

УДК 681.513.5

Алгоритм управляемой имитации сети Петри

А.Н. Сочнев*

Сибирский федеральный университет,
660041 Россия, Красноярск, пр. Свободный, 79¹

Received 3.09.2010, received in revised form 10.09.2010, accepted 17.09.2010

В данной статье приводится обобщенный алгоритм выполнения операций по управлению процессом смены состояний сети Петри. Моделью устройства управления для сети Петри выступает другая сеть Петри или предварительно обученная нейросеть. Использование нейросети в ходе имитационного эксперимента с сетью Петри позволяет экстремизировать требуемый критерий оптимальности процесса.

Ключевые слова: сеть Петри, нейросеть, сетевой контроллер.

1. Общие принципы сетевого моделирования

Эффективным методом исследования и оптимизации сложных технических и производственных систем служит имитация процесса их работы на основе предварительно сформированных моделей. Известны различные математические средства модельного представления систем, каждое из которых имеет свои сферы применения, достоинства и недостатки. Для моделирования дискретных производственных систем широко применяется математический аппарат сетей Петри [1]. Среди основных преимуществ сетей Петри можно отметить то, что они позволяют с требуемой точностью представлять ветвящиеся, параллельные, циклические процессы, обладают средствами анализа, а также моделирования в реальном масштабе времени.

Процесс моделирования можно условно разделить на две стадии.

1. Формирование структуры и параметров модели на основе свойств исходной системы (объекта управления).
2. Имитационные эксперименты для достижения требуемого значения показателя качества работы системы.

Качество работы системы оценивается по численному значению выбранного критерия оптимальности. Основным средством оптимизации в сетевых моделях являются правила приоритета, которые выбирают из множества разрешенных операций одну по заранее определенному принципу. Следует заметить, что при использовании приоритетов невозможно сказать о том, что получено наилучшее решение из множества возможных. Доказано только то, что при применении приоритетных правил значение критерия оптимальности получается лучше, чем без их использования.

* Corresponding author E-mail address: lesek@mail.ru

¹ © Siberian Federal University. All rights reserved

Сеть Петри формально описывается набором вида:

$$R = \{P, T, D, \mu_0\},$$

где $P = \{p_i\}$ – непустое конечное множество позиций, $i = \overline{1, n}$; $T = \{t_j\}$ – непустое конечное множество переходов, $j = \overline{1, m}$; $D = D^+ - D^-$ – отношение инцидентности позиций и переходов; $\mu_0 : P \rightarrow R^+$ – начальная маркировка сети; R^+ – множество целых неотрицательных чисел; n – количество позиций; m – количество переходов [2].

Матрица инцидентий позволяет определить уравнение, формирующее механизм изменения маркировки сети:

$$\mu[k+1] = \mu[k] + D \cdot u[k],$$

где $u[k]$ – вектор-столбец длины m , имеющий единственный ненулевой элемент в позиции j , равный 1 и, соответственно, определяющий, какой из переходов срабатывает на текущем такте управления.

Условие срабатывания переходов сети имеет следующий вид:

$$D^- \cdot u[k] \leq \mu[k].$$

2. Элементы структуры сетевой модели

Для придания имитационной модели (а следовательно, и объекту) желаемых свойств предлагается наряду с моделированием объекта управления (ОУ) синтезировать модель системы управления, добавив управляющее устройство (УУ). Сетевая модель разделяется на две составляющие: собственно модель объекта управления, имитирующая поведение ОУ, и модель управляющего устройства – сетевой контроллер (СК). Модель устройства управления дополняется структурными блоками, способными реализовывать сложные задачи управления с учетом различных критериев оптимальности, а также в условиях действия возмущений.

Под моделью объекта управления понимается сетевой эквивалент объекта, т.е. модель в традиционном понимании. Модель устройства управления включает в себя в минимально необходимом наборе следующие блоки:

- 1) блок проверки условий срабатывания переходов;
- 2) блок приоритетной выборки переходов.

Очевидно, что наличие первого из них обязательно. Второй используется при желании оптимизировать процессы в моделируемой системе. Вектор управления, формируемый моделью устройства управления должен указывать, номера переходов сети, активируемых на каждом такте моделирования.

В зависимости от характера задач управления сетевую модель устройства управления предлагается дополнить блоками на основе **сетей Петри** и **нейросетей**.

Сеть Петри предлагается использовать для решения следующих задач:

- обеспечение принадлежности вектора состояния модели множеству допустимых значений;
- обеспечение требуемой степени инвариантности модели к возмущениям и ограниченной оптимизации процессов в моделируемой системе.

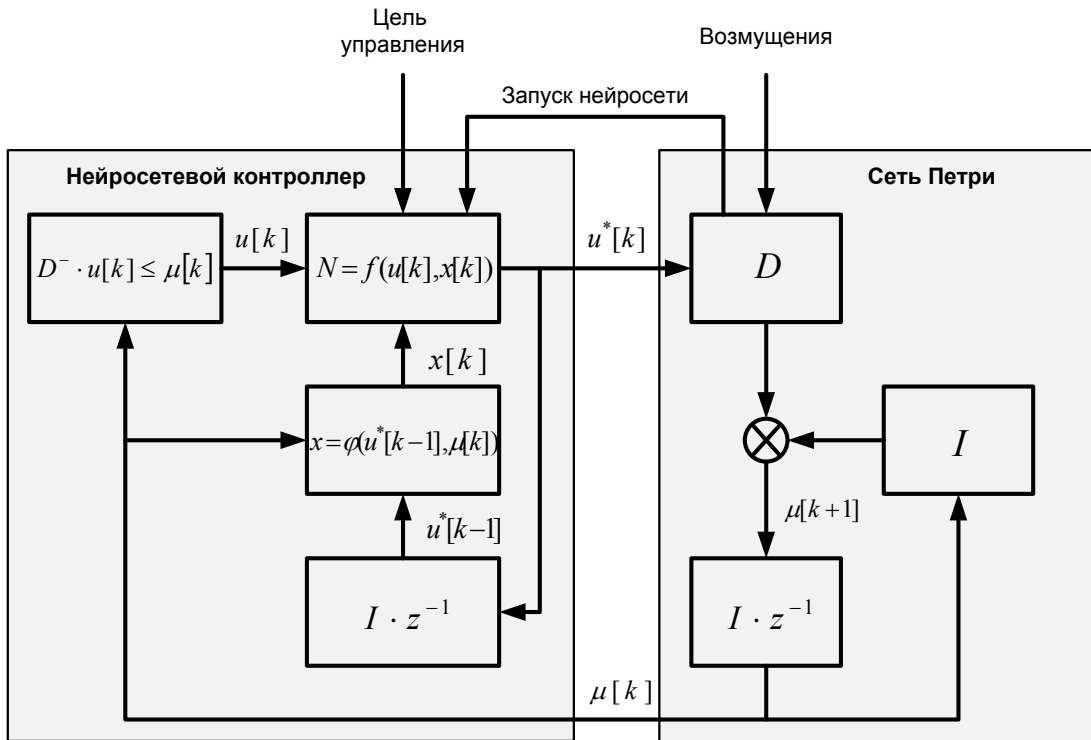


Рис. 1. Структура системы оптимального управления с нейросетевым контроллером

Нейросеть используется в тех случаях, когда требуется реализация функций искусственного интеллекта, т.е. для создания систем интеллектуального управления [3, 4]. Ниже рассматривается вопрос применения нейросетей.

Рассмотрим задачу оптимизации процессов в системе по заданному критерию. В качестве нового метода оптимизации предлагается предварительно создавать в составе устройства управления **нейросетевой контроллер**, использующий количественные характеристики качества функционирования системы (рис. 1). При использовании предлагаемого метода выбор активируемого перехода будет производиться не правилом приоритета, а сформированным нейросетевым контроллером.

Классическая схема выбора активируемых переходов сети имеет свои недостатки. Основной из них заключается в том, что для выработки управляющего воздействия на сеть не используется информация о текущем состоянии элементов модели, в то время как для оптимального управления они необходимы. Для того чтобы получать эти данные, в модели системы управления формируется обратная связь о состоянии качественных характеристик, влияющих на значение критерия оптимальности.

Перечислим основные характеристики.

1. Количество срабатываний каждого перехода

$$\alpha_j[k] = \alpha_j[k-1] + 1, \text{ если } u_j^*[k] = 1, j = 1, m.$$

2. Время суммарной активности каждого перехода

$$\beta_j[k] = \beta_j[k-1] + \delta_j, \text{ если } u_j^*[k] = 1, j = 1, m.$$

где δ_j – время выполнения j -го перехода.

3. Коэффициент занятости каждого перехода

$$\gamma_j[k] = \frac{\beta_j[k]}{\tau}, \quad j = 1, m,$$

где τ – модельное время.

Далее под качественным состоянием модели объекта управления будем понимать вектор, содержащий перечисленные выше характеристики:

$$x[k] = (\alpha[k], \beta[k], \gamma[k]).$$

Нейросетевой контроллер должен реализовывать некоторую функцию

$$u^*[k] = f(u[k], x[k]).$$

Основной задачей при создании подобной системы управления является обеспечение взаимодействия сети Петри и нейросети. Указанная задача усложняется тем, что они функционируют по-разному и предназначены для разных областей применения.

Можно выделить следующие основные принципы использования нейросети в ходе имитационного процесса сети Петри.

1. Каждому переходу сети приписывается определенная операция (или последовательность операций), которая должна выполняться при его срабатывании. Нейросеть представляется подобным образом как операция, соответствующая одному переходу сети. Запуск нейросети осуществляется при срабатывании указанного перехода (рис. 2).
2. Состояние сети определяется не только текущей маркировкой позиций, но и вектором состояний переходов.

Вектор наблюдения сети Петри представлен элементами вектора x , т.е. те элементы, которые его составляют, являются наблюдаемыми параметрами модели объекта управления.

Вектор управления сети Петри u представлен выходными значениями нейросети, воздействующими на сеть Петри.

Алгоритм совместного функционирования нейросети и сети Петри можно определить такой последовательностью действий:

1. Установление активных переходов сети Петри. Проверка условия срабатывания.

$$D^- \cdot u[k] \leq \mu[k].$$

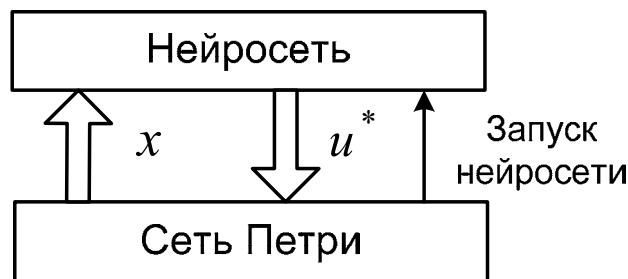


Рис. 2. Информационные потоки между сетью Петри и нейросетью

2. Проверка наличия среди активных переходов таких, которые содержат нейросети. В случае, если такие есть, то запуск нейросетей. Формирование выходного вектора нейросетей y .

3. Бинаризация значений выходного вектора y

4. Пересчет вектора активации переходов

$$u_j^*[k] = u_j[k] \cdot y_j[k], \quad j = 1, m.$$

5. Удаление маркеров из входных позиций активных переходов

$$\mu[k+1] = \mu[k] - D^- \cdot u^*[k].$$

6. Добавление в выходные позиции переходов маркеров

$$\mu[k+1] = \mu[k] + D^+ \cdot u^*[k].$$

7. Расчет элементов вектора состояния переходов

8. Пересчет модельного времени

$$\tau = \tau + \Delta \tau.$$

9. Если $\tau < \tau_{\max}$, то переход к пункту 2, в противном случае прекращение имитационного эксперимента.

3. Пример формирования управляющей нейросети

В качестве объекта исследования рассматривается Учебно-исследовательский гибкий производственный комплекс (УИ ГПК), входящий в структуру Политехнического института ФГОУ ВПО «Сибирский федеральный университет» и предназначенный для обучения студентов элементам CAD/CAM-технологий, методам автоматизированного управления и программирования устройств числового программного управления (ЧПУ) оборудования. В состав УИ ГПК входят: автоматизированный складской комплекс РСК-250, два токарно-винторезных станка 16К20Ф3 с системой ЧПУ 2Р22, токарный станок Наас TL-1 и фрезерный станок Наас ТМ-1.

В структуре учебно-исследовательского роботизированного технологического комплекса можно выделить две производственные ячейки (рис. 3).

Для составления плана-графика производственного процесса задается номенклатура деталей, производимых УИ ГПК. Предусматривается обработка деталей тридцати типов. Предполагается, что тара с заготовками присутствует в роботизированном складском комплексе. Готовые изделия также направляются в ячейки склада. Приоритет всех изделий установлен одинаковым.

Для составления суточного плана-графика необходимо определить все параметры основного и вспомогательного производственных процессов (табл. 1).

Представим описанную систему имитационной моделью на основе сети Петри. Формируемая модель будет отображать движение продуктов производства от входа системы к выходу. Исходными данными для ее формирования служат типовые технологические процессы токарной, фрезерной и токарно-фрезерной обработки. Модель является иерархической и имеет два уровня детализации [1]. Первый уровень отображает процесс перемещения объектов производства в контексте всей системы в целом, и простые элементы модели на этом уровне – производственные ячейки (рис. 4).

Второй уровень детализации отображает процесс перемещения объектов производства в каждой производственной ячейке. Модели содержат технологические операции и операции перемещения объектов производства (рис. 5, 6).

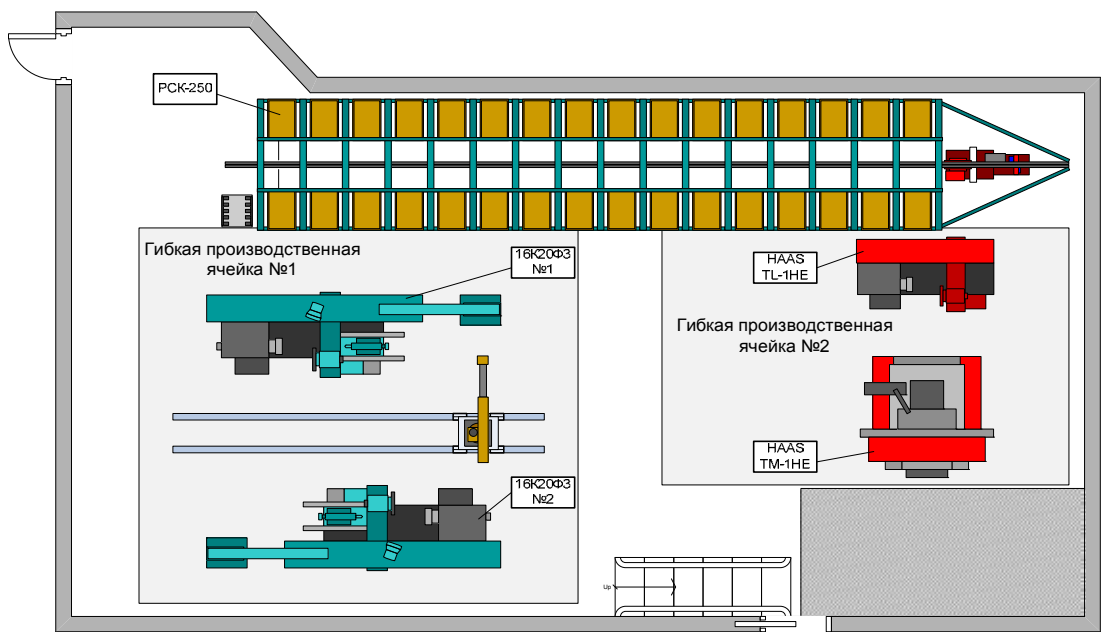
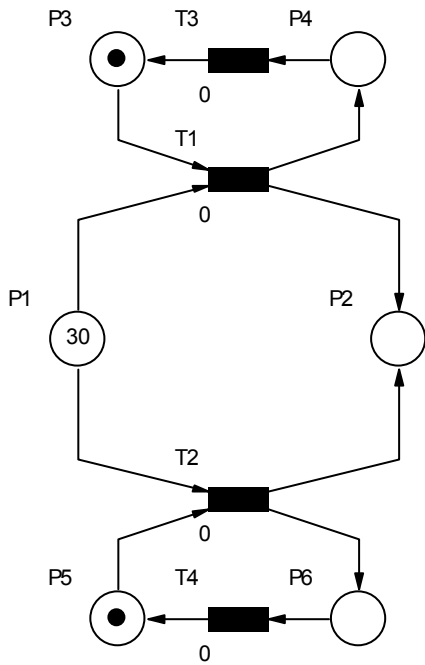


Рис. 3. Структура гибкого производственного комплекса

Таблица 1. Перечень продукции моделируемой производственной системы

№	Наименование изделия	Время токарной обработки 1, мин	Время токарной обработки 2, мин	Время фрезерной обработки, мин	Требуемый объем выпуска	Объем партии	Общее время операций, мин
1	Болт 1	2,56	0	0	6	6	15,36
2	Вал 1	2,8	0	0	8	4	22,4
3	Ввертыш	8,4	6,5	3,4	2	2	36,6
4	Вал 2	6,3	0	0	18	6	113,4
...
29	Полумуфта 1	11,6	0	6,9	2	2	37
30	Палец	5,2	6,8	0	36	6	432



Функциональное назначение **позиций** модели:
 P1 – заготовки изделий в ячейках склада,
 P2 – готовые изделия в ячейках склада,
 P3 – производственная ячейка №1 готова,
 P4 – обработка очередного изделия в ячейке №1
 завершена,
 P5 – производственная ячейка №2 готова
 P6 – обработка очередного изделия в ячейке №2
 завершена

Функциональное назначение **переходов**:
 T1 – обработка изделия в ячейке №1,
 T2 – обработка изделия в ячейке №2,
 T3 – переналадка оборудования в ячейке №1,
 T4 – переналадка оборудования в ячейке №2

Рис. 4. Сеть Петри верхнего уровня

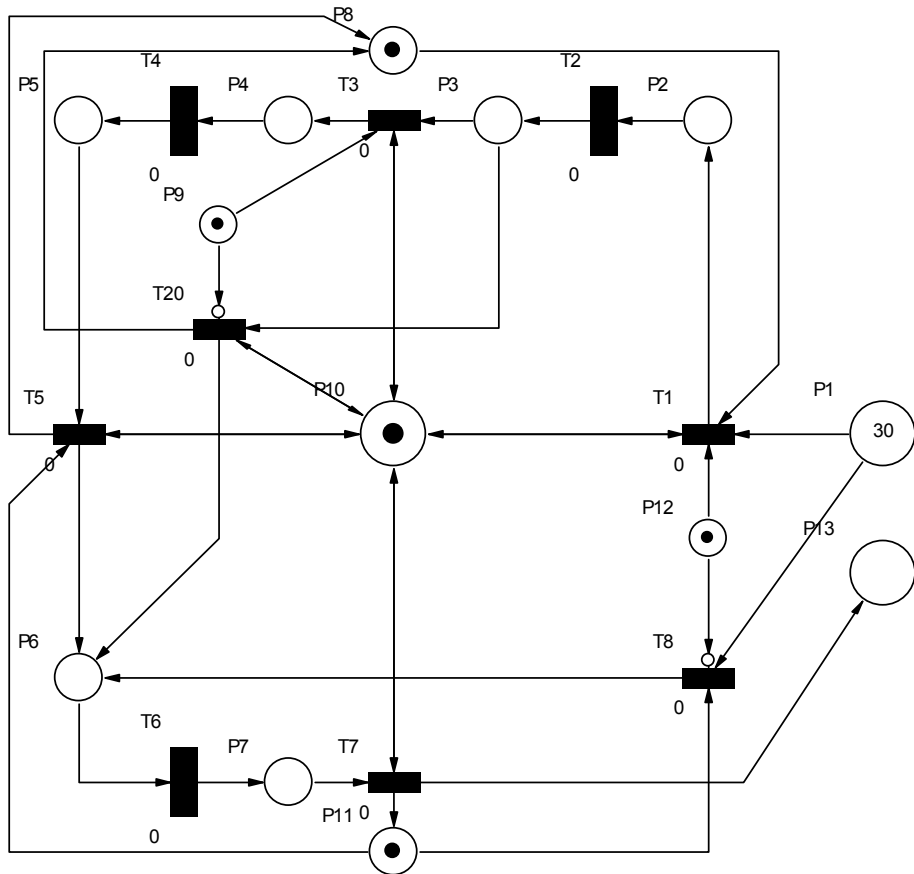


Рис. 5. Сетевая модель первой производственной ячейки

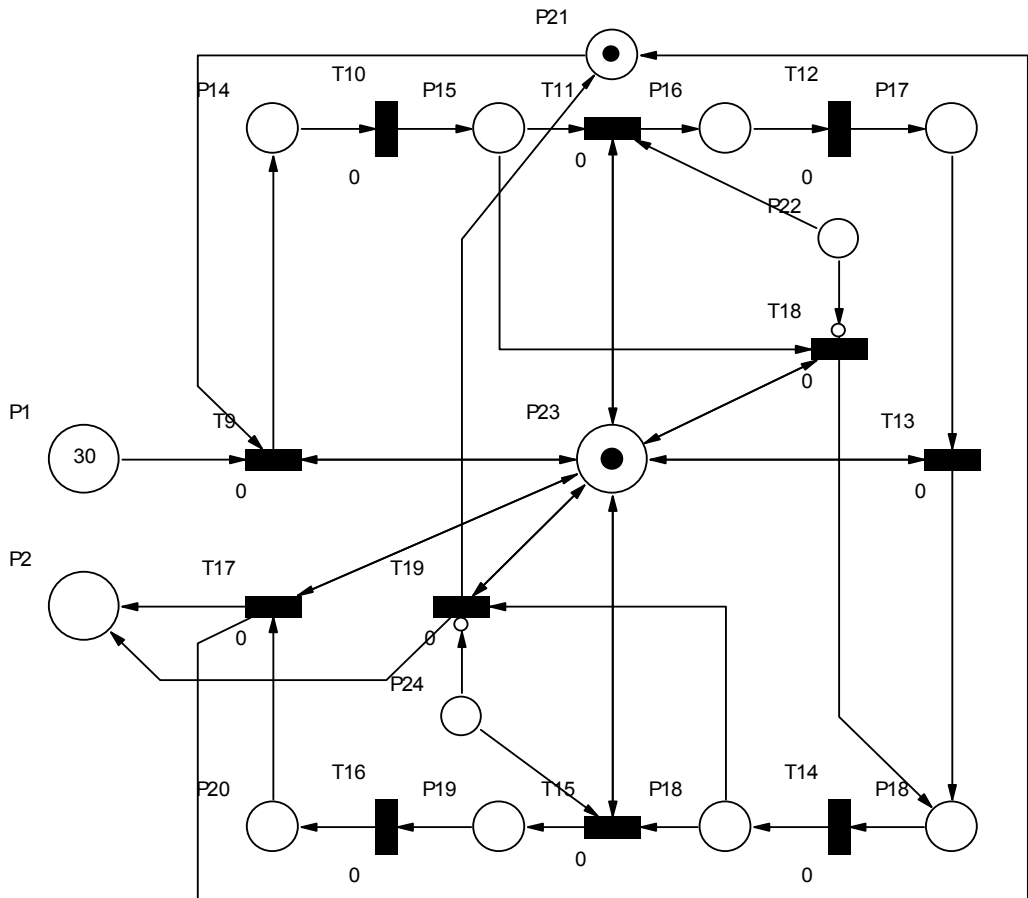


Рис. 6. Сетевая модель второй производственной ячейки

Функциональное назначение позиций в приведенных моделях:

- P1 – заготовки изделий в локальном накопителе,
- P2 – готовые изделия в локальном накопителе,
- P3 – изделия после обработки с первой установки в станке TL-1HE,
- P4 – заготовки в токарном станке TL-1HE,
- P5 – изделия после обработки со второй установки в токарном станке TL-1HE,
- P6 – заготовки в обрабатывающем центре Haas TM-1HE,
- P7 – готовые изделия после фрезерной обработки в станке Haas TM-1HE,
- P8 – состояние токарного станка TL-1HE (свободен или занят),
- P9 – определение количества установок (одна или две),
- P10 – состояние робота – манипулятора (свободен или занят),
- P11 – фрезерный станок TM-1HE свободен,
- P12 – определение типа обработки (токарно-фрезерная или только фрезерная),
- P13 – заготовки в токарном станке TL-1HE,
- P14 – заготовки в токарном станке 16K20Ф3 №1,
- P15 – изделия после обработки с первой установки в станке 16K20Ф3 №1,
- P16 – заготовки в токарном станке 16K20Ф3 №1,

- P17 – изделия после обработки со второй установки в станке 16К20Ф3 №1,
- P18 – изделия после обработки с первой установки в станке 16К20Ф3 №2,
- P19 – заготовки в токарном станке 16К20Ф3 №2,
- P20 – изделия после обработки со второй установки в станке 16К20Ф3 №2,
- P21 – состояние станков ячейки №2 (свободны или заняты),
- P22 – определение количества установок (одна или две),
- P23 – состояние робота – манипулятора (свободен или занят),
- P24 – определение количества установок (одна или две),
- P25 – заготовки в токарном станке 16К20Ф3 №2.

Функциональное назначение переходов:

- T1 – установка изделия из входного накопителя в токарный станок TL-1HE,
- T2 – токарная обработка с первой установки на станке TL-1HE,
- T3 – переустановка изделия в станке TL-1HE,
- T4 – токарная обработка со второй установки на станке TL-1HE,
- T5 – перенос изделия в ячейке №1 с токарного станка на обрабатывающий центр,
- T6 – фрезерная обработка на станке ТМ-1HE,
- T7 – перенос готового изделия в накопитель,
- T8 – установка изделия из входного накопителя в токарный станок ТМ-1HE,
- T9 – установка изделия из входного накопителя в токарный станок 16К20Ф3 №1,
- T10 – токарная обработка с первой установки на станке 16К20Ф3 №1,
- T11 – переустановка изделия в станке 16К20Ф3 №1,
- T12 – токарная обработка со второй установки на станке 16К20Ф3 №1,
- T13 – перенос изделия на станок со станка 16К20Ф3 №1 на станок 16К20Ф3 №2,
- T14 – токарная обработка с первой установки на станке 16К20Ф3 №2,
- T15 – переустановка изделия в станке 16К20Ф3 №2,
- T16 – токарная обработка со второй установки на станке 16К20Ф3 №2,
- T17 – перенос изделия из токарного станка 16К20Ф3 №2 в выходной накопитель ,
- T18 – перенос изделия на станок со станка 16К20Ф3 №1 на станок 16К20Ф3 №2,
- T19 – перенос изделия из токарного станка 16К20Ф3 №2 в выходной накопитель,
- T20 – перенос изделия из токарного станка TL-1HE на станок ТМ-1HE.

После формирования модели производственной системы сформируем структуру и параметры нейросети, используемой для оптимизации процессов в системе. Нейросеть формируется на основе выявления качественных зависимостей различных параметров.

Нейросетевой контроллер содержит однонаправленную искусственную нейронную сеть наиболее распространенного типа – многослойный персептрон [5]. Нейросеть содержит входной слой, выходной и один скрытый слой (рис. 7). Количество нейронов входного слоя соответствует числу учитываемых факторов. Количество нейронов в выходном слое равно количеству альтернативных маршрутов изделий в производственной системе (ПС). Для рассматриваемой системы входные нейроны воспринимают информацию о текущих значениях коэффициентов загрузки основного технологического оборудования. Система содержит четыре металлообра-

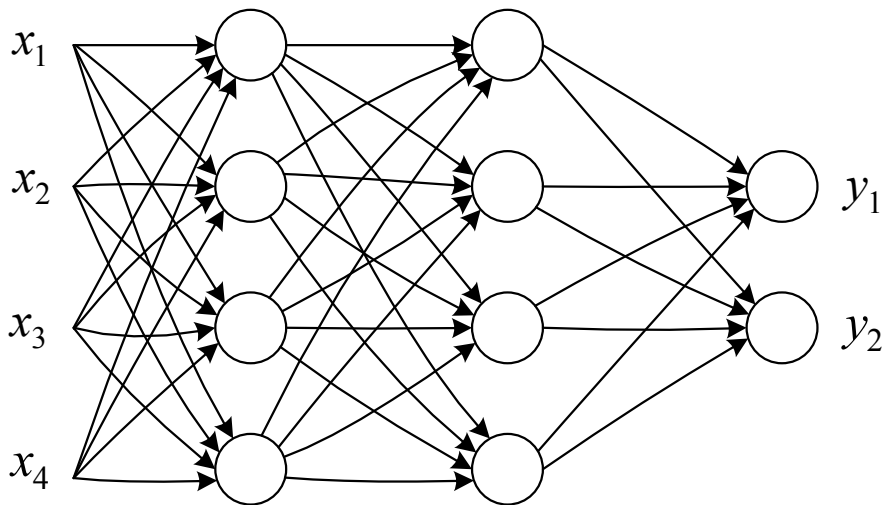


Рис. 7. Структура нейросети

батывающих станка, поэтому количество нейронов входного слоя в нейросети $N=4$. Рассматриваемая ПС содержит две ячейки, соответственно, количество нейронов выходного слоя $M=2$. Количество нейронов промежуточного слоя K должно составлять от N до $3N$. Функция активации нейрона для каждого нейрона сигмоидальная (логистическая). Дальнейшее уточнение количества нейронов в сети и типа функции активации возможно по результатам обучения нейросети и экспериментов.

Создание, обучение и исследование нейросетей осуществляется в пакете Neural Network Toolbox.

Обучение нейросети. Начальная стадия обучения нейросети – это выбор критерия оптимальности, достижение которого она должна обеспечивать. Этот критерий выбирается разработчиком для каждой системы исходя из ее особенностей. Для рассматриваемого роботизированного комплекса выберем следующие критерии:

- 1) минимизация времени производства;
- 2) максимизация загрузки первой ячейки;
- 3) максимизация загрузки первого (токарного) станка второй ячейки.

Рассмотрим примеры обучения нейросети для первого критерия.

Минимизация времени производства

Приведенный критерий максимизируется выравниванием интенсивностей обслуживания заявок в параллельно работающих подсистемах. Соответственно, сформируем обучающую выборку, решающую указанную задачу. Обучение производится на основе градиентного алгоритма обратного распространения ошибки с адаптацией длины шага. Допустимая погрешность классификации входного вектора (квадрат отклонения) выбрана равной 0,001. Максимальное количество шагов обучения – 10000.

Текст программы Matlab

```
p=[[1;1;1;1] [0.6;0.2;0.6;0.3] [0.1;0.2;0.2;0.1] [0.2;0.3;0.6;0.2]
[0.8;0.6;0.7;0.3] [1;0.6;0.8;0.5] [0.7;0.4;0.8;0.7] [1;0.8;1;0.6]];
```

p

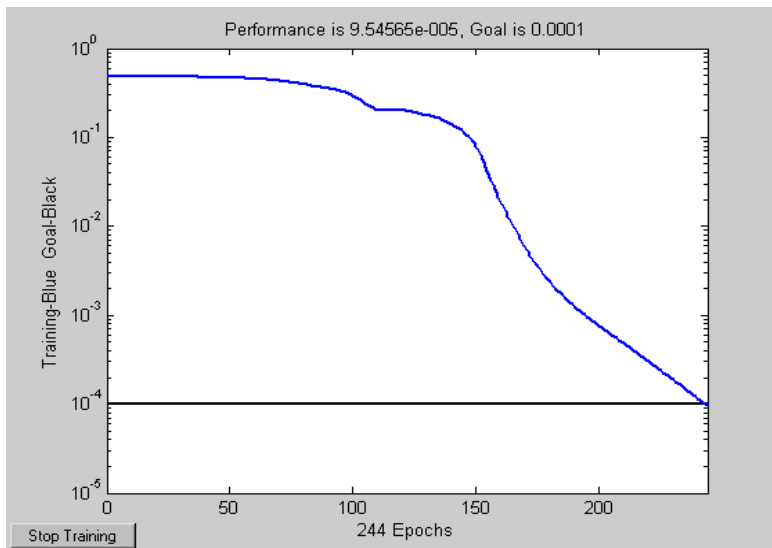


Рис. 8. График процесса обучения нейросети

```
t=[[1;1] [1;0] [1;0] [1;0] [0;1] [0;1] [1;0] [0;1]];
t
%создание и тестирование нейросети
net = newff([0 1; 0 1; 0 1; 0 1],[4 4 2],{'logsig' 'logsig' 'logsig'},'traingdx');
net.trainParam.epochs = 10000;
net.trainParam.goal = 1e-4;
net=train(net,p,t);
a=[[0;0.1;0;0.3] [0.4;0.6;0.8;0.5] [0.5;0.2;0.3;0.7] [1;0.9;0.9;0.7]
[0;0;0;0] [0.9;0.6;0.8;0.6] [0.25;0.2;0.3;0.17] [1;0.19;0.9;0.75]];
y=sim(net,p);
y
y=sim(net,a);
y
gensim(net);
```

Обучение сети заканчивается за 244 шага (рис. 8).

В результате выполнения программы в пакете Simulink формируется модель нейросети (рис. 9).

Реакция обученной нейросети на обучающую выборку соответствует ожидаемой.

```
y = 0.9405 0.9820 0.9833 0.9823 0.0463 0.0328 0.9661 0.0537
0.9999 0.0134 0.0125 0.0129 0.9624 0.9785 0.0344 0.9885
```

Анализ реакции на произвольный вектор позволяет сделать вывод о том, что нейросеть правильно интерпретирует возникающие ситуации. Появляющиеся заявки направляются на обработку в ячейку с сравнительно меньшей текущей загрузкой технологического оборудования.

```
y = 0.9995 0.9991 0.9995 0.3166 0.9981 0.0947 0.9982 0.0453
0.0136 0.0162 0.0133 0.9833 0.0190 0.9864 0.0191 0.9874
```

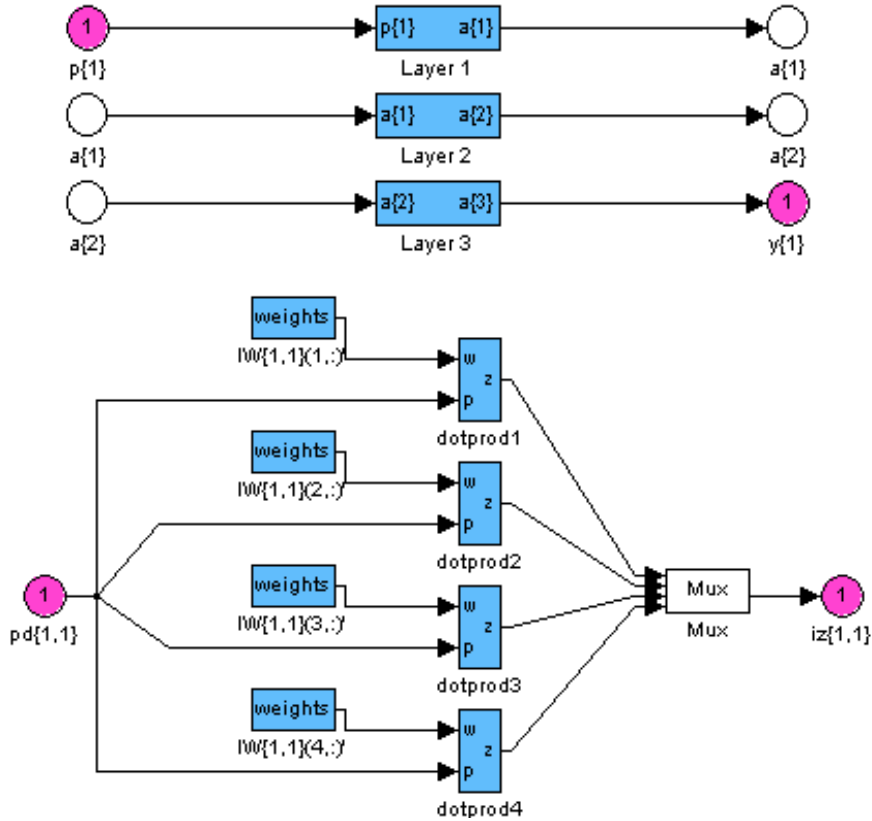


Рис. 9. Структура модели нейросети в пакете Simulink

Аналогично производится обучение сети и для других критериев с соответствующим изменением обучающей выборки. Анализ показывает, что нейросеть правильно интерпретирует возникающие ситуации. Возникающие заявки направляются на обработку в большинстве случаев во вторую ячейку.

Сформированные элементы вектора y используются для выбора активируемых на каждом такте имитации переходов модели. В рассматриваемом примере в модели, приведенной на рис. 2, могут сработать при указанной маркировке переходы t_1 и t_2 . Если $y_1 > y_2$, то срабатывает t_1 , в противном случае – t_2 .

Заключение

Основными преимуществами предложенного метода оптимизации в отличие от оптимизации правилами приоритета являются:

- 1) структурная прямая взаимосвязь критерия оптимальности и факторов, оказывающих на него влияние;
- 2) реализация сложных правил выбора переходов, зависящих от множества факторов и, возможно, меняющихся во времени.

Статья опубликована при поддержке Программы развития Сибирского федерального университета.

Список литературы

1. Емельянов, В. В. Оперативное управление в ГПС / В. Ф. Горнев, В. В. Емельянов, М. В. Овсянников. – М.: Машиностроение, 1990. – 256 с.
2. Котов, В. Е. Сети Петри / В. Е. Котов. – М.: Наука, 1984. – 160 с.
3. Осовский С. Нейронные сети для обработки информации / Пер. с польского И.Д. Рудинского. – М.: Финансы и статистика, 2004. – 344 с.: ил.
4. Бодянский, Е.В. Нейро-фаззи сети Петри в задачах моделирования сложных систем / Е.В. Бодянский, Е.И. Кучеренко, А.И. Михалев. – Днепропетровск.: Системные технологии, 2005. – 311 с.
5. Круглов, В. В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. – М.: Горячая линия – Телеком, 2001. – 382 с.

Algorithm of Controlled Petri Net Simulation

Alexey N. Sochnev

Siberian Federal University,

79 Svobodny, Krasnoyarsk, 660041 Russia

This article presents a general algorithm to perform control operations of Petri nets state performance. Control device model for Petri nets appear to one Petri net or pre-trained neural network. Using neural networks in the simulation experiment with a Petri net allows searching extreme of desired process optimality criterion.

Keywords: Petri net; neural network; net controller.
