

Использование муравьиного алгоритма для создания системы принятия экономических решений в автоматной модели производства

Введение

В работе [1] приведена вычислимая экономическая модель, представляющая процесс производства в виде конечного автомата. В данной работе описывается система принятия экономических решений в рамках указанной модели, использующая муравьиный алгоритм [2]. В работе описывается мотивация и излагается результат использования указанного алгоритма.

Постановка задачи

Приведем список требований к поведению экономических агентов:

- ограниченная интеллектуальность;
- ограниченные знания об окружающей среде;
- простота поведения каждого экономического агента;
- субоптимальность принимаемых агентами решений (следствие предыдущих пунктов).

Цель работы – разработка алгоритма принятия экономических решений в автоматной модели, отвечающего сформулированным требованиям, на основе использования муравьиного алгоритма.

Краткое описание модели

Процесс производства любого товара можно разделить на этапы, проходя через которые изделие превращается из заготовки в товар, пригодный к употреблению. Таким образом, моделируя производство удобно выделять *состояния* готовности товара. В процессе производства заготовка постоянно претерпевает изменения, переходит из одного состояния готовности в другое. Следовательно, можно выделить *переходы* между состояниями (их же иногда называют *ребрами*).

Такие рассуждения приводят к выводу: процесс производства можно моделировать при помощи *конечного автомата*. Более того, в такой модели очень легко интерпретируются издержки, необходимые для того, чтобы изделие перешло из одной стадии готовности в другую. Это *входные воздействия* конечного автомата. *Выходным воздействием* автомата будет сам готовый товар. Отметим, что и в реальном и в моделируемом мире *товар*, изготавливаемый на одном производстве, может являться *сырьем* для другого производства. Эти два понятия обозначают одно и то же: *ресурсы*. В настоящей работе будет упоминаться слово «сырье», когда речь идет о его использовании в производстве и «товар», когда речь идет о готовом изделии (результате производства). Слово «ресурс» будет употребляться, когда нет смысла специально отмечать: речь идет о потреблении или изготовлении.

В итоге получаем следующую модель. Стадии готовности товара — состояния конечного автомата. Процесс обработки заготовки (переход от одной стадии готовности в другую) – переход конечного автомата. Каждый моделируемый экономический агент использует свой конечный автомат для производства. Помимо агентов в модельном экономическом мире существует *рынок* – место, где агенты покупают и продают *ресурсы*. Агенты взаимодействуют посредством рынка. Более подробное описание можно найти в работе [1].

Таким образом, имеет место автоматная мультиагентная система. Существуют примеры применения систем такого класса в компьютерной анимации [3] и разработке реактивных систем [4]. Однако этот подход не применялся в экономике.

Перейдем к описанию того, какие экономические решения принимают агенты, и какой алгоритм принятия решений использовался в работе [1]. Для этого введем следующее определение. Под *развилкой* будем понимать состояние конечного автомата, из которого выходит больше одного ребра.

Когда процесс производства приходит в развилку, агенту приходится принимать экономическое решение: куда двигаться дальше. От того, какой путь будет выбран, зависят издержки и результат производства, а, следовательно, и прибыль агента. Для принятия таких решений и будет использоваться муравьиный алгоритм.

Мотивация

В работе [1] предложен наиболее простой способ принятия подобных решений. Приведем его описание. Под *производственным циклом* будем понимать замкнутую последовательность переходов конечного автомата. В начале агент выбирает какой-либо производственный цикл и придерживается его. Периодически агент запускает процедуру анализа обстановки на рынке. Она состоит в том, что при помощи *обхода в глубину* [5] перебираются все возможные производственные циклы. Каждый цикл оценивается с точки зрения прибыльности. Если в результате найден более выгодный цикл, то агент далее придерживается уже его.

Отметим следующие недостатки предложенного подхода:

- субоптимальность моделируется не самым лучшим образом, так как перебираются все варианты. Единственным параметром субоптимальности является период между запусками процедуры анализа;

- алгоритм принятия решений не обладает свойством *real-time*: процесс анализа занимает много времени, однако запускается он периодически;

- расчеты, осуществляемые при анализе, являются плохо масштабируемыми;

- во время проведения анализа агент пользуется знаниями сразу обо всем своем производстве, желательно пользоваться более ограниченными знаниями.

В данной работе предлагается использовать *муравьиный алгоритм* (описание алгоритма приведено в следующем разделе). Достоинством является то, что его удобно использовать для анализа графов, а принятая в данной работе модель производства – конечный автомат – частный случай графа. Как будет показано далее, этот алгоритм позволит устранить перечисленные недостатки.

Описание алгоритма

Муравьиный алгоритм (*ant colony optimization, ACO*) – один из эффективных полиномиальных алгоритмов для поиска оптимальных маршрутов на графах. В частности, это наиболее популярный алгоритм поиска приближенного решения задачи коммивояжера [2]. В основе алгоритма лежит поведение муравьиной колонии – маркировка более удачных путей большим количеством феромона. В экономическом контексте очень легко проинтерпретировать работу с феромоном: чем больше прибыль, тем больше оставляемого феромона. Принимая решение, в каком направлении будет идти производство, агент руководствуется оставленным феромоном.

В итоге получаем такой алгоритм. В моделируемом экономическом мире будет существовать два типа участников: *агенты* и *муравьи*. Каждому агенту принадлежит несколько муравьев. Конечный автомат муравья совпадает с конечным автоматом его *хозяина* (агента, которому он принадлежит). И агенты и муравьи занимаются одним и тем же: покупают сырье, производят товары и получают прибыль. Различие состоит в том, что муравьи работают с «игрушечным» рынком. Это означает, что все их покупки и продажи не влияют на тот рынок, с которым работают агенты.

Муравьи считают свою прибыль и оставляют феромон на переходах конечного автомата. Чем больше была последняя прибыль муравья, тем больше он оставит феромона. Со временем оставленный муравьями феромон «испаряется». Поэтому следы феромона отображают актуальную информацию о том, где ожидаемая прибыль максимальна. Когда агенту необходимо делать выбор, по какому пути продолжать производство, он случайным образом выбирает переход конечного автомата, который ведет из текущего состояния. Причем вероятность каждого возможного варианта пропорциональна феромону, оставленному на данном переходе.

Выбор того, куда двигаться муравью производится следующим образом. Цель муравья состоит не в получении прибыли, а в изучении автомата. Поэтому он стремится совершать те переходы, которые ведут в наименее изученные вершины конечного автомата. Для описания алгоритма действий муравья потребуется понятие *ветвистости перехода* – число развилки, которые можно будет встретить по пути к терминальному состоянию, если осуществить данный переход. Ветвистость показывает то, насколько сложно муравью изучать подграф, располагающийся после этого перехода.

Пусть тогда муравей действует по следующему алгоритму: из данной вершины конечного автомата он выбирает случайный переход, таким образом, что вероятность каждого варианта пропорциональна его ветвистости. Действуя по такому правилу, муравей будет часто посещать те области конечного автомата, которые наиболее сложны для изучения. Этот принцип обеспечит то, что агент будет всегда располагать актуальной информацией обо всем конечном автомате.

На рис. 1 приведен пример автомата, изучаемого муравьем. Когда у муравья появляется выбор: идти из состояния 0 в состояние 1 или 3, то выбор состояния 3 он делает чаще, так как фрагмент графа {3, 4, 5, 6} содержит одну развилку, а {1, 2} – их не содержит. Поэтому фрагмент графа {3, 4, 5, 6} более сложен для анализа, чем фрагмент {1, 2}.

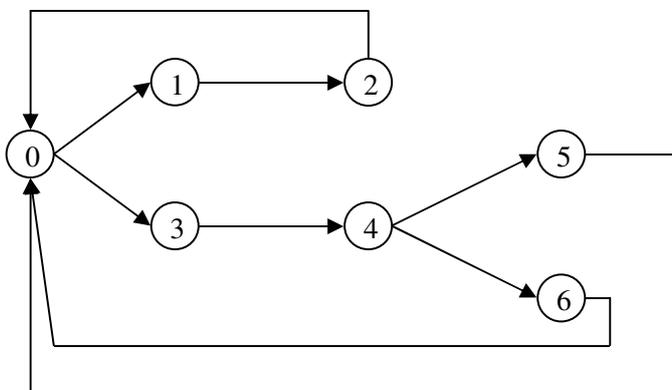


Рис. 1. Изучение муравьем конечного автомата

На рис. 2 приведена общая схема принятия агентами экономических решений при помощи муравьиного алгоритма.



Рис. 2. Общая схема алгоритма

Достоинства предлагаемого метода

В данном разделе приведен анализ того, насколько предложенный метод соответствует требованиям, сформулированным в начале статьи.

Муравьиный метод принятия решений о продвижении по автомату хорошо соответствует принципу ограниченной интеллектуальности экономического агента: муравьи оценивают прибыльность своего пути лишь приблизительно.

Так же выполняется свойство ограниченности знаний агента об окружающей среде: информация о том, какой путь приносит большую прибыль, поступает от муравьев с некоторой задержкой. Особенностью предлагаемого метода является то, что интеллектуальность агента легко регулируется: чем больше муравьев, тем больше точность оценок.

Говоря о простоте действий агентов, следует отметить, что набор муравьев является оценивающей системой, результатами которой руководствуется экономический агент. Эта система демонстрирует достаточно сложное поведение, однако действия каждого муравья крайне просты. Он лишь случайным образом «гуляет» по графу и оставляет больше феромона на путях, которые оказались более прибыльными.

В итоге получаем, что использование муравьиного алгоритма позволяет устранить недостатки, описанные в разделе «мотивация»:

- можно гибко регулировать субоптимальность поведения агентов;
- алгоритм обладает свойством *real-time*: число элементарных операций, выполняемых на каждом шаге для моделирования действий муравьев – одинаково;
- алгоритм моделирования поведения муравьев является хорошо масштабируемым;
- знания каждого муравья об окружающем мире в достаточной степени ограничены, и поэтому приведенный алгоритм отвечает свойству неполноты знаний агента.

Отметим интересную особенность метода: тот факт, что адаптация распределения феромона к изменению окружающей среды происходит с некоторой задержкой соответствует обстановке в реальном мире. Прежде чем предприниматель адаптирует свое производство к последним изменениям рынка должно пройти какое-то время.

Программная реализация

Муравьиный алгоритм принятия экономических решений встраивался в уже существующий проект вычислимой экономической модели, описанный в работе [1]. Предложенный метод является новой стратегией поведения агента на рынке. В данном разделе речь пойдет об интерфейсах взаимодействия существующей модели и реализации муравьиного алгоритма.

Следующий интерфейс является основным при моделировании экономических агентов:

```

public interface IEconomicAgent {
    public void processPreProduceStage();
    public void processProduceStage();
    public void processPostProduceStage();
}

```

Методы `processPreProduceStage`, `processProduceStage`, `processPostProduceStage` вызываются, когда агенту необходимо произвести стадию закупки, производства и продажи соответственно.

Реализация интерфейса `IEconomicAgent`, `AutomataAgent` использует автоматную модель производства:

```

public class AutomataAgent implements IEconomicAgent {
    private INode state;
    private IWalkStrategy walkStrategy;
    @Override
    public void processPreProduceStage() { ... }
    @Override
    public void processProduceStage() { ... }
    @Override
    public void processPostProduceStage() { ... }
}

```

Поле `state` — состояние конечного автомата модели производства. Поле `walkStrategy` — стратегия выбора пути в автомате.

Стратегия поведения экономического агента абстрагируется при помощи интерфейса:

```

public interface IWalkStrategy {
    public IOutcome doWalk(INode state);
}

```

Единственный метод `doWalk` возвращает объект типа `IOutcome` – переход конечного автомата из состояния `state`, по которому следует продвигаться.

В процессе исследования была разработана новая реализация интерфейса `IWalkStrategy`, `FeromonStorage`:

```

public class FeromonStorage implements IWalkStrategy {
    private Map<IOutcome, FeromonNote> outcome2note;
    @Override
    public IOutcome doWalk(INode state) { ... }
}

```

Поле `outcome2note` — отображение из множества переходов автомата во множество объектов типа `FeromonNote`:

```

public class FeromonNote {
    public double[] feromonHistory;
    public void storeFeromon(double feromonValue) { ... }
    public double computeFeromon() { ... }
}

```

Как видно из кода, FeromonNote хранит оставленный муравьями феромон в поле feromonHistory. Метод storeFeromon позволяет оставлять муравью порцию феромона, а метод computeFeromon вычисляет сколько феромона находится на данном переходе с учетом того, что феромон имеет свойство со временем испаряться.

Описанная система встраивает реализацию муравьиного алгоритма в механизм принятия решений автоматной модели экономики.

Результаты

В процессе исследования была разработана и экспериментально опробована новая программная реализация стратегии поведения агентов. Все эксперименты, описанные в работе [1] были проведены с использованием этой стратегии. Результаты экспериментов, как и ранее, согласуются с экономической теорией.

Целью исследования была разработка новой модели поведения экономических агентов, которая бы в большей степени соответствовала реальности. В ходе исследования такая модель была разработана и испытана. Было показано, что новый алгоритм поведения лучше отвечает предъявляемым требованиям.

Выводы

Для моделирования процесса производства был выбран такой математический объект, как конечный автомат. Этот выбор обладает тем достоинством, что конечный автомат достаточно прост для анализа. Существует развитый математический аппарат для работы с конечными автоматами. В частности, при разработке системы принятия экономических решений, возможно использование алгоритмов, применяемых для анализа графов (конечный автомат – частный случай графа).

В данной работе для анализа автомата, моделирующего процесс производства, применялся муравьиный алгоритм. Как было показано выше, этот алгоритм хорошо соответствует таким принципам построения мультиагентных моделей, как субоптимальность поведения агентов, ограниченность знаний об окружающей среде и т. д.

Список литературы

1. *Гниломедов И. И.* Моделирование экономических агентов при помощи конечных автоматов /Научные доклады «Интегрированные модели, мягкие вычисления, вероятностные системы и комплексы программ в искусственном интеллекте». М.: Физматлит. 2009, с. 72–89.
2. *Dorigo M., Stutzle T.* Ant Colony Optimization. MIT Press, 2004.
3. *Rudomin I., Millan E., Hernandez B.* Fragment shaders for agent animation using finite state machines // Simulation Modelling Practice and Theory. V. 13, I.8, November 2005, pp. 741–751.
4. *Naumov L., Shalyto A.* Automata Theory for Multi-Agent Systems Implementation Integration of Knowledge Intensive Multi-Agents Systems /Proceedings of International Conference Integration of Knowledge Intensive Multi-Agent Systems: Modeling, Exploration and Engineering. KIMAS-03. IEEE Boston Section. 2003, pp. 65–70.
5. *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы. Построение и анализ. М.: Вильямс, 2005.