

ПОИСК РЕШЕНИЙ СИСТЕМАМИ КОНКУРИРУЮЩИХ АВТОМАТОВ

Н.В.Хаджинова, М.П.Ревотюк, Белорусский государственный университет информатики и радиоэлектроники

Объект рассмотрения – методы спецификации задач поиска оптимальных многошаговых решений, формализуемых в терминах поиска кратчайших путей на конечных множествах вариантов. Цель исследования – разработка шаблона базового класса представления таких задач в рамках объектно-ориентированных технологий, допускающего распараллеливание процесса решения на сети агентов [1,2].

Анализ методов поиска кратчайших путей показывает, что общей схемой поискового алгоритма (простого перебора, метода динамического программирования, ветвей и границ и др.) является волновой характер сканирования пространства решений. Статическим представлением процесса решения является структура данных $\langle V, R, P, s, f \rangle$, где V – множество состояний, подлежащих упорядочению построением дерева путей; P – номер предшествующего состояния; R – потенциал состояния или оценка по заданному критерию, s и f – исходное и целевое состояния, $s, f \in V$.

Первоначально положим $R_i = \infty$, $P_i = i$, $i \in V$, тогда оптимальное решение можно найти обратным движением из состояния f : $\{s, P \dots (P(P \dots (P_f))) \dots P(P_f), P_f, f\}$.

Волновой процесс на очереди императивно определяемых состояний Q из начального состояния s пусть моделируется рекуррентным алгоритмом

```
for (Q={s}; !Empty(Q); Wakeup(Q));
```

Порождение волны из любого состояния x здесь представлено следующей функцией, учитывающей лишь факт достижения этого состояния

```
template <class Node> void Wakeup(Node x) {  
    if (x!=f) {  
        Iterator next(x); // Итератор альтернатив развития волны  
        while (y=next()) Plan(x, y);  
    } else while (!Empty(Q)) Destroy(Q);  
}
```

Здесь функция планирования отражает намерение реализации элементарного автоматного перехода $x \rightarrow y$, $x, y \in V$

```
template <class Node, class Estimation> void Plan(Node x, Node y) {  
    Estimation Rxy=R(x)+W(x, y); // Оценка плана перехода x->y  
    if (R(y)>Rxy) {  
        Q-=y; // Уничтожение конкурента  
        R(y)=Rxy, P(y)=x;  
        Q+=y; // Утверждение плана перехода x->y  
    }  
}
```

Таким образом, процесс волнового поиска привязан лишь к моментам выхода автоматов из активного состояния. Другие состояния доступны для наблюдения и анализа. В результате спецификация задачи может быть представлена в рамках удобной для практической реализации компонентной модели PME (Properties, Methods, Events).

Литература

1. Ревотюк М.П., Кузнецова Н.В. Агентная система кооперации ресурсов вычислительной среды для решения задач выбора//Известия Белорусской инженерной академии, № 1(15)/1, 2003. – С. 265-268.
2. Ревотюк М.П., Кузнецова Н.В. Оценка эффекта распараллеливания жадных алгоритмов//Известия Белорусской инженерной академии, № 1(17)/3, 2004. – С.206-209.