



системы» / Владим. гос. ун-т ; сост. С. А. Сорокин. – Владимир : Изд-во Владим. гос. ун-та, 2006 – 36 с.

А.П. Михайлов

АЛГОРИТМ ОПТИМАЛЬНОЙ СТРУКТУРИЗАЦИИ ДАННЫХ ПО ПРОГРЕССИВНОСТИ

(Поволжский государственный университет телекоммуникаций
и информатики)

Особенностью задач принятия многокритериальных решений является большое количество допустимых вариантов решений. Задача лица, которое принимает решение, состоит в том, чтобы структурировать это множество, выделив наиболее эффективные решения, а также последовательно расположив их по эффективности. При этом практически все известные методы включают в себя в той или иной мере субъективный момент (например, в теории полезности. мысленная лотерея при парном сравнении критериев эффективности между собой).

Наиболее объективным методом структуризации множества решений является метод расслоения данного множества на слои, с последующим выделением подслоев используя принцип Парето-оптимальности. В соответствии с ним, нужно выделить 1-й слой оптимальных по Парето вариантов (элементов), после чего убрать их из рассмотрения и в оставшемся подмножестве выделить 2-й, 3-й слой и т.д. Однако данный метод имеет и недостаток – разрешающая способность метода невысока, т.к. каждый рассмотренный слой может содержать достаточно большое количество объектов.

С.А. Пиявский («Онтология проектирования» №1(7), 2013) предложил усилить метод послыного расслоения множества вариантов в многокритериальных задачах принятия решений, используя, вместе с понятием Парето-оптимальности, понятие прогрессивности вариантов решений [1]. В данном случае после того, как будет выделен слой оптимальных по Парето вариантов, внутри этого слоя рассматривается подслой элементов оптимальных по прогрессивности, они убираются из рассмотрения, из оставшихся элементов выделяется следующий подслой и т.д. В настоящем докладе представлен алгоритм оптимальной структуризации данных по прогрессивности.

Представлена блок-схема разработанного алгоритма нахождения прогрессивных вариантов (рис. 1) для слоя Парето-оптимальных вариантов. На схеме изображен «Блок 1», он отвечает за подсчет количества вариантов и критериев, задавая размерность будущего массива данных. После подсчета размерности массива алгоритм переходит к «Блоку 2». В данном блоке отображен процесс отбора Парето-оптимальных критериев. В «Блоке 3» формируется новая матрица, которая состоит только из оптимальных по



Парето вариантов. Помимо создания матрицы оптимальных вариантов подсчитывается размерность нового массива. «Блок 4» содержит в себе подсчет матриц прогрессивности для каждого варианта. Для этого попарно сравниваются вектора строки матриц и определяется минимальный вектор (по каждой компоненте), добавление которого к вектору первой строки делает ее неэффективным по Парето в сравнении с вектором второй из сравниваемых строк.



Рис. 1. Блок-схема алгоритма нахождения прогрессивных вариантов

Так формируется множество матриц, их количество соответствует числу Парето-оптимальных вариантов данного слоя. После того, как сформировались все матрицы прогрессивности, алгоритм переходит к «Блоку 5», который отражает процесс обработки полученных матриц на нахождение Парето-оптимальных вариантов. Этот блок необходим для того, чтобы отсеять менее прогрессивные варианты. Более же прогрессивные варианты в «Блоке 6» переносятся и объединяются в новую матрицу. Согласно «Блоку 7», данная матрица



обрабатывается на Парето-оптимальность и т.д. Процесс повторяется до тех пор, пока не исчерпаются все варианты решения.

Таким образом, исходное множество вариантов решения разделяется на последовательность слоев Парето-оптимальных вариантов убывающей эффективности, при этом некоторые из этих слоев дополнительно могут расслаиваться по прогрессивности.

Литература

1. Пиявский С.А. Два новых понятия верхнего уровня в онтологии многокритериальной оптимизации // Онтология проектирования, 2013. №1(7). - С. 65-85.
2. Пиявский С.А. Прогрессивность многокритериальных альтернатив // Онтология проектирования, 2013. №4(10). - С. 60-71.

С.А. Никулин, Д.Ю. Полукаров

НЕКОТОРЫЕ ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ СЕТЕЙ С ЯЧЕЙСТОЙ ТОПОЛОГИЕЙ

(Самарский университет)

Введение

Самоорганизующиеся сети в наше время играют очень большую роль. С ними можно столкнуться в промышленности, в городской среде. Хорошим примером является «интернет вещей». Такие сети не имеют постоянной структуры и могут изменяться буквально на лету. Узлы в сети имеют связь только со своими соседями – всей структуры сети они не знают. Каждый узел сети может выполнять функцию ретранслятора сигнала, что повышает ее отказоустойчивость: в случае поломки или выхода из строя одного из узлов – сигнал может пойти по альтернативному маршруту и дойти до своего адресата. Примером самоорганизующихся сетей являются mesh-сети, ad-hoc сети, сенсорные сети.

Задачи разработки новых протоколов маршрутизации характерны для любых типов сетей [2],[3]. Часто встает вопрос об организации протокола маршрутизации и в сетях с ячейистой топологией. Разработать его довольно сложно, ведь устройства находятся в постоянном движении. Одним из хороших решений является алгоритм «жадного продвижения» (Greedy forwarding algorithm).

Алгоритм жадного продвижения

Жадное продвижение — алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным. Т.е с каждым новым локальным решением, мы должны приближаться к конечному решению.

В нашем же случае, мы должны найти наиболее короткий путь передачи сигнала из точки А в точку В (за наименьшее количество хопов).