

## Об одном способе решения задачи операционного планирования на ситуационной сети

*Д.А. Андреев, А.Н. Панфилов*

*Южно-Российский государственный политехнический университет (НПИ)  
имени М.И. Платова, г. Новочеркасск*

**Аннотация:** В данной статье рассматривается задача построения плана работ в виде цепочки управляющих воздействий, изменяющих систему, и методы ее решения. Строится задача нахождения оптимального пути на ситуационной сети с математическим описанием начальных условий и требованиями к результату. Кратко описывается стандартное решение данной задачи: метод нечетких продукций. Рассматриваются основные проблемы метода нечетких продукций. Описывается решение задачи построения цепочки действий оператора на ситуационной сети с помощью механизма динамического программирования. Предлагается комбинированный метод нахождения цепочек действий с процедурным построением ситуационной сети с использованием метода динамического программирования с ограничениями. Описывается первый шаг и шаг  $l$  алгоритма решения данной задачи. Рассматриваются преимущества данного метода перед другими.

**Ключевые слова:** ситуационное моделирование, динамическое программирование, сложные системы, нечеткая логика, нечеткие ситуационные сети, нечеткий вывод, нечеткая логика, нечеткие множества, теория принятия решений, теория оптимального управления, динамическое программирование.

В статье [1] была дана постановка задачи построения подграфа нечеткой ситуационной сети, соединяющего граф штатных ситуаций  $G(S, V)$  и незапланированную ситуацию  $S^*$ . Так как решение данной задачи заключается в построении подграфа нечеткой ситуационной сети, следует рассмотреть способы решения этой подзадачи. Ситуации могут быть заданы в явном виде или процедурно построены с помощью алгоритмов, как показано в [2]. При процедурном построении используется два подхода – прямой и обратный.

Прямой метод заключается к применению управляющих воздействий к начальным состояниям и сформированным при предыдущих итерациях до построения всего множества состояний. Обратный метод состоит в нахождении связей между уже найденными состояниями.

Построенная ситуационная сеть дает возможность проводить поиск целевого состояния среди ранее сформированных элементов ситуационной сети. Способ нахождения целевого состояния зависит от взаимосвязи компонент состояния. Если составляющие элементы системы независимы друг от друга (когда исследователь принимает решение изучать систему без учета взаимодействия ее составляющих), то целевое состояние определяется как состояние ситуационной сети, обладающее наибольшей степенью нечеткой включенности в целевое состояние. Математически данное утверждение отражается формулой вида:

$$v(S_h, S_u) = \bigwedge_{y_i \in Y} (v(\mu_{S_h}(y_i), \mu_{S_u}(y_i))) \quad (1)$$

$$v(\mu_{S_h}(y_i), \mu_{S_u}(y_i)) = \bigwedge_{T_i^j \in T_i} (\mu_{S_h}(T_i^j) \rightarrow \mu_{S_u}(T_i^j)) \quad (2)$$

$$v(S_h, S_u) = \max_{S_h \in S} (v(S_h, S_u)) \rightarrow S_h \approx S_u \quad (3)$$

В случае, когда происходит рассмотрение системы, в которой учет взаимодействия ее компонент необходим, используются методы анализа и сравнения графов взаимодействий компонент состояний [2].

Построенную ситуационную сеть используют для нахождения траектории выполнения процесса от текущей ситуации до целевой. Таким образом, задача сводится к нахождению наиболее предпочтительного пути между двумя точками на графе. Предпочтительность пути может быть определена с помощью одного из нескольких критериев оптимальности. Если для каждого управляющего воздействия задано потребление определенного ресурса (например, времени реализации данного воздействия), то критерием предпочтительности может выступать минимальность затраты данного ресурса на траектории процесса. В случае, если же задана выгода, получаемая при реализации данного управления, то как критерий

оптимальности можно использовать максимальность получаемой выгоды при проходе от начального состояния к конечному.

Математически начальные условия для решения данной задачи могут быть выражены следующими условиями:

$$\exists s^{нач} \in G(S, V) \quad (4)$$

$$\exists S^{запр} = \{s_1^{запр}, \dots, s_q^{запр}\} \quad (5)$$

$$\exists S^{цел} = \{s_1^{цел}, \dots, s_s^{цел}\} \quad (6)$$

$$\exists w(s_i, u_i) = f : S, U \rightarrow R \quad (7)$$

$$\exists W(s^{нач}, \bar{U}) = f : S, \{U\} \rightarrow R \quad (8)$$

Таким образом, данная задача сводится к нахождению такой цепочки управляющих воздействий, которая удовлетворяет следующим условиям

$$s_i = F(s^{нач}, \bar{U}^*) = \varphi(\dots\varphi(\varphi(s^{нач}, u_1^*), u_2^*)\dots, u_l^*) \in S^{цел} \quad (9)$$

$$\forall s_i = \varphi(\dots\varphi(\varphi(s^{нач}, u_1^*), u_2^*)\dots, u_i^*) \notin S^{запр}; \quad (10)$$

$$W(s^{нач}, \bar{U}^*) \rightarrow \max(\min) \quad (11)$$

Одним из основных методов решения данной задачи является метод нечетких продукций – механизма логического вывода, в рамках которого производится с помощью правил нечеткого вывода. Для функционирования данного метода требуется разработать базу нечетких правил (продукций), по которым из предпосылок будут выводиться заключения [3]. Недостатком данного механизма является достаточно сильная громоздкость, необходимость определенности ситуационной сети на этапе поиска пути, а также сильная привязка решения к предметной области.

Данных недостатков лишен метод динамического программирования, так как для работы этого метода не требуется определенная база знаний. Для

определения задачи динамического программирования требуется установить следующие компоненты задачи [4].

*Состояние*  $S^d$  (индекс  $d$  в обозначениях параметров задачи введен, чтобы их обозначения не совпадали с параметрами, описывающими ситуационную сеть) характеризуется совокупностью параметров, однозначно описывающих поведение системы. Состояния в задаче принятия решений в данном случае могут быть описаны состояниями ситуационной сети  $S_h$  в виде, описанном в статье [1].

*Этапы* – подзадачи принятия решения, определяемые способом деления задачи принятия решения на составляющие – совокупность состояний, отстоящих на одинаковое количество управляющих воздействий от конечного состояния (при обратном проходе) или от начального состояния (при прямом проходе).

*Управление*  $U_i^d$  – целенаправленное воздействие на систему. В качестве элемента, описывающего управления в задаче, берутся управляющие воздействия из состояний ситуационной сети. В таком случае принятием решения выступает выбор управляющего воздействия при попадании оператора в определенную ситуацию, или на каждом этапе прохождения траектории процесса.

*Условное оптимальное управление*  $U_i^d(S^d)$  – управление, на котором достигается максимизация или минимизация критерия предпочтительности.

*Оператор перехода*  $\varphi(S^d, U_i^d)$  – механизм, который описывает переход от одного состояния к другому. В данном случае он может быть представлен переходом от одной ситуации к другой в ситуационной сети. Математически он может быть представлен совокупностью операций, представленных в прямом методе процедурного построения ситуационной сети [4,5].

*Локальный доход*  $w(S^d, U_i^d)$  – степень прироста критерия оптимальности при применении управления к состоянию. Для задачи минимизации затрат –

---

ресурс, потраченный при применении управления  $U_i^d$  в состоянии  $S^d$ , для задачи максимизации выгоды – полученный доход при применении определенного управления на определенном состоянии. При этом локальный доход может быть прикреплен к управлению непосредственно, или задаваться как функция от компонент текущего и следующего после применения управления состояния.

*Условный оптимальный доход* – суммарный показатель критерия оптимальности после нескольких этапов применения управления [6]. Оператором перехода может служить функция перехода между ситуациями, которая указана в статье [1]

Решение задачи строится следующим образом. Если ситуационная сеть уже построена, состав характеристик системы можно принять неизменным, а конечное состояние или их совокупность явно определены, то полученную задачу динамического программирования можно решить методом обратного прохода от результата (или результатов) к начальному состоянию. Поиск решений производится до тех пор, пока каждый путь не достигает начальной точки, затем среди множества оставшихся конкурирующих траекторий ведения процесса выбирается наиболее предпочтительная.

В иных случаях, можно использовать комбинацию метода прямого построения ситуационной сети с прямым проходом динамического программирования, который удобен для решения задач построения бортовых документаций [7]. Данный подход имеет ряд преимуществ:

- минимизация времени вычислений, так как при раздельном выполнении операций построения сети и нахождения кратчайшего пути операция переходов между операциями дублируется,
- оптимизация ситуационной сети с помощью выбора наиболее предпочтительных путей среди конкурирующих,

Определение конкурирующих решений и выбор наиболее предпочтительного среди них аналогично подходу, представленному ранее. На каждом этапе идет проверка по формулам (2), соответствует ли данное состояние одному из конечных или одному из «тупиковых» состояний. При этом классы конечных и тупиковых состояний могут задаваться с помощью вектора значений компонент состояния, тогда в (2) множество  $Y$  всех компонент состояний ситуационной сети может быть заменено соответственно на множество компонент, критичных для определения класса конечных  $Y_k$  или тупиковых состояний  $Y_m$ .

Первый шаг метода состоит в получении новых состояний из начального состояния  $S^0$  с помощью воздействия операторов переходов (воздействием на каждую характеристику системы) получается ряд новых ситуаций  $S^1_1, S^1_2, S^1_i \dots S^1_n$  [4, 5]. После для каждого состояния формируется цепь состояний вида  $\overline{S^1_j} = (S^0, S^1_j), j = \overline{1, n}$ , которая отражает траекторию движения системы. Для каждой цепи находится функция глобального дохода  $W(\overline{S^1_j})$ . Затем для каждой из полученных ситуаций производится сравнение по формулам (1-3) с множествами целевых состояний, запрещенных состояний, а также с множеством полученных ранее состояний. Цепи, конечное состояние которых совпало с одним из состояний запрещенной группы, исключаются из рассмотрения. Цепи, конечное состояние которых совпало с одним из состояний целевой группы, помещаются во множество результирующих цепей, где функции глобального дохода цепей сравниваются между собой, и в качестве решения задачи выводится цепь с наиболее экстремальными значениями функции глобального дохода. Цепи с совпадающими конечными состояниями определяются как конкурирующие. Данное определение цепей возможно, так как поведение в ситуации не зависит от истории переходов между состояниями системы [8].

---

Конкурирующие цепи сравниваются по функции глобального дохода, и цепь с менее экстремальным значением функции глобального дохода

Отдельный шаг прохода метода на итерации  $l$  производится по следующей схеме: Сначала из текущей ситуации  $S_i^l$ , (где  $l$  – номер уровня (итерации работы алгоритма), на котором получена ситуация,  $i$  – номер ситуации на данном уровне ) получается ряд новых ситуаций [4, 5]. Затем также формируются цепочки состояний  $\bar{S}_j^l = (S^0, S_i^1, \dots, S_i^l)$ . Затем для каждого полученного состояния производятся описанные ранее действия.

Поиск решений производится до тех пор, пока каждый путь не достигнет одного из целевых или тупиковых состояний, затем среди множества оставшихся конкурирующих траекторий ведения процесса выбирается наиболее предпочтительная.

Таким образом, формулами (1-11) описывается задача нахождения оптимального пути на ситуационном графе. Также представляется решение данной задачи в виде механизма процедурного построения графа, представленного в [2], с методом динамического программирования с учетом ограничений. Данный подход позволяет решить задачу, представленную в [1]. Описанный метод справляется с задачей построения цепочек состояний оптимального плана действий более универсально, что позволяет облегчить этап изменения описательной модели при изменении состава самой системы [9, 10].

### Литература

1. Андреев Д. А., Панфилов А. Н., Скоба А. Н. Управление операционными процессами операторов сложных систем // Инженерный вестник Дона, 2017, №3 URL: [ivdon.ru/ru/magazine/archive/n3y2017/4322](http://ivdon.ru/ru/magazine/archive/n3y2017/4322)



2. Астанин С.В., Жуковская Н.К. Управление бизнес-процессами на основе их моделирования нечеткими ситуационными сетями // Управление большими системами: сборник трудов, 2012. №37 С. 145-163

3. Борисов В.В., Федулов А.С., Зернов М.М., Основы нечеткого логического вывода: Учебное пособие для вузов / М.: «Основы нечеткой математики». Вып. 4., 2014, 122 с.

4. Черноморов Г.А. Теория принятия решений: Учебное пособие. 3е изд. перераб. и доп. изд. Новочеркасск: Ред. журн. «Изв. вузов. Электромеханика», 2005, 448 с.

5. Robert T. Clemen, Terence Reilly Making Hard Decisions: An Introduction to Decision Analysis. Duxbury: Thomson Learning, 2001, 733 p.

6. Блейк Р., Моутон Д. Научные методы управления. Киев: Наукова думка, 2000, 240 с.

7. Operations Data File Standards. International Space Station Program : National Aeronautics and Space Administration International Space Station Program Johnson Space Center Houston, Texas ; Revision L, December 2004. – 196 p. – (International Space Station Program Operations Data File Standards)

8. Поспелов Д.А. Ситуационное управление. Теория и практика. – М.: Наука, 1986, 228 с.

9. Фараонов А.В. Ситуационные центры как инструмент оценки подготовки специалистов и эффективности принятия решения // International Journal of Open Information Technologies, 2015. Vol 3 No 4, С. 1-7.

10. Андреев Д. А., Панфилов А. Н., Погорелов А. С. Математическое моделирование и исследование процессов интерактивного взаимодействия в виртуальной среде: проблематика и основные тезисы // Инженерный вестник Дона, 2015, №2 URL: [ivdon.ru/ru/magazine/archive/n3y2015/3135](http://ivdon.ru/ru/magazine/archive/n3y2015/3135)



## References

1. Andreev D. A., Panfilov A. N., Skoba A. N. Inzhenernyj vestnik Dona (Rus), 2017, №3. URL: [ivdon.ru/ru/magazine/archive/n3y2017/4322](http://ivdon.ru/ru/magazine/archive/n3y2017/4322)
2. Astanin S.V., Zhukovskaja N.K. Upravlenie bol'shimi sistemami: sbornik trudov, 2012. №37, pp. 145-163
3. Borisov V. V., Fedulov A. S., Zernov M. M. Osnovy nechetkogo logicheskogo vyvoda: Uchebnoe posobie dlya vuzov [Fundamentals of fuzzy inference. Tutorial for institutes of higher education]. Moscow: Hot Line. Telecom, Fundamentals of fuzzy mathematics, Book 4, 2014, 122 p.
4. Chernomorov G.A. Teoriya prinyatiya resheniy: Uchebnoe posobie [Decision Theory: Textbook]. 3e izd. pererab. idop. izd. Novocherkassk: Red. zhurn. «Izv. vuzov. Elektromekhanika», 2005, 448 p.
5. Robert T. Clemen, Terence Reilly Making Hard Decisions: An Introduction to Decision Analysis. Duxbury: Thomson Learning, 2001, 733 p.
6. Bleyk R., Mouton D. Nauchnye metody upravleniya [Scientific methods of control]. Kiev: Naukova dumka, 2000, 240 p.
7. Operations Data File Standards. International Space Station Program : National Aeronautics and Space Administration International Space Station Program Johnson Space Center Houston, Texas ; Revision L, December 2004. 196 p. (International Space Station Program Operations Data File Standards)
8. Pospelov D.A. Situatsionnoe upravlenie. Teoriya i praktika. [Situational management. Theory and practice]. M.: Nauka, 1986, 228 p.
9. Faraonov A.V. International Journal of Open Information Technologies, 2015. Vol 3 no 4, pp. 1-7.
10. Andreev D. A., Panfilov A. N., Pogorelov A. S. Inzhenernyj vestnik Dona (Rus), 2015, №2. URL: [ivdon.ru/ru/magazine/archive/n3y2015/3135](http://ivdon.ru/ru/magazine/archive/n3y2015/3135)