

РЕАЛИЗАЦИЯ МОДЕЛИ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ ОПТИМИЗАЦИИ СЕВООБОРОТОВ НА ЯЗЫКЕ PROLOG

Н.М. Светлов

Цель статьи состоит в описании программной реализации модели динамического программирования для оптимизации севооборотов [1] и рассмотрении методических аспектов её применения.

Удобным средством реализации моделей динамического программирования являются Prolog-системы [2]. Рекуррентное соотношение модели оптимизации севооборотов имеет вид

$$q(c_t) = \max_{c_{t-1} \in F(c_{t-1})} (q(c_{t-1}) + p(c_{t-1}, c_t)), t=1...T,$$

где t — номер поля в севообороте, T — число полей в севообороте, c_t — культура, занимающая t -е поле в 1-й год использования севооборота, c_{t-1} — культура, занимающая поле $t-1$ в тот же год, (культура c_t является её предшественником), $p(c_{t-1}, c_t)$ — чистый доход с 1 га культуры c_{t-1} , выращиваемой после культуры c_t . Это соотношение формализовано на языке Prolog триарным правилом **формула**. Первый домен его содержит наименование культуры, которую планируется выращивать на первом поле севооборота, второй — номер поля (нумерация полей в данном случае обратная), третий — число полей, на которых ещё предстоит разместить культуры. Версия этого правила приведена ниже.

```
формула(Первая, 1, _) :- retractall(q(_, _)),
    assert(q(1, [Первая], 0)), !.
формула(Первая, НомерПоля, ОсталосьПолей) :- НомерПоля1
    is НомерПоля - 1, ОсталосьПолей1 is ОсталосьПолей +
    1, формула(Первая, НомерПоля1, ОсталосьПолей1),
    retract(q(НомерПоля1, [Культура | Культуры],
    НакопленныйЧистыйДоход)), предшественник(Культура,
    Предшественник, ЧистыйДоход),
    НакопленныйЧистыйДоход1 is НакопленныйЧистыйДоход +
    ЧистыйДоход, замена(НомерПоля, ОсталосьПолей,
    Первая, [Предшественник, Культура |
    Культуры], НакопленныйЧистыйДоход1), fail.
формула(_, _, _).
```

Для первого поля правило порождает триарный факт **q**, содержащий в своих доменах соответственно номер поля, список уже размещённых культур, величину суммы чистых доходов с 1 га культур, вошедших в список. Предварительно удаляются все факты **q**, порождённые предшествующей работой программы (если они существуют). Для других полей сначала посредством рекурсивного обращения к правилу **формула** строятся наилучшие последовательности культур от первого поля до предшествующего данному. Далее последовательно происходит консультация с фактами **q** для предшествующего поля (использованные правила **q** тут же удаляются) и триарными фактами **предшественник**¹, первый домен которых содержит культуру, второй — её предшественника, третий — значение соответствующего этой паре чистого дохода с 1 га. На основе консультации строится возможный список размещения культур по полям от первого до данного, который передаётся правилу **замена**.

Правило **замена** проверяет, существует ли допустимая последовательность культур, начинающаяся от последней размещённой. Если нет, данный список размещённых культур игнорируется. В противном случае проверяется, есть ли уже факт **q** для данного года, хранящий список размещённых культур, последняя среди которых та же, что и в анализируемом списке. Если такого факта нет, он порождается на основе текущего списка и соответствующей ему величины накопленного чистого дохода. Если есть, то он заменяется на факт с текущими списком и накопленным чистым доходом только в том случае, если накопленный чистый доход анализируемого списка больше. Ниже приведена реализация правила **замена**.

¹ Для удобства работы пользователя можно присвоить этим фактам более короткое имя либо предусмотреть программный интерфейс для редактирования базы фактов "предшественник".

```
замена(_,ОсталосьПолей,Первая,[Культура|_],_) :- пот
допустимая_последовательность(Культура,Первая,
ОсталосьПолей),!.
замена(НомерПоля,_,_,[Культура|_],
НакопленныйЧистыйДоход):-q(НомерПоля,
[Культура|_],НЧД1),НЧД1>НакопленныйЧистыйДоход,
!.
замена(НомерПоля,_,_,[Культура|Культуры],
НакопленныйЧистыйДоход):-retract(n_(НомерПоля,
[Культура|_],_)),assert(q(НомерПоля,
[Культура|Культуры],НакопленныйЧистыйДоход)),!.
замена(НомерПоля,_,_,[Культура|Культуры],
НакопленныйЧистыйДоход):-assert(q(НомерПоля,
[Культура|Культуры],НакопленныйЧистыйДоход)),!.
```

Правило **допустимая_последовательность** проверяет существование допустимой последовательности от последней размещённой культуры до культуры на первом поле. Оно может быть реализовано следующим образом:

```
допустимая_последовательность(Культура,Первая,1):-!,
предшественник(Культура,Первая,_).
допустимая_последовательность(Культура,Первая,
ОсталосьПолей):-ОсталосьПолей1 is ОсталосьПолей-
1,предшественник(Культура,Культура1,_),
допустимая_последовательность(Культура1,Первая,
ОсталосьПолей1).
```

Результат применения правила **формула** — один или более фактов q^2 . Для удобства пользователя целесообразно использовать интерфейс к правилу **формула**, включающий нижеприведённый набор правил (возможны и другие варианты):

² Если правило "формула" не создало ни одного факта q , значит, из данного множества пар "культура-предшественник" невозможно составить севооборот с указанным числом полей, т.е. множество решений задачи оптимизации севооборота пусто.

```
севооборот(ЧислоПолей,Первая,Севооборот,
ЧистыйДоходНаГектар):-findall(Вариант/
ЧистыйДоход,интерфейс(Первая,Вариант,ЧистыйДоход,
ЧислоПолей),ВсеВарианты),max(Севооборот/
НаибольшийЧистыйДоход,ВсеВарианты),
ЧистыйДоходНаГектар is НаибольшийЧистыйДоход /
ЧислоПолей.
интерфейс(Первая,[Культура|Культуры],ЧистыйДоход,
ЧислоПолей):-формула(Первая,ЧислоПолей,1),
retract(q(_, [Культура|Культуры],
НакопленныйЧистыйДоход)),
предшественник(Культура,Первая,ЧД),ЧистыйДоход is
НакопленныйЧистыйДоход + ЧД.
max(X,[X]) :- !.
max(M,[Y|Z]) :- max(M1,Z),better(M1,Y),M = M1, ! ; M
= Y.
better(A / B,D / E) :- B > E.
```

Для определения оптимального севооборота, состоящего из заданного числа полей при условии, что культура для первого поля задана, достаточно ввести в ответ на приглашение Prolog-системы запрос, подобный следующему.

```
севооборот(10, "Озимая пшеница", X, D).
```

В ответ выдаются значения переменных X и D . Первая из них содержит список культур, составляющих наилучший возможный десятипольный севооборот (в хронологическом порядке), первое поле в котором занято озимой пшеницей. Вторая — математическое ожидание чистого дохода с 1 га этого севооборота в год.

Реализация рассматриваемой модели была осуществлена в демонстрационной Prolog-системе Prolog Inference Engine фирмы Borland International. В базу фактов было включено 30 фактов **предшественник**. Тестирование программы производилось на персональной ЭВМ IBM AT 486SX-S с тактовой частотой центрального процессора 33 МГц (без мате-

матического сопроцессора). Оно показало, что формирование оптимального севооборота из 5 полей заняло 1,10 сек. при использовании динамического программирования против 2,74 сек. при использовании метода полного перебора. Для десятипольного севооборота затраты времени составили 6,26 сек. против 8 мин. 15,49 сек. методом полного перебора. Оптимальный двадцатипольный севооборот был обнаружен за 27,40 сек. Время, затрачиваемое на его построение методом полного перебора, так и осталось не установленным.

Приведённая выше рекуррентная формула отражает простейшую постановку задачи оптимизации севооборотов, не учитывающую многие технологические и агрономические ограничения. Для реализации модели с ограничениями [1] необходимо добавить к программе правила, осуществляющие проверку требуемых ограничений. Обращение к ним из правила **допустимая_последовательность** позволит сохранить в базе фактов **q** наилучшие варианты достижения состояний, возможных на данном этапе рекурсивных вычислений, исключая те, для которых выполнение ограничения невозможно. В каждом факте **q** должны быть предусмотрены дополнительные домены для хранения достигнутого значения каждого ограничения.

Приведённые выше правила можно легко модифицировать для разработки оптимального плана освоения нового севооборота и для поиска оптимального севооборота без ограничения по числу полей на основе подходов, рассмотренных в [1].

Большое значение для формирования высокоэффективного севооборота имеет выбор правил **предшественник**. Совокупность культур, выращивание которых допустимо в рамках данного севооборота, и их связь с предшественниками во многом определяют получаемое решение. Формируя множество допустимых культур, следует учитывать не только соответ-

ствие их почвенным, климатическим и другим естественным условиям, но и характерные для полей севооборота и соседних с ними флору и фауну, особенности ведения животноводства в данном хозяйстве и соседствующих с ним предприятиях и личных хозяйствах граждан, возможности охраны урожая, его хранения и вывоза и т.п.

Круг допустимых предшественников каждой культуры в рамках перечисленных условий должен быть возможно более широк. Отсутствие информации о математическом ожидании чистого дохода ни в коем случае не должно быть основанием исключения предшественника из списка возможных. Лучше использовать самую грубую оценку нижней границы этой величины, чем исключать возможный предшественник. Если этот предшественник действительно используется (согласно решению) в оптимальном севообороте, появляется существенный стимул к уточнению величины соответствующего чистого дохода. Исключение некоторых предшественников может быть хорошим способом корректировки оптимального севооборота, не отвечающего некоторым технологическим и (или) агрономическим ограничениям.

Библиографический список

1. Светлов Н.М. Модели динамического программирования для оптимизации севооборотов // Труды Независимого Аграрно-Экономического Общества России. Вып. I: Проблемы формирования аграрного рынка России. М.: Изд-во МСХА, 1997.
2. Братко И. Программирование на языке Пролог для искусственного интеллекта: Пер. с англ. — М.: Мир, 1990.