

**А.М.Алексеев**

*(Украина, Днепрпетровск, Национальный горный университет)*

## **ГЕНЕРАЦИЯ ПРОЕКТОВ ПЛАНОВ БОЕВЫХ ДЕЙСТВИЙ ПРИ ТУШЕНИИ ПОЖАРОВ НА ШАХТАХ С ИСПОЛЬЗОВАНИЕМ ПРЕЦЕДЕНТОВ**

Сложность процессов составления планов ликвидации аварий (ПЛА), и принятия оперативных решений при тушении пожаров на шахте, требует создания компьютерных систем поддержки принятия решений руководством объекта управления в экстремальных ситуациях. В данном случае возникает задача создания баз знаний, по которым возможно оперативно принимать в экстремальных условиях обоснованные, точные решения. Модель знаний данной предметной области должна содержать:

- описания возможных аварийных ситуаций;
- процедуры конкретизации или обобщения этих ситуаций;
- процедуры их адаптации к реальным условиям, сложившимся на объекте управления.
- схемы принятия решений по ликвидации аварий.

Все это обуславливает применение методов ситуационного управления для автоматизации процессов составления ПЛА и принятия решений при оперативной ликвидации непредвиденных аварийных ситуаций на шахтах.

В статье рассматривается вопрос принятия решений на основании онтологии прецедентов. Работа соответствует ТЗ научного направления ГП – 407 ДНУ «Интеллектуальные компьютерные технологии обработки данных, прогнозирования и управления».

Управленческое решение по тушению пожара (РТП) представляет собой тактический план боевых действий или нахождение выхода из конкретной проблемной ситуации [1].

Исходными данными для постановки задачи планирования, решаемой алгоритмом генерации проектов планов боевых действий, являются:

- характеристика оперативной обстановки, (значения атрибутов объектов предметной области - ПО), обобщения логических выражений, процессы, события и действия подразделений ГВГСС на текущий момент времени;
- множество целей (боевых задач), которые могут быть поставлены перед отделениями ГВГСС в текущей ситуации;
- ограничения, в том числе определяющие порядок приоритета достижения целей.

Таким образом, алгоритм формирования проектов планов боевых действий, в каждый момент времени обрабатывает информацию, получаемую в результа-

те выбора различных вариантов, тем самым предлагая вероятные сценарии развития оперативной обстановки.

При автоматическом формировании тактических планов боевых действий при тушении пожаров множество боевых задач (спасение людей, ликвидация очага пожара, защита шахтных объектов, эвакуация материальных ценностей) и ограничения, устанавливающие порядок их достижения, определяются на основе правил, полученных из онтологии боевых действий и значений обобщений. Эти обобщения определяются в результате функционирования модели причинно-следственных связей ПО или вводятся пользователем.

В работе [2] описывается система поддержки принятия решений, успешно используемая при проведении операций по спасению гражданского населения, военнослужащих тылового обеспечения или конкретных лиц из зон военных конфликтов. Такие операции обычно включают быстрое, точечное применение сил (военного прикрытия спасательной операции), например, при эвакуации дипломатических лиц с заранее запланированным отходом.

Система содержит функциональный модуль планирования общецелевого назначения (HICAP - Hierarchical Interactive Case-Based Architecture for Planning) [3].

HTN - планирование обладает рядом преимуществ, которые обуславливают применение его в перечисленных выше системах:

- простота в реализации алгоритма;
- наглядность алгоритма;
- гибкость в выборе методов, используемых для поиска решений и адаптации планов;
- полнота и непротиворечивость алгоритмов, основанных на HTN-планировании.

Предлагаемый подход (иерархическое сетевое планирование боевых действий - ИСП БД), базируясь на указанных выше идеях HTN-планирования [2,3,4], позволяет учитывать следующие особенности рассматриваемой предметной области:

- параллельное выполнение нескольких тактических задач (соответствует нестрогой упорядоченности задач, входящих в план боевых действий);
- наличие большого количества различных операций, используемых для реализации тактических целей в исследуемой предметной области (предметная область представляет собой достаточно большой объем знаний о действительности).

#### *Иерархическое сетевое планирование боевых действий.*

В иерархическом сетевом планировании боевых действий каждое состояние среды представляется множеством атомов, имеющих значение *истина* (true) в этом состоянии. Действия, называемые в ИСП БД *элементарными задачами*, соответствуют переходам состояний, т.е. каждое действие – это частичное отображение множества состояний по отношению к множеству состояний. В ИСП БД происходит поиск метода реализации поставленных целей в пространстве задач. На рис.1. приведен алгоритм ИСП БД.

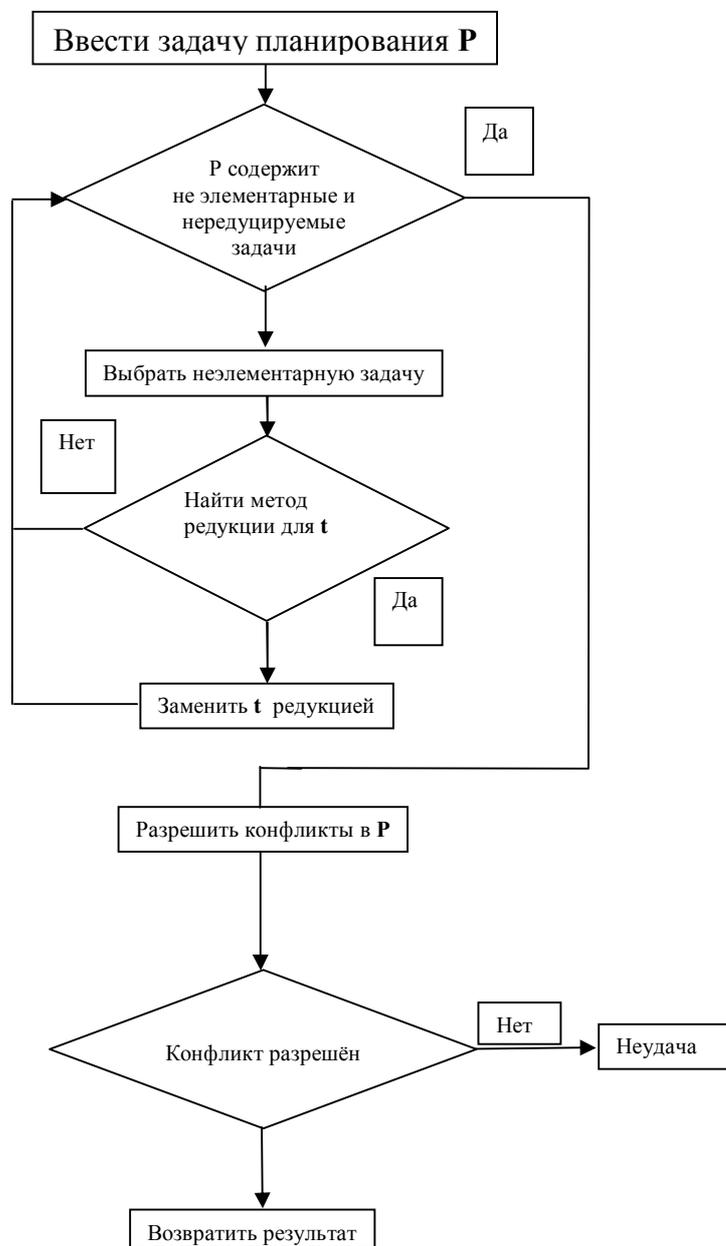


Рис.1. Алгоритм ИСП БД

Порожденная сеть задач может содержать конфликты, являющиеся следствием их взаимодействий между задачами. Например, если потрачено достаточно много времени на поиск транспортного средства (например, вертолета), то можно потерять шансы спасти людей или потушить пожар. Процессы нахождения и разрешения таких конфликтов (например, введением дополнительных ограничений) выполняются соответствующими процедурами, называемыми критикой. Критика впервые была введена в системе NOAH [4,5], для идентификации и управления различного вида взаимодействиями в различных сетях задач.

*Формализованное описание алгоритма формирования планов боевых действий при ликвидации аварий.*

Формальный аппарат для построения оперативных планов выполнения боевых действий основывается на использовании специального языка иерархического сетевого планирования. Синтаксис этого языка описывается ниже.

Назовем  $L$  языком иерархического сетевого планирования боевых действий (ИСП БД) язык первого порядка (с некоторыми расширениями), сигнатура которого представлена семеркой:

$$\Sigma = \langle S, V, C, P, F, T, N \rangle,$$

где —  $S = S^0 \cup S^d$  — множество сортов, разделенное на два независимых множества сортов (типов) данных и сортов объектов;  $V$  — множество символов переменных;  $C$  — конечное множество символов констант;  $F$  — конечное множество символов операций;  $P$  — конечное множество символов элементарных тактических операций;  $T$  — конечное множество символов составных тактических операций;  $N$  — множество символов, используемых для маркировки задач.

Описанные выше характеристики оперативной обстановки имеют сорта, относящиеся к объектному типу:

$$S_{ag} \cup S_g \cup S_0 \cup S_{fp} \cup S_f \cup S_r \cup S_e \cup S_{ro} \cup S_p \cup S_d \subset S^0.$$

Основные, используемые для определения значений атрибутов объектов ПО, типы данных:  $\text{bool}, \text{string}, \text{int}, \text{real}, \text{set} \in S^d$ .

Аналогично для объектов с  $n$  параметрами

$$y_0^m((x_i^1, K_i^{j_1}), \dots, (x_i^n, K_i^{j_n})) = \text{true}, \text{ если } x_i^1 \in K_i^{j_1}, \dots, x_i^n \in K_i^{j_n}.$$

*Состояние оперативной обстановки* в момент времени  $t$  интерпретируется алгоритмом ИСП БД в виде списка основных атомов. Если  $P$  есть  $k$ -мерная предикатная константа, и  $t_1, \dots, t_k$  — термы, то  $P(t_1, \dots, t_k)$  — атом.

Например, объект тушения, объект противопожарного водоснабжения, структурная часть объекта пожара (горная выработка, подземный склад, кабельный тоннель и т.п.) имеют структуру, определяемую соответствующим классом  $C_{S_0}$ . Объекты ПО могут быть простыми и составными (т.е. включать другие объекты).

Самым верхним (корневым классом) для всей иерархии классов объектов ПО является класс *Domain Object* ( $h, \text{type}$ ), включающий атрибуты, общие для всех объектов ПО:

$h$  — название объекта сорта  $\text{string}$ ,  $\text{type}$  — тип объекта ПО;

$REL$  — множество ассоциативных отношений  $rel$ , сортов  $S_{rel_1}, \dots, S_{rel_n} \in S_{rel}$  между объектами ПО;

$OP(t)$  — множество очагов пожара сортов  $fp_i$  видов  $S_{fp_1}, \dots, S_{fp_n} \in S_{fp}$ , выявленных в момент времени  $t$ :  $op_i(h, \text{obj}, \text{type}, \text{ch}, \text{sq}, v_l, v_s)$ .

Атрибуты  $\text{type}, \text{ch}, \text{sq}, v_l, v_s$  определяют, соответственно, тип очага пожара, характер распространения огня, площадь сечения горной выработки, линейную скорость распространения продуктов горения, быстроту изменения площади очага пожара и направление развития пожара.

$F(t)$  – множество пожарно-спасательных подразделений (отделений ГВГСС)  $f_i$  видов  $S_{f_1}, \dots, S_{f_n}$ , привлеченных на момент времени  $t$  к тушению пожара (отделение, часть, отряд и т.д.).

$R(t)$  – множество единиц техники и ПТВ (пожарных автоцистерн, подземных систем пожаротушения, генераторов инертных газов)  $r_i$  сортов  $s_{r_1}, \dots, s_{r_n} \in S_r$  используемых в момент времени  $t$  для тушения пожара;

$OS(t)$  – множество используемых огнетушащих средств  $os_i : s_{os_1}, \dots, \in S_{os}$ .

$P(t)$  – множество процессов  $p_i$  сортов  $s_{p_1}, \dots, s_{p_n} \in S_p$ , характеризующих оперативную обстановку в момент времени  $t$ :

$$P(h, obj, t, t_0, per, nc),$$

где  $h$  – название процесса, а  $obj$  – ссылка на связанный с ним объект,  $t$  – продолжительность процесса,  $t_0$  – время начала процесса (по абсолютной шкале времени),  $per$  – период динамики процесса;  $nc$  – необходимое условие – предикат, определяющий возможность (*true*), невозможность (*false*) инициации данного процесса.

Если  $t_1, \dots, t_k$  – константы, то  $P(t_1, \dots, t_k)$  – основной атом. Каждая функция принадлежности  $y_i$  отображается в соответствующий атом  $P_i$ . При этом

$$P_1(t) = y_i(x_i, K_i^j)$$

.....

$$P_1(t_1, \dots, t_n) = y_0^n((x^1, K^{j_1}), \dots, (x^n, K^{j_n}), t_1 = x^1, \dots, t_n = x^n).$$

Множество задач, выполнение которых началось по времени раньше остальных задач в сети, обозначим через *first[d]*. Множество задач, выполнение которых закончилось по времени позже остальных задач в сети, обозначим через *last[d]*.

Так, например, формальное описание сети задач, изображенной на рис.2, имеет вид:

$[(n_1 : do I \text{ подать пену на тушение } (v_1))(n_2 : provide \text{ "защита конструкций } (v_2), begin](n_3 : achieve \text{ [ликвидировать очаг возгорания } (v_3)](n_4 : achieve \text{ [разобрать завал}(v_3)](n_5 : achieve \text{ [эвакуировать людей } (v_3)](n_6 : provide \text{ [защита конструкций}(v_2), end]); (n_1 < n_5) \wedge (n_2 < n_5) \wedge (n_3 < n_5) \wedge (n_4 < n_5) \wedge (n_3 < n_6); \wedge \text{ подача пены обеспечена } (v_1), n_3) \wedge n_2 \text{ защита конструкций обеспечивается } (v_2), n_5) \wedge (n_3, \text{ очаг ликвидирован } (v_3), n_5) \wedge (n_4, \text{ завал разобран}(v_3), n_5) \wedge (\text{подача пены обеспечена } (v_1), n_2) \wedge (\text{взрывоопасных веществ нет } (v_3), n_5) \wedge \neg (v_1 = v_2) \wedge \neg (v_1 = v_3) \wedge \neg (v_2 = v_3)].$  При нахождении алгоритма  $s$ , решающего задачу планирования боевых действий  $P = \langle d, D(t) \rangle$  возможны два варианта:

1. Пусть  $d$  – элементарная сеть тактических задач (т.е. сеть, содержащая только элементарные тактические операции) и  $S(t)$  - исходное состояние оперативной обстановки, тогда план  $s$  является расширением  $d$  в  $S(t)$  и обозначается как  $s \in comp(d, D(t))$ .

Если план  $s$  выполним (т.е. предусловия каждого действия в  $s$  удовлетворены) и соответствует общему порядку выполнения элементарных тактических задач в  $d$ , то с учетом ограничений  $d$ , для неэлементарной сети задач  $d$  имеет место  $comp(d, D(t)) = 0$ .

2. Пусть  $d$  – неэлементарная сеть тактических задач, содержащая неэлементарную тактическую задачу  $\{n : a\}$ ;  $m = (a', d')$  – метод и  $q$  – унификатор для  $a\_u\_a'$ , тогда  $reduce(d, n, m)$  определяется как сеть задач, полученная из  $dq$  путем замены  $(n : a)$   $q$  на узлы задачи  $d'q$ , изменяя при этом формулу ограничений  $j$  для  $d'q$  на  $j'$  (так же, как для  $compose$ ), и включая формулу ограничений  $d'q$ . Обозначим множество редукций  $d$  как  $red(d, D(t))$ . Редукция подразумевает формализацию декомпозиции задач.

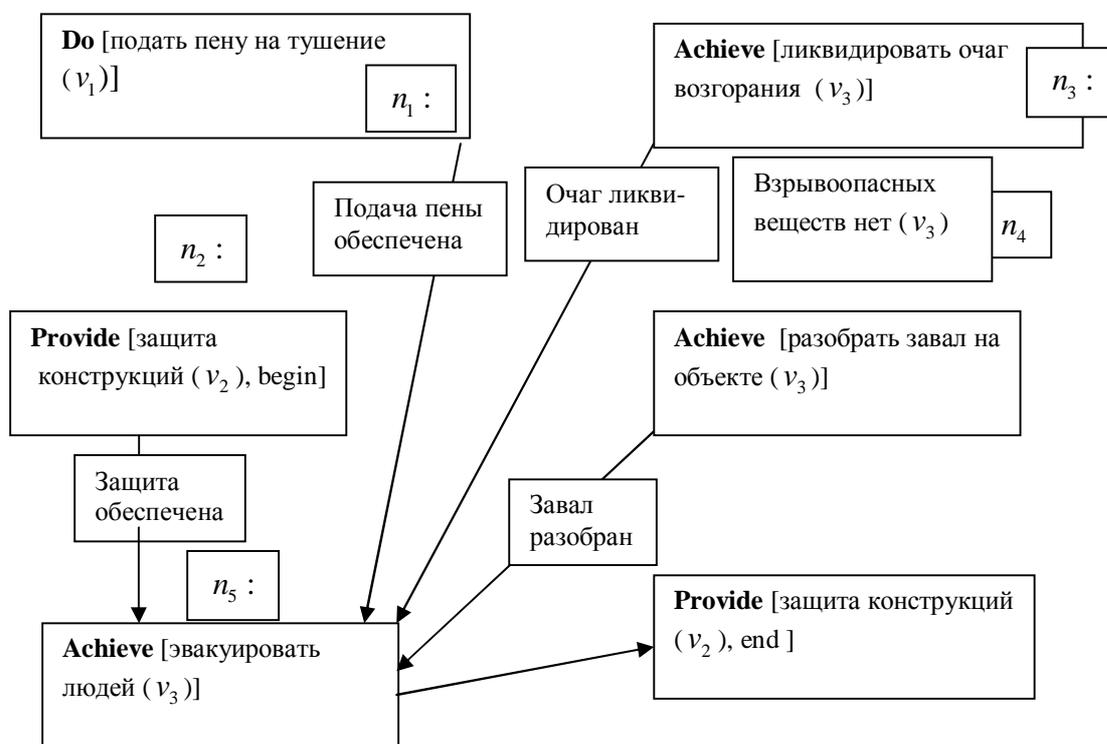


Рис.2. Пример графического представления сети задач

Если план  $s$  выполним (т.е. предусловия каждого действия в  $s$  удовлетворены) и соответствует общему порядку выполнения элементарных тактических задач в  $d$ , то с учетом ограничений  $d$ , то для неэлементарной сети задач  $d$  имеет место  $comp(d, D(t)) = 0$ .

План  $s$  является алгоритмом для элементарной сети задач  $d$  в исходном состоянии  $I$ , если  $s \in comp(d, D(t))$ ;  $s$  является планом для неэлементарной сети задач  $d$  в исходном состоянии  $I$ , если  $s$  является планом для редукции  $d' \in red(d, D(t))$  в исходном состоянии  $I$ .

Множество планов  $sol(d, D(t))$ , решающих некоторую задачу планирования  $P = \langle d, D(t) \rangle$ , определяется как,  $sol_{n+i}(d, D(t)) = sol_n(d, D(t)) \cup_{d' \in red(d, D(t))} sol_n(d', D(t))$ ,  $sol(d, D(t)) = \cup_{n < \omega} sol_n(d, D(t))$ , где  $sol_n(d, D(t))$  – множество планов, которое сформировано за  $n$  шагов, а  $sol(d, D(t))$  – множество планов, которое может быть сформировано за любое конечное число шагов.

Унификаторы для задач (целей)  $a$  и  $a'$  определяются следующим образом. Если для сети задач имеется:  $a \in \{achieve[result']; provide[l', flag']; do[f(x_1, \dots, x_k)]; perform[t(x_1, \dots, x_k)]\}$ ,  $a' \in \{achieve[result']; provide[l', flag']; do[f(y_1, \dots, y_k)]; perform[t(y_1, \dots, y_k)]\}$  существует такая подстановка  $q$ , что  $(lq = l'q$  и  $flag = flag')$ ,  $resultq = result'q$ ,  $f(x_1q, \dots, x_kq) = f(y_1q, \dots, y_kq)$  или  $t(x_1q, \dots, x_kq) = t(y_1q, \dots, y_kq)$ , соответственно, то  $a$  и  $a'$  называются унифицируемыми, т.е.  $aq = a'q$ , а  $q$  – унификатором.

В случае, если существует множество задач  $a'_i \in A$ , для которых  $aq = a'_i$ , то для решения задачи нахождения ближайшей к  $a'$  цели в  $A$  предлагается решать с помощью алгоритма AASM.

Используя изложенное выше, мы можем описать процедуру ИСП БД следующим образом: (рис. 3). В соответствии с алгоритмом AASM шаги 1 и 2 формально можно представить так: вход: текущая задача (ситуация)  $a_i$ ; выход – прецедент  $\bar{a}_i$ , близкий к текущей ситуации.

Шаг 1. Идентификация атрибутов текущей задачи (ситуации)  $a_i$  посредством присвоения значений ключевым индексам:

$$a_i : y_l = (y_{l1}, \dots, y_{lN}) \in C = F_1 \times \dots \times F_N.$$

Шаг 2. Нахождение в базе знаний ( $\bar{C} = \{x_1, \dots, x_m\}, \bar{C} \subset C$ ) прецедентов

$$a_i : x_i = (x_{i1}, \dots, x_{iN}) \in \bar{C}, \text{ схожих ситуаций } a_i, \text{ т.е. имеющих:}$$

$$\min_{x_i \in \bar{C}} |d(x_i, y) = \left( \sum w_{ij} d_j(x_{ij}, y_j)^p \right)^{1/p}, \text{ где } w_{ij} = \begin{cases} p_{ij}, & \text{если } -x_{ij} \geq y_j - u - F_j = [0,1] \\ q_{ij}, & \text{если } -x_{ij} < y_j - u - F_j = [0,1] \\ p_{ij} = q_{ij}, & \text{если } -F_j \text{ конечно} \end{cases}$$

Шаг 3. Адаптация прецедента, т.е. модификация варианта решения, представленного в прецеденте, для его согласования с новой ситуацией по специальным правилам.

Шаг 4. коррекция полученного решения автоматически с использованием критики или с участием пользователя.

## **Выводы**

1. Реализован механизм генерации тактических планов боевых действий, в которых предусматривается возможность одновременного (параллельного) выполнения нескольких боевых задач.

2. Введен новый тип задач — обеспечивающая задача.

3. С учетом специфики исследуемой предметной области изменен механизм автоматического формирования плана. Так, в HTN-алгоритме выполня-

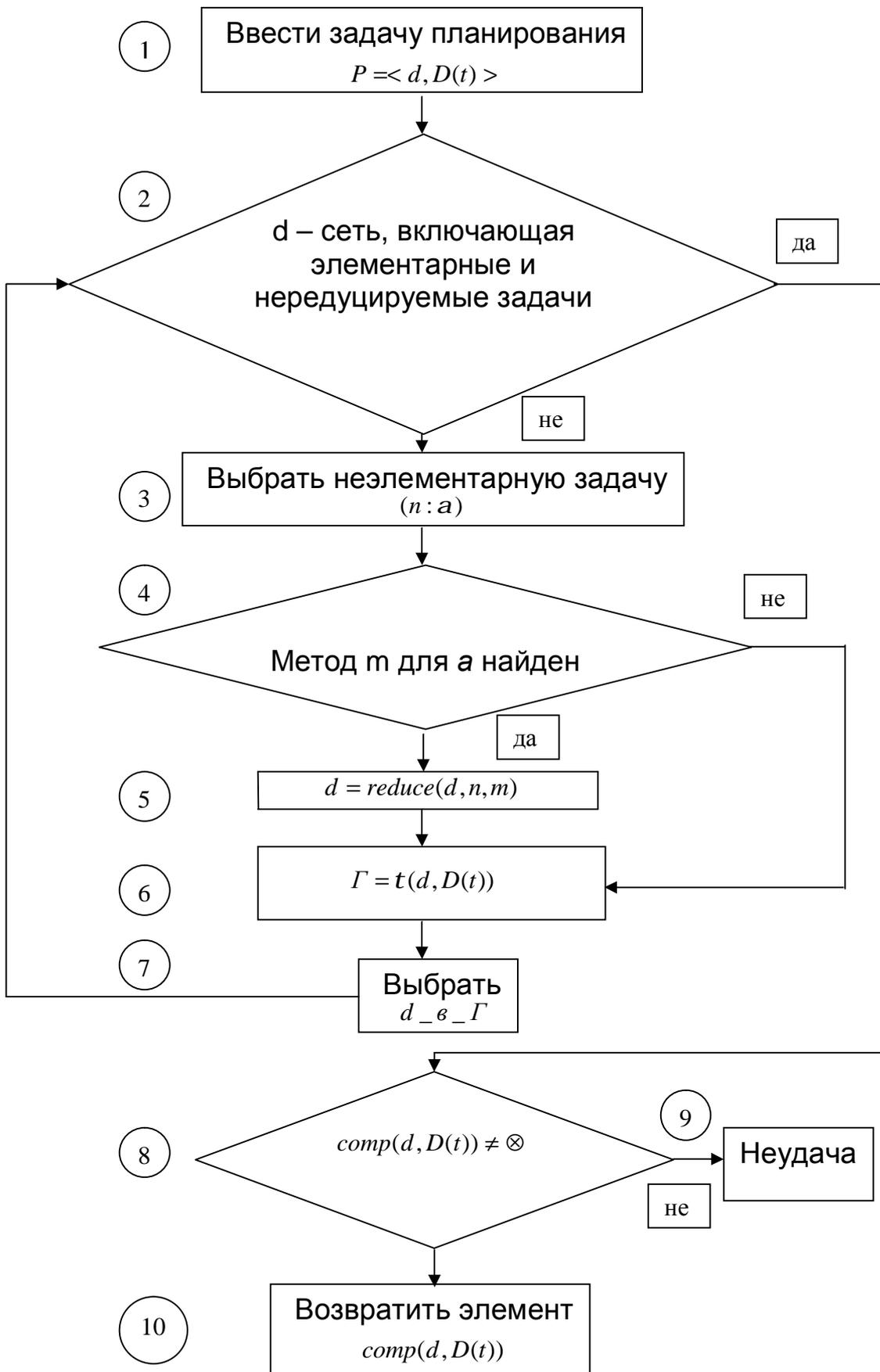


Рис. 3.Формальная интерпретация алгоритма ИСП БД

ется последовательный перебор всех элементарных (не декомпозируемых) задач и автоматически определяется (с учетом введенных ограничений) порядок их реализации.

4. Поиск метода решения боевой задачи производится не последовательным перебором имеющихся в базе данных методов, а используется критерий близости с прецедентом в базе знаний.

5. Предложен критерий поиска аналогичных прецедентов с использованием анизотропной и асимметричной метрики.

С учетом отмеченных выше отличий ИСП БД и для обеспечения описания процесса формирования тактических планов с использованием прецедентов авторам был дополнен синтаксис языка и адаптирована семантика, лежащие в основе алгоритма.

### Список литературы

1. Терехнев В.В. Управление силами и средствами на пожаре: Учебн. пособие / В.В Терехнев, А.В.Терехнев – М.: Академия ГПС МЧС России, 2003. – 261 с:
2. Munoz-Avila H., AhaD., BallasJ. BreslowL., Nau D. *Using guidelines to constrain interactive case-based HTN planning*. Tech. Report AIC-99-004, Naval Center for Applied Research on AI, Naval Research Lab., Washington, DC, 1999:  
<http://www.aic.nrl.naw.mil/papers/1999/AIC-99-004.ps.Z>
3. Erol K., Nau D, Hendler J., Tsuneto R. *A Critical Look at Critics in HTN Planning*. In IJCAI-95, Montreal, August, 1995:  
<http://www.es.umd.edu/users/kutluhan/Papers/IJC AI-95 .ps>
4. Munoz-AvilaH, AhaD.W., Nail D.S., Weber R., Breslow L, YamanF. *SiN: Integrating Case-based Reasoning with Task Decomposition*. Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA: Morgan Kaufmann, August 2001. pp. 999 -1004:  
<http://www.cs.umd.edu/~nau/papers/SiN-IJCAI-2001 .pdf>
5. Tsuneto R., Erol K., Hendler J., Nau D. *Commitment Strategies in Hierarchical Task Network Planning*. In AAAI-96, Portland, August, 1996:  
<http://www.cs.umd.edu/users/reiko/aaai96.ps>