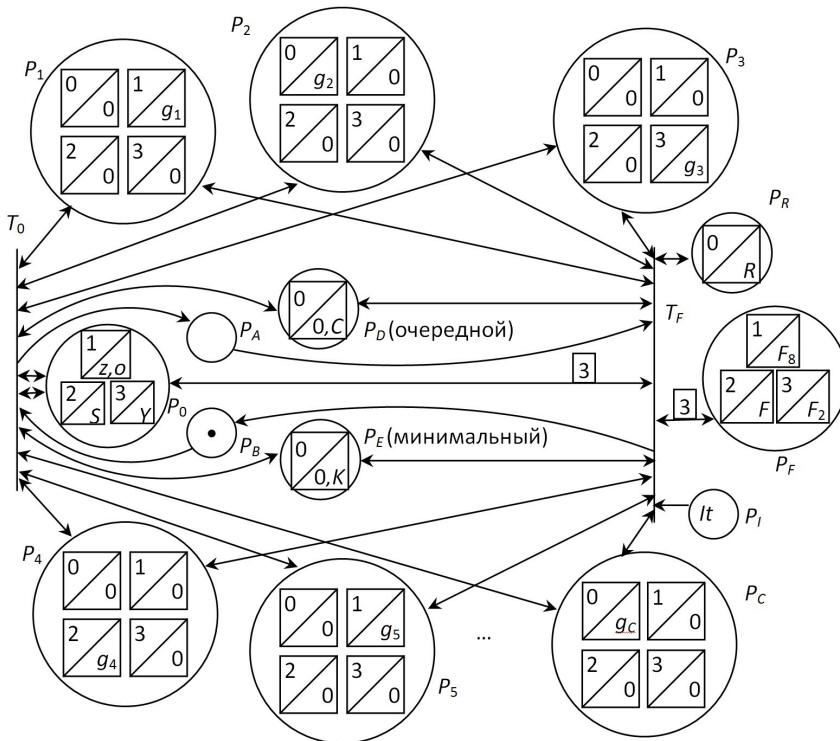


Рис. 3.1. J-сеть, моделирующая процесс оптимизации

Fig. 3.1. J-net modeling the optimization process

а сама разметка $m(x)$ представляет собой набор цветных маркеров в позиции x . Рассмотрим содержимое маркеров по позициям:

- P_0 : маркеры пронумерованы от 1 до 3 и содержат соответственно:
 - 1) z, o – номер этапа и наилучшую (т.е. с наименьшим соответствующим значением F) к данному моменту σ не всегда полной длины (вначале нулевой длины);
 - 2) S – модифицированную σ (вначале нулевой длины);
 - 3) Y – наилучшую к данному моменту σ полной длины.
 - P_F : элементы $m(P_F)$ имеют номера 1–3 и содержат соответственно:
 - 1) F – текущее значение целевой функции F ;
 - 2) F_8 – минимальное из найденных к данному моменту значений F на очередном шаге (при полной или неполной длине σ). Под шагом понимается последовательность действий по реализации распределения нагрузки в соответствии с конкретной σ либо при её первоначальном получении. Вначале F_8 – большое число, которое обозначим символом Ω ;

- 3) F_2 – минимальное из найденных к данному моменту значений F при полной длине σ (вначале – Ω).
- P_R : наилучшее из найденных распределений нагрузки, вначале пустое.
 - P_E : сведения о виде нагрузки (число) и о номере позиции из P_1, \dots, P_C , при которых было достигнуто наименьшее значение для F_8 в процессе реализуемого таким образом поиска. Вначале содержатся пустые значения, что обозначается (\emptyset, \emptyset) .
 - P_D : информация о том, маркер с каким номером и из какой позиции в последний раз изменялся переходом T_0 , чтобы можно было узнать, какой маркер изменять следующим. Вначале содержатся пустые значения, что обозначается (\emptyset, \emptyset) . В дальнейшем изложении соответствующий маркер из соответствующей позиции из P_1, \dots, P_C , что определяется информацией из $m(P_D)$, будем обозначать $M(P_D)$, а соответственно из $m(P_E) - M(P_E)$.

Рассмотрим работу перехода T_F . Её можно разделить на шаги:

- 1) Из каждой позиции P_1, \dots, P_C в T_F отбирается один маркер, в первую очередь несущий, и только если таких нет, любой нуль-маркер. В T_F также отбираются все элементы из $m(P_0)$ и $m(P_F)$, так как кратности дуг равны числу маркеров. Из остальных входных позиций отбирается по одному маркеру. После отбора маркеров переход вычисляет значение целевой функции F на основе пустых и непустых значений (см. блок 09 блок-схемы алгоритма (БСА) (это значение затем временно присваивается маркеру 2 из P_F , однако по окончании срабатывания заменяется на \emptyset , см. ЛД ниже)).
- 2) Затем этот переход сравнивает F и F_8 (см. блок 10 БСА). Если $F < F_8$, то:
 - 2.1) значение F_8 для отобранного маркера номер 1 из P_F заменяется на F (см. блок 11 БСА);
 - 2.2) если этап первый, то O для отобранного маркера номер 1 из P_0 заменяется на S (причем, если длина S на 1 меньше C , то заменяется на S с дописанным номером позиции с временно несущим маркером);
 - 2.3) информация в $m(P_E)$ заменяется на информацию в $m(P_D)$ (в блоке 11 БСА это $i1$ и Min);
 - 2.4) если σ , в том числе на первом этапе при анализе последнего элемента, имеет полную длину (т.е. в каждой позиции P_1, \dots, P_C есть несущий маркер), то этот переход сравнивает F_2 и F_8 с учетом этапа и логики алгоритма имитации отжига. На первом этапе это простое сравнение. На втором этапе при сравнении учитывается также вероятность перехода к состоянию с большим значением целевой функции из состояния с меньшим её значением (см. блок 29 БСА). Такое сравнение для обоих этапов вместо символа «<» обозначим символом « \prec ». То есть, даже если $F_8 > F_2$, но произошло событие для указанной вероятности, то условие « $F_8 \prec F_2$ » считается выполненным. Для построения ЛД это не принципиально. Если $F_8 \prec F_2$, то:
 - 2.4.1) значение F_2 для отобранного маркера номер 3 из P_F заменяется на F_8 (см. блок 31 БСА);

- 2.4.2) значение Y для маркера 3 из P_0 заменяется на O ;
- 2.4.3) если нет отобранных нуль-маркеров из P_1, \dots, P_C , то значение R для $m(P_R)$ заменяется на упорядоченную совокупность значений номеров несущих маркеров из P_1, \dots, P_C .
- 3) Затем, если этап 1, и $M(P_D)$ соответствует последнему виду нагрузки в конце очереди, либо этап 2, и $M(P_D)$ соответствует последнему виду нагрузки, то присваивается $F_8 \leftarrow \Omega$ (см. блок 05 БСА).
- 4) Затем в P_F присваивается $F \leftarrow \emptyset$ (т.к. это временное значение).
- 5) После этого все информационные маркеры возвращаются в свои позиции, а в P_B помещается обычный маркер.

Под очередью далее понимается либо простая очередь (на 1 этапе), либо последовательность O для 2 этапа. В очередь не включаются маркеры из позиций, содержащих несущие маркеры на ВПО.

Переход T_0 работает следующим образом.

В соответствии с дугами, в него отбираются $m(P_D)$, маркеры с номерами 1, 2 из P_0 , а также $m(P_B)$ и $m(P_E)$. Далее возможны четыре ситуации, при которых из каждой другой (не указанной в ситуации) позиции P_1, \dots, P_C отбирается в первую очередь несущий маркер, и только если таковых нет, любой нуль-маркер. Действия по ситуациям следующие:

- 1) Если этап 1, и $m(P_D)$ указывал на последний маркер в очереди, то это значит, что очередь закончилась и нужно установить какому-то нуль-маркеру значение и увеличить длину S . Тогда:
 - 1.1) отбирается $M(P_D)$ и назначается нуль-маркером;
 - 1.2) отбирается $M(P_E)$ и назначается несущим на ВПО (в блоке 12 БСА это операция $O_{i1} \rightarrow Min$);
 - 1.3) к S (у маркера номер 2 в P_0) дописывается номер позиции для $M(P_E)$ (см. блок 12 БСА);
 - 1.4) в $m(P_D)$ заносится информация о расположении первого по очереди маркера, причем этот маркер становится несущим (в БСА это значит $i := 1, j := 0$ для блока 08);
 - 1.5) в $m(P_E)$ заносится (\emptyset, \emptyset) .
- 2) Если этап 2, и не производится установка номера этапа 2, и не достигнут конец O , то если рассмотрен последний элемент нагрузки для P_i (i определяется из маркера в P_D), тогда:
 - 2.1) отбирается $M(P_E)$ – это маркер из P_i , и назначается несущим на ВПО;
 - 2.2) в P_D вносятся данные о следующем элементе O (для первого вида нагрузки), и $M(P_D)$ становится временно несущим.

- 3) Когда длина S достигнет C (это проверяется у маркера 2 из P_0 и является случаем, совместным со случаем 1), а этап 1 (т.е. $k = C$ в блоке 04 БСА), это будет означать, что первый этап завершен, и переход T_0 устанавливает номер этапа у маркера номер 1 из P_0 в 2. При этом, а также на 2 этапе при достижении конца O , он:
- 3.1) производит случайное изменение в O у маркера номер 1 из P_0 , а если O пустое, то вместо этого копирует его из S (см. блоки 41, 42 БСА);
 - 3.2) вносит в P_D данные о первом элементе O ;
 - 3.3) в $m(P_E)$ заносится (\emptyset, \emptyset) ;
 - 3.4) все несущие маркеры, в т.ч. ВПО, делает нуль-маркерами, а маркер, определяемый информацией из P_D , несущим.
- 4) В остальных случаях:
- 4.1) прошлый $M(P_D)$ становится нуль-маркером;
 - 4.2) очередь продвигается, и маркер в P_D начинает указывать на очередной элемент;
 - 4.3) отбирается $M(P_D)$ и становится несущим.

Затем все информационные маркеры возвращаются в свои позиции, а обычный маркер помещается в P_A . Срабатывание переходов T_0 и T_F чередуется, что обеспечивается последовательным переносом обычного маркера из позиции P_A в P_B и обратно. Этим чередованием обеспечивается вычисление целевой функции в ответ на каждое изменение маркировки переходом T_0 , в частности, при присвоении нуль-маркеру значения.

Работа сети продолжается, пока не закончатся маркеры в позиции P_I . Первоначально их там столько, какова величина, обозначаемая It , т.е., сколько должно быть сделано итераций. В итоге наилучшее из найденных значений F будет находиться в маркере 3 из позиции P_F , а в $m(P_R)$ – наилучшее из найденных распределений нагрузки.

4. Лента достижимости

Для сети построена ЛД. Она не может использоваться для решения задачи минимизации, поскольку не предназначена для этого, однако применяется для проверки правильности модели. ЛД, в частности, для минимизации своего размера, наделена следующими особенностями:

- $C = 2$, $It = 45$ (чтобы достичь всех возможных состояний с учетом дублирования);
- количество ячеек в таблицах макропозиций P_1, \dots, P_C равно двум, а не четырём, т.е. не учитывается почасовая оплата и прочая нагрузка;
- значения функции не числовые, а в виде $F(O_1, O_2)$, где O_1 – вид нагрузки для первого элемента нагрузки, O_2 – соответственно, для второго;
- для дублирующих M_τ может быть разным: а) количество маркеров в P_I , б) $F(O_1, O_2)$, т.е. значение маркера номер 2, в P_F , в) значение S на этапе 2, так как S на этапе 2 не может быть изменено и не используется;

- номера информационных маркеров для краткости не записываются явно, а определяются записанным порядком.

ЛД построена вручную, поскольку при её построении выявлялись и исправлялись ошибки в логике работы сети, что удобнее выполнять при внимательном рассмотрении каждой группы M_τ . Она описывает 41 M_τ на этапе 1 и 228 на втором этапе, итого 269 M_τ , в том числе 122 M_τ , в которых имеется результат успешной проверки условия переходом T_F и 72 M_τ , полученные в результате случайного изменения или не изменения (из-за фактора случайности) последовательности переходом T_0 . В ЛД 71 M_τ отмечены как дублирующие, т.е. после них варианты результатов при срабатывании переходов не рассматривались.

Рассмотрим фрагменты данной ленты (Табл. 4.1, 4.2).

Таблица 4.1. Лента достижимости (первый фрагмент)

Table 4.1. Reachability tape (first fragment)

Прочее				Маркеры в позициях									
Сравнение F и F_8	Маркировка	Возбуждён	Сработал из маркировки	$P_0 - z, o, S, Y$	P_1	P_C	P_A	P_B	P_D	P_E	$P_F - F_8, F, F_2$	P_I	P_R
Other				Tokens in places									
Comparison F and F_8	Marking	Enabled	Fired from marking	$P_0 - z, o, S, Y$	P_1	P_C	P_A	P_B	P_D	P_E	$P_F - F_8, F, F_2$	P_I	P_R
	M_0	T_0	—	1, \emptyset , \emptyset, \emptyset	0, 0		—	1	\emptyset, \emptyset	\emptyset, \emptyset	$\Omega, \emptyset, \emptyset, \Omega$	45	\emptyset
—	M_1	T_F	M_0T_0	1, \emptyset , \emptyset, \emptyset	$g_1, 0, 0$		1	—	0, 1	\emptyset, \emptyset	$\Omega, \emptyset, \emptyset, \Omega$	45	\emptyset
—	M_2	T_0	M_1T_F	1, \emptyset , \emptyset, \emptyset	$g_1, 0, 0$		—	1	0, 1	0, 1	$(0, \emptyset), \emptyset, \Omega$	44	\emptyset
—	M_3	T_F	M_2T_0	1, \emptyset , \emptyset, \emptyset	0, g_1	0, 0	1	—	1, 1	0, 1	$(0, \emptyset), \emptyset, \Omega$	44	\emptyset
$(1, \emptyset) \geq (0, \emptyset)$	$M_4; F \geq F_8$	T_0	M_3T_F	1, \emptyset , \emptyset, \emptyset	0, g_1	0, 0	—	1	1, 1	0, 1	$(0, \emptyset), \emptyset, \Omega$	43	\emptyset
$(1, \emptyset) < (0, \emptyset)$	$M_4^1; F < F_8$	T_0	M_3T_F	1, \emptyset , \emptyset, \emptyset	0, g_1	0, 0	—	1	1, 1	1, 1	$(1, \emptyset), \emptyset, \Omega$	43	\emptyset

В Таблицах 4.1, 4.2 показаны некоторые характерные ситуации, достижимые сетью Петри:

- 1) Начальная маркировка M_0 и несколько последующих.
- 2) Последняя из рассматриваемых $M_\tau - M_{21}^{55}$.

Таблица 4.2. Лента достижимости (второй фрагмент)

Table 4.2. Reachability tape (second fragment)

Прочее				Маркеры в позициях									
Сравнение F и F_8	Маркировка	Возбуждён	Сработал из маркировки	$P_0 - z, o, S, Y$	P_1	P_C	P_A	P_B	P_D	P_E	$P_F - F_8, F, F_2$	P_I	P_R
Other				Tokens in places									
Comparison F and F_8	Marking	Enabled	Fired from marking	$P_0 - z, o, S, Y$	P_1	P_C	P_A	P_B	P_D	P_E	$P_F - F_8, F, F_2$	P_I	P_R
(1, 1) < (1, 0)	$M_{20}^{15}; F < F_8$	T_0	$M_{19}^{15}T_F$	2, 1_2, 1_2, 1_2	0, g1	0, g_2	—	1	1, 2	1, 2	$\Omega, \emptyset, (1,1)$	35	1, 1
(0, 1) < (0, 0); (0, 1) < (1, 0)	$M_{20}^{34} \equiv M_{20}^{32}; F < F_8; F_8 \prec F_2$	T_0	$M_{19}^6T_F$	2, 1_2, 2_1, 1_2	g1 , 0	0, g_2	—	1	1, 2	1, 2	$\Omega, \emptyset, (0,1)$	35	0, 1
не изменяется	$M_{21}^{27} \equiv M_{21}^{37}$	T_F	$M_{20}^{27}T_0$	2, 1_2, 2_1, 2_1	g_1 , 0	0, 0	1	—	0, 1	\emptyset	$\Omega, \emptyset, (0,1)$	35	0, 1
изменяется	$M_{21}^{55} \equiv M_{21}^9$	T_F	$M_{20}^{27}T_0$	2, 2_1, 2_1, 2_1	0, 0	g_2 , 0	1	—	0, 2	\emptyset	$\Omega, \emptyset, (0,1)$	35	0, 1

- 3) M_τ с результатом успешной проверки условия переходом T_F .
- 4) M_τ , полученные в результате изменения или не изменения последовательности переходом T_0 .
- 5) Дублирующие M_τ .
- 6) Одна из M_τ с объединением условий – M_{20}^{34} .
- 7) M_τ с несущими маркерами на ВПО (выделены жирным шрифтом).

Символ \emptyset означает отсутствие вида нагрузки, либо отсутствие последовательности, либо отсутствие распределения, в зависимости от столбца, где встречается этот символ.

Ячейки таблиц, содержащие символ тождественного равенства \equiv , содержат дублирующие M_τ . Жирным шрифтом в первом столбце выделены условия, которые выполняются, а противоположные этим условия выполняться не могут в связи с историей работы сети, поэтому соответствующие M_τ не включены в ЛД.

Символ F перед скобками для краткости представления в статье не записывался, однако он в действительности есть. Например, выражение « $(0, 1) < (0, 0)$ » означает « $F(0, 1) < F(0, 0)$ ». По той же причине не использованы символы круглых скобок для представления наборов значений в столбцах 5, 6, 7, 10, 11, не показан столбец с номерами, а в столбце $P_F \dots$ не показаны промежуточные состояния при срабатывании перехода.

5. Отбраковка недостижимых состояний с помощью систем неравенств

В модели нет информации о том, какая из M_τ обладает большим или меньшим значением F , поэтому рассматриваются все варианты, возможные в результате сравнений этих значений. Например, неизвестно, что меньше: $F(0, 1)$ или $F(1, \emptyset)$. Здесь $F(0, 1)$ означает значение целевой функции при выборе для первого элемента нагрузки ИП по основной ставке, а для второго – по совместительству; $F(1, \emptyset)$ – соответственно для второго – не выбрано, а для первого – по совместительству (это иллюстрирует выше-приведённую фразу «на основе пустых и непустых значений»). Поэтому рассматриваются два результата сравнения – $F(0, 1)$ и $F(1, \emptyset)$. Для корректности ЛД необходимо учитывать, какое из значений принималось меньшим ранее, если производилось подобное сравнение. Соответственно, невозможный результат сравнения не рассматривается. Такие результаты сравнения в совокупности формируют историю сравнений.

В первом столбце ЛД пронумерованы только строки с неравенствами. Итоговая история сравнений описывает 88 систем неравенств. В процессе проектирования J-сети была рассмотрена также несколько иная логика работы сети (регулярно не устанавливались пустые значения), тогда число неравенств в системе достигало 21, а количество M_τ – 435, в связи с чем появилась необходимость автоматизации анализа таких систем. С целью такого анализа разработано программное обеспечение на C#, которое на основе известных соотношений (меньше или не меньше) определяет некоторые из неизвестных соотношений. Здесь используются четыре очевидные правила – свойства транзитивности:

$$(A \geq B) \& (B \geq C) \rightarrow (A \geq C); (A \geq B) \& (B > C) \rightarrow (A > C); \\ (A > B) \& (B > C) \rightarrow (A > C); (A > B) \& (B \geq C) \rightarrow (A > C).$$

Логика программы реализуется с помощью двух вложенных циклов, поэтому лежащий в её основе алгоритм определения соотношений имеет временную сложность $O(n^3)$, где n – число обозначений значений F (таких как $F(0, 1)$, $F(1, \emptyset)$ и т.п.). В литературе (см., например, [19], [20]) описываются различные подходы к решению систем неравенств, однако они более сложные для понимания разработки реализующего их программного обеспечения.

Программа, реализующая разработанный алгоритм определения соотношений, применена для решения систем неравенств в различных проектах J-сети на этапах её проектирования.

6. Анализ ленты достижимости

Рассмотрим ЛД. К первому этапу относятся 57 M_τ , остальные ко второму, и их значительно больше. Для всех M_τ на втором этапе искомое распределение нагрузки непустое, в отличие от ЛД для первоначального проекта сети, содержащего просчёты в том, что нагрузка не всегда полностью распределялась по ИП.

Переход T_F срабатывает до 10 раз, после чего M_τ становятся только дублирующими и не рассматриваются. Количество рассматриваемых M_τ ЛД по шагам: 1, 2 – 1, 3 – 2, 4 – 4 (3), 5 – 6 (4), 6 – 4, 7 – 8 (16), 8 – 16, 9 – 20, 10 – 26 (22), 11 – 36 (56), где в скобках указано число M_τ после срабатывания T_0 , если отличается от первого числа (после срабатывания T_F).

В ЛД нет бесконечного накопления маркеров, что говорит об отсутствии соответствующей ошибки в модели. Со временем число маркеров в позиции P_I уменьшается, в конце работы до 0, после чего возбужденных переходов не будет, и сеть перестает работать. Это моделирует завершение работы программы по достижении установленного числа итераций It . В построенной ЛД встречаются все варианты последовательности отбора. Поскольку в последовательности только два элемента, ЛД может считаться относительно короткой. Если бы там было, например, 5 элементов, то если число итераций было бы неограниченным, эта лента была бы не только в $5!/2! = 60$ раз протяжённее из-за перестановок в последовательности, но и еще в $3^5/3^2 = 27$ раз длиннее из-за увеличения числа комбинаций нахождения маркеров в 5 позициях вместо 2 (итого, по крайней мере, в 1620 раз). Для обычных сетей Петри ЛД при небольшом числе узлов не так велика, но в случае J-сети размер многократно возрастает из-за большого числа разных комбинаций значений маркеров.

Таким образом, при незначительном увеличении C ЛД получится во много раз длиннее, и если It небольшое, все ситуации охвачены не будут. Однако при $C = 2$ построенная ЛД показывает корректность работы алгоритма оптимизации.

J-сети являются относительно редко упоминаемой в литературе разновидностью аппарата сетей Петри, и в свободном доступе отсутствуют сведения о построении для подобной сети ЛД. Таким образом, впервые построена лента достижимости для J-сети. Анализ работы алгоритма оптимизации, проведенный посредством J-сети с помощью ЛД, показывает корректность его работы.

СПИСОК ЛИТЕРАТУРЫ

1. McEneaney J. E., Morsink P. Curriculum Modelling and Learner Simulation as a Tool in Curriculum (Re)Design. *Journal of Learning Analytics*. 2022. Vol. 9, No 2. P. 161–178. DOI: 10.18608/jla.2022.7499
2. Nyéki L. Modeling of Higher Education Processes using Colored Petri Nets. *XXVIII Multimedia in Education International Conference*. 2022. P. 62–68. Режим доступа: https://www.researchgate.net/publication/377218660_Modeling_of_Higher_Education_Processes_using_Colored_Petri_Nets
3. Castaldi P., Mimmo N. Representing the dynamics of student learning and interactions with a university curriculum. *IFAC-PapersOnLine*. 2024. Vol. 58, no. 16. P. 211–216. DOI: 10.1016/j.ifacol.2024.08.488

4. Vanit-Anunchai S. Teaching Low-Code Formal Methods with Coloured Petri Nets. *Formal Methods Teaching Workshop*. Springer. 2023. P. 96–104. DOI: 10.1007/978-3-031-27534-0
5. Rong J. A Practical Exploration of the Integration of Civics into Student Education and Management in Colleges and Universities Empowered by Information Integration Technology in the Context of Three-Whole Parenting. *Applied Mathematics and Nonlinear Sciences*. 2024. Vol. 9, no. 1. P. 1–18. Режим доступа: <https://sciendo.com/article/10.2478/amns.2023.2.01398>
6. Huang L., Zheng J., Lajoie S. P., Chen Y., Hmelo Silver C. E., Wang M. Examining university teachers' self-regulation in using a learning analytics dashboard for online collaboration. *Education and Information Technologies*. 2024. Vol. 29. P. 8523–8547. DOI: 10.1007/s10639-023-12131-7
7. Nyeki L. The Application of Petri Nets in Modeling the Educational Process // X. Ágoston Trefort Conference. Budapest. 2020. P. 202–213. Режим доступа: <https://www.researchgate.net/publication/377218178>
8. Юдицкий С. А. Сценарный подход к логическому моделированию систем рыночной экономики // Системы управления, связи и безопасности. 2015. № 2. С. 147–164. Режим доступа: <https://scgs.intelgr.com/archive/2015-02/06-Iuditskii.pdf>
9. Желтов П. В. Модели поиска и копирования символьных данных на J-сетях // Прикладная информатика. 2012. Т.40, № 4. С. 81–87. Режим доступа: http://www.appliedinformatics.ru/r/articles/article/index.php?article_id_4=1312
10. Димитриев А. П., Лавина Т. А. Применение алгоритма оптимизации последовательности отбора для распределения учебной нагрузки преподавателя по индивидуальным планам // Современные научоемкие технологии. 2024. № 4. С. 15–20. DOI: 10.17513/snt.399613
11. Виноградов Г. П., Кирсанова Н. В. Модель интерактивного планирования нагрузки ППС кафедры // Вестник Тверского государственного технического университета. 2017. Вып. 32. С. 106–111. Режим доступа: <https://core.ac.uk/reader/151241237>
12. Ивахненко Д. А. Применение моделей двусторонних рынков в задаче распределения учебной нагрузки между преподавателями кафедры // Современная экономика: проблемы и решения. 2021. Т. 141, № 9. С. 16–28. DOI: <https://doi.org/10.17308/meps.2021.9/2667>
13. Султанова С. Н., Тархов С. В. Модели и алгоритмы поддержки принятия решений при распределении учебной нагрузки преподавателей // Вестник Уфимского государственного авиационного технического университета. 2006. Т. 7, № 3 (16). С. 107–114.
14. Димитриев А. П., Лавина Т. А. Алгоритм распределения учебной нагрузки преподавателя по индивидуальным планам с применением технологии искусственного интеллекта // Современные научоемкие технологии. 2023. № 5. С. 13–18. DOI: 10.17513/snt.39610

15. Whitley D. A genetic algorithm tutorial. *Statistics and Computing*. 1994. Vol. 4. P. 65–85. DOI: 10.1007/BF00175354
16. Dimitriev A. P., Bazhenov R. I. Time indicators of effective optimization algorithms in group load control modeling // IOP Conference Series: Materials Science and Engineering (MSE). 2021. Vol. 1019. 012038. DOI:10.1088/1757-899X/1019/1/012038
17. Kirkpatrick S., Gelatt C. D., Vecchi M. P. Optimization by Simulated Annealing. // *Science*. 1983. Vol. 220, No 4598. P. 671–680. DOI: 10.1126/science.220.4598.671
18. Котов В. Е. Сети Петри. М.: Наука. Главная редакция физико-математической литературы, 1984. 160 с.
19. Евтушенко Ю. Г., Посыпкин М. А., Рыбак Л. А., Туркин А. В. Отыскание множеств решений систем нелинейных неравенств // *Журнал вычислительной математики и математической физики*. 2017. Т. 57, № 8. С. 1248–1254. DOI: 10.7868/S0044466917080075
20. Лапиков И. И., Никонов В. Г. Адаптивный алгоритм решения систем неравенств с k-значными неизвестными // Труды Военно-космической академии имени А.Ф.Можайского. 2016. № 650. С. 88–94.

*Поступила 10.02.2025; доработана после рецензирования 22.10.2025;
принята к публикации 26.11.2025*

Авторы прочитали и одобрили окончательный вариант рукописи.

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

REFERENCES

1. J. E. McEneaney, P. Morsink, “Curriculum Modelling and Learner Simulation as a Tool in Curriculum (Re)Design”, *J. Learn. Anal.*, **9**:2 (2022), 161-178. DOI: 10.18608/jla.2022.7499
2. L. Nyéki, “Modeling of Higher Education Processes using Colored Petri Nets”, *XXVIII Multimedia in Education International Conference*, 2022, 62–68, Available at: https://www.researchgate.net/publication/377218660_Modeling_of_Higher_Education_Processes_using_Colored_Petri_Nets.
3. P. Castaldi, N. Mimmo, “Representing the dynamics of student learning and interactions with a university curriculum”, *IFAC-PapersOnLine*, **58**:16 (2024). DOI: 10.1016/j.ifacol.2024.08.488
4. S. Vanit-Anunchai, “Teaching Low-Code Formal Methods with Coloured Petri Nets”, *Formal Methods Teaching Workshop*, Springer, 2023, 96-104 DOI: 10.1007/978-3-031-27534-0.
5. J. Rong, “A Practical Exploration of the Integration of Civics into Student Education and Management in Colleges and Universities Empowered by Information Integration Technology in the Context of Three-Whole Parenting”, *Applied Mathematics and Nonlinear Sciences*, **9**:1 (2024), 1–18, Available at: <https://sciendo.com/article/10.2478/amns.2023.2.01398>.

6. L. Huang, J. Zheng, S. P. Lajoie, Y. Chen, C. E. Hmelo Silver, M. Wang, "Examining university teachers' self-regulation in using a learning analytics dashboard for online collaboration", *Educ Inf Technol*, **29** (2024), 8523–8547. DOI: 10.1007/s10639-023-12131-7
7. L. Nyeki, "The Application of Petri Nets in Modeling the Educational Process", *X. Ágoston Trefort Conference.*, Budapest, 2020, 202–213, Available at: <https://www.researchgate.net/publication/377218178>.
8. S. A. Iuditskii, "The Scenary Approach to the Logical Modeling of Systems of a Market Economy", *Sistemy upravleniya, svyazi i bezopasnosti*, 2015, no. 2, 147—164 (In Russ.), Available at: <https://scgs.intelgr.com/archive/2015-02/06-Iuditskii.pdf>.
9. P. V. Zheltov, "[Models for searching and copying symbol data on J-nets]", *Prikladnaya informatika*, **4**:40 (2012), 81–87 (In Russ.).
10. A. P. Dimitriev, T. A. Lavina, "Application of the selection sequence optimization algorithm to distribute the teacher's educational load according to individual plans", *Sovremennye naukoemkie tekhnologii*, 2024, no. 4, 15–20 (In Russ.). DOI: 10.17513/snt.39967
11. G. P. Vinogradov, N. V. Kirsanova, "[Interactive model planning of the load between the staff of the chairs]", *Vestnik Tverskogo gosudarstvennogo tekhnicheskogo universiteta*, 2017, no. 32, 106–111 (In Russ.), Available at: <https://core.ac.uk/reader/151241237>.
12. D. A. Ivakhnenko, "[Application of the twosided markets theory for university department teaching workload distribution]", *Sovremennaya ekonomika: problemy i resheniya*, **9**:141 (2021), 16–28 (In Russ.). DOI: 10.17308/meps.2021.9/2667
13. S. N. Sultanova, S. V. Tarkhov, "[Models and algorithms of desizion support at teachers' educational load distribution]", *Vestnik Ufimskogo gosudarstvennogo aviationsionnogo tekhnicheskogo universiteta*, **7**:3 (16) (2006), 107–114 (In Russ.), Available at: <https://elibrary.ru/item.asp?id=9289661>.
14. A. P. Dimitriev, T. A. Lavina, "Algorithm for distributing the teacher's educational load according to individual plans using artificial intelligence technology", *Sovremennye naukoemkie tekhnologii*, 2023, no. 5, 13–18 (In Russ.). DOI: <https://doi.org/10.17513/snt.39610>
15. D. Whitley, "A genetic algorithm tutorial", *Stat Comput*, **4** (1994), 65–85. DOI: 10.1007/BF00175354
16. A. P. Dimitriev, R. I. Bazhenov, "Time indicators of effective optimization algorithms in group load control modeling", *IOP Conf. Ser.: Mater. Sci. Eng.*, **1019** (2021), 012038. DOI: 10.1088/1757-899X/1019/1/012038
17. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, **220**:4598 (1983), 671–680. DOI: 10.1126/science.220.4598.671
18. V. E. Kotov, *[Petri Nets]*, Nauka Publ., Moscow, 1984 (In Russ.), 160 p.

19. Yu. G. Evtushenko, M. A. Posypkin, L. A. Rybak, A. V. Turkin, “Finding sets of solutions to systems of nonlinear inequalities”, *Zhurnal vychislitel'noy matematiki i matematicheskoy fiziki*, **57**:8 (2017), 1248–1254 (In Russ.). DOI: 10.7868/S0044466917080075
20. I. I. Lapikov, V. G. Nikonov, “[Adaptive algorithm for solving systems of inequalities with k-valued unknowns]”, *Trudy Voenno-kosmicheskoy akademii imeni A.F.Mozhayskogo*, 2016, no. 650, 88–94 (In Russ.).

Submitted 10.02.2025; Revised 22.10.2025; Accepted 26.11.2025

The authors have read and approved the final manuscript.

Conflict of interest: The authors declare no conflict of interest.