# Способ и программные средства структурно-параметрической настройки нечетких когнитивных моделей основе генетических алгоритмов

Рассматриваемая в работе нечеткая когнитивная модель задается в виде ориентированного взвешенного графа. Вершины графа соответствуют концептам – значимым для цели моделирования факторам предметной области, а дуги отражают причинно-следственные связи между концептами. В качестве диапазона значения силы связи используется интервал [-1, 1], где -1 – самая сильная отрицательная связь, а 1 – самая сильная положительная. Каждому концепту дополнительно присваивается значение из интервала [0, 1], отражающее его состояние в начальный момент времени.

Из всех концептов нечеткой когнитивной модели выделяют управляющие и целевые. Первые отражают факторы, на которые можно повлиять и скорректировать таким образом поведение и конечное состояние моделируемой системы. Вторые представляют собой важные для исследования факторы, которые невозможно изменить непосредственно. Конечное состояние целевых концептов зависит от начального состояния управляющих.

Среди связей нечеткой когнитивной модели также выделяют управляющие связи. Значение силы этих связей не известно исследователю заранее, а может изменяться в некотором разрешенном диапазоне. Изменение силы управляющей связи приводит к изменению конечного состояния целевых концептов в результате моделирования.

Структурно-параметрическая настройка нечеткой когнитивной модели представляет собой подбор начального состояния ее управляющих концептов, а также значений управляющих связей таким образом, чтобы в результате ее моделирования по времени, целевые концепты максимально приблизились к желаемому состоянию. Настройка начального состояния управляющих концептов отражает параметрическую оптимизацию, а настройка силы управляющих связей – структурную. Желаемое конечное состояние задается экспертом исходя из целей моделирования. Моделируемая предметная область может иметь определенные ограничения, поэтому эти ограничения должны быть отражены в ограничениях на состояние управляющих концептов.

Не все когнитивные модели являются устойчивыми, поэтому целесообразно выбирать максимальное модельное время. Также моделирование по времени может осуществляться различными способами, поэтому необходимо предоставить возможность выбирать способ, подходящий для конкретной модели.

Таким образом, можно обозначить входные данные для структурно-параметрической оптимизации:

* A – матрица смежности когнитивной модели для предметной области;
* Uv – вектор управляющих концептов;
* Rv – вектор ограничений, накладываемых на управляющие концепты;
* Ua – вектор управляющих связей;
* Ra – вектор ограничений, накладываемых на управляющие связи;
* Dv – вектор моделей динамики для отдельных концептов;
* X0 – вектор начального состояния концептов;
* G – вектор целевых концептов;
* Y – вектор желаемого состояния целевых концептов;
* D – модель динамики для когнитивной модели;
* T – максимальное модельное время.

В качестве примера возможных исходных данных рассмотрим когнитивную модель «Управление ИТ-проектом» (рисунок 1).



Рисунок 1 – Управление ИТ-проектом

Матрица смежности A рассматриваемой когнитивной модели представлена в таблице 1.

Таблица 1 – Матрица смежности A рассматриваемой когнитивной модели

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | К1 | К2 | К3 | К4 | К5 | К6 | К7 | К8 | К9 | К10 |
| К1 | 0 | 0 | 0 | 0,2 | -0,35 | 0,1 | 0 | 0 | 0 | 0 |
| К2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| К3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| К4 | 0 | -0,1 | 0,3 | 0 | -0,35 | 0 | 0 | 0 | 0 | 0 |
| К5 | 0 | 0,7 | 0,7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| К6 | 0 | 0 | 0 | 0 | -0,15 | 0 | 0 | 0 | 0 | 0 |
| К7 | 0 | 0 | 0 | 0 | -0,2 | 0 | 0 | 0 | 0,5 | 0 |
| К8 | 0 | 0 | 0 | 0 | -0,15 | 0 | 0 | 0 | 0 | 0 |
| К9 | 0,1 | 0,3 | -0,3 | 0 | -0,15 | 0,1 | 0 | 0,05 | 0 | 0 |
| К10 | 0 | 0,8 | -0,7 | 0 | -0,2 | 0 | -0,1 | -0,3 | -0,1 | 0 |

Управляющими концептами являются: «Взаимодействие с заказчиком», «Планирование», «Число сотрудников», тогда Uv = {К1, К4, К10}.

Пусть взаимодействие с заказчиком является хорошим с небольшой возможность увеличения, планирование является средним со средней возможностью улучшения, а число сотрудников является средним с возможностью изменения в обоих направления, тогда Rv = {0,8 – 0.9; 0,5 – 0,8; 0,3 – 0,7}.

Также известно, что влияние числа сотрудников на квалификацию команды и гибкие методологии может быть немного скорректировано путем внедрения различных политик найма сотрудников, поэтому Ua = {К10 – К8, К10 – К9}, Ra = {-0,4 – -0,2; -0,15 – 0,15}.

Вектор начального состояния концептов X0 представлен в таблице 2.

Таблица 2 – Вектор начального состояния концептов X0

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| К1 | К2 | К3 | К4 | К5 | К6 | К7 | К8 | К9 | К10 |
| 0,8 | 0,5 | 0,5 | 0,6 | 0,4 | 0,7 | 0,3 | 0,6 | 0,5 | 0,5 |

Целевыми концептами являются «Стоимость проекта», «Временные затраты», тогда G = {К2, К3}.

Желаемое состояние целевых концептов формулируется в виде значений или диапазонов. Пусть необходимо сократить стоимость при сохранении разумных временных затрат, тогда Y = {0,0; 0,5}.

Существует несколько моделей динамики для нечетких когнитивных моделей:

(D1)

(D2)

(D3)

(D4)

Пусть в нашем случае D = D4, а Dv = ∅.

Максимальное модельное время T определяется число итераций при моделировании по времени, зададим T = 100.

Структурно-параметрическая настройка выполняется с использованием генетических алгоритмов, поэтому к входным данным необходимо добавить настройки генетического алгоритма:

* N – размер поколения;
* S – условие остановки.

В качестве условия остановки могут выступать в зависимости от выбора эксперта абсолютная погрешность, максимальное число поколений или максимальное число поколений без улучшений. Зададим для нашего примера N = 100, S = 0.01 (абсолютная погрешность).

Для реализации генетического алгоритма требуется алгоритмы генерации новой особи, селекции, определения приспособленности, скрещивания и мутации. Рассмотрим псевдокод каждого из алгоритмов.

Алгоритм 1: Генерация новой особи

Input: Uv, Rv, Ua, Ra

Output: C – chromosome

Algorithm:

uv = {};

for u, r in zip(Uv, Rv) {

uv[u] = random that >= r.min and <= r.max;

}

ua = {};

for u, r in zip(Ua, Ra) {

ua[u] = random that >= r.min and <= r.max;

}

C = { uy, ua };

Данный алгоритм создает особь, значения управляющих концептов и связей которой удовлетворяют наложенным ограничениям. Например, для предложенных исходных данный алгоритм может сгенерировать следующую особь: C = { uv: { 'К1': 0.85, 'К4': 0.63, 'К10': 0.52 }, ua: { 'К10 – К8': -0.21, 'К10 – К9': -0.11 } }.

Алгоритм 2: Селекция

Input: Ci[] – array of chromosomes with length N

Output: Co[] – array of chromosomes with length N

Algorithm:

Co = [];

for i in range(N) {

c1 = random c from Ci[];

c2 = random c from Ci[];

if fitness of c1 > fitness of c2 {

add c1 to Co;

} else {

add c2 to Co;

}

}

Данный алгоритм представляет из себя турнирный отбор из 2 особей на каждой итерации.

Алгоритм 3: Определение приспособленности

Input: A, С, X0, G, Y, D, Dv, T

Output: F – fitness

Algorithm:  
state = X(0) ∩ С.uv;

for t in range(0, T) {

next\_state = state copy;

for r in A {

Dr = if C in Dv { Dv[C] } else { D };

new\_state[r.name] = Dr(r, С.ua, t, state);

}

state = new\_state;

}  
Y(s) = state ∩ G;

F = 1/Σ(Y(s) – Y)^2;

Данный алгоритм осуществляет моделирование по времени особи, получая тем самым ее конечное состояние. Приспособленность особи оспеределяется как обратная сумма квадратов разностей между рассчитанным и желаемым конечными состояниями.

Алгоритм 4: Скрещивание особей

Input: I1, I2, Rv, Ra

Output: I3

Algorithm:

uv = {};

for u in Uv {

p1 = C1.uv[u];

p2 = C2.uv[u];

if (p1 > p2) {

(p1, p2) = (p2, p1);

}

min = max(p1 – 0.5 \* (p2 – p1), Rv[u].min);

max = min(p2 + 0.5 \* (p2 – p1), Rv[u].max);

uv[u] = random that >= min and <= max;

}

ua = {};

for u in Ua {

p1 = C1.ua[u];

p2 = C2.ua[u];

if (p1 > p2) {

(p1, p2) = (p2, p1);

}

min = max(p1 – 0.5 \* (p2 – p1), Ra[u].min);

max = min(p2 + 0.5 \* (p2 – p1), Ra[u].max);

ua[u] = random that >= min and <= max;

}

I = { uy, ua };

Данный алгоритм представляет из себя скрещивание смешиванием.

Алгоритм 5: Мутация особи

Input: C

Output: null

Algorithm:

if random > 0.5 {

u, r = random (u, r) from zip(Uv, Rv);

C.uv[u] = random that >= r.min and <= r.max;

} else {

u, r = random (u, r) from zip(Ua, Ra);

C.ua[u] = random that >= r.min and <= r.max;

}

Данный алгоритм изменяет значение случайно выбранного концепта или связи на новое, которое удовлетворяет наложенным ограничениям.