



Облако тэгов за май-июнь, 2009

## Факты и планы

Переворачивая с ног на голову “план/факт” анализ, мы хотим Вас ознакомить с основными событиями за май-июнь и планами на ближайшие месяцы.

### Факты и “слова-аттракторы” :-)

Закончились семинары по агентному моделированию	муравьи, генетические алгоритмы, Star Logo
Доклады на конференции Oracle ISV Forum	ненормальные методы, междисциплинарный подход
Новый шаблон сайта	приветствуем Internet Explorer
TEDx Russia	локализация TED ( <a href="http://www.tedx.ru">www.tedx.ru</a> )

### Планы

Семинары в сфере управления проектами разработки ПО	Peoplemind, ненормальные методы, взаимоотношения с клиентами
BITed	TED в ГУ ВШЭ, генетические алгоритмы

### В выпуске:

Генетические алгоритмы

3

Показатели эффективности

5

# Генетические алгоритмы



Рис. 1: Генетические алгоритмы

Что же такое генетические алгоритмы? Напомню, что алгоритм – это последовательность шагов для преобразования входного параметра в выходной. К примеру, если нам нужно написать программу для робота, который собирает мусор с территории вокруг дома, то мы должны описать как робот будет принимать решение о том или ином действии (выходной параметр), к примеру поднять мусор или двигаться дальше, на основе информации о состоянии внешней среды (входной параметр), к примеру есть ли на данном участке территории мусор. То есть мы должны сесть и придумать для робота оптимальную стратегию уборки территории!

Но люди ленивы по своей природе и решили, что можно заставить компьютер найти оптимальную стратегию для робота. Вы просто пишете программу, которая ищет оптимальный алгоритм. То есть вы поднимаетесь на уровень абстракции выше: программируете не оптимальный алгоритм, а алгоритм нахождения оптимального алгоритма! Вот оно как!

Как же написать такой алгоритм? Для этого вы используете принципы биологической эволюции, а именно:

- отбор – выживает та популяция особей (в нашем случае, алгоритмов), которая лучше других приспособлена к окружающей среде;
- размножение – выжившие популяции размножаются и передают свои отличительные особенности потомкам;
- мутация – в редких случаях происходит мутация в потомстве и потомство приобретает характеристики, отличные от характеристик родителей.

## Описание генетического алгоритма

Допустим, что у нас есть робот, который должен убирать территорию, представленную в виде квадрата из 100 клеток (10×10). По всей территории случайным образом разбросан мусор. Давайте опишем генетический алгоритм поиска оптимальной стратегии уборки территории для нашего робота.

	1	2	3	4	5	6	7	8	9	10
1	●									
2										
3	-							-		
4			-							
5					-					
6										
7		-					-			
8										
9			-						-	
10				-	-	-				

Легенда:  
- мусор  
● робот

Рис. 2: Робот-сборщик мусора

Что в нашем случае есть стратегия (алгоритм) уборки? Стратегия – это набор правил, согласно которым робот должен принимать решения в ответ на состояние окружающей среды.

Как робот узнает о состоянии окружающей среды? Наш робот может видеть всего на одну клетку вверх, влево, вправо, и вниз, и разумеется на клетку, в которой он находится сейчас. Так робот сканирует окружающую среду. Какие могут быть состояния окружающей среды? В клетке может быть мусор, она может быть пуста или эта клетка может быть стеной (например, если робот находится в клетке 1,1 то на севере и западе у него стена).

Какие действия может принять робот в ответ на состояние окружающей среды? У него есть 7 возможных действий: сделать шаг вверх, вниз, налево, направо, пойти “куда глаза глядят” (random), а также поднять мусор или не поднимать ничего. Таким образом получается, что стратегия – это набор правил о том, какое из 7 действий совершить в зависимости от того или иного состояния внешней среды. Схематически представим одно из правил в следующем виде:

Сверху	Снизу	Слева	Справа	Текущая клетка	Действие
Стена	Пусто	Стена	Пусто	Пусто	Шаг вправо

**Рис. 3: Правила поведения**

Сколько может быть таких правил в стратегии? Очевидно, что нужно перебрать все возможные комбинации окружающей среды: 243 правила (3 в пятой степени). И каждой возможной комбинации нужно однозначно сопоставить одно из 7 возможных действий.

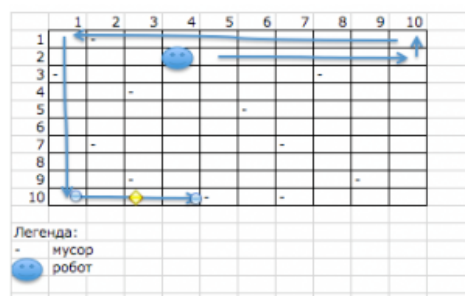
Вот последовательность шагов для нахождения оптимального алгоритма:

1. Создать изначальную популяцию возможных алгоритмов (стратегий) уборки территории – возьмем 200 алгоритмов. Самый простой способ: выбрать случайным образом (random) действия в ответ на каждое состояние среды.
2. Прогнать каждый алгоритм для 100 возможных конфигураций расположения мусора на территории.
3. Подсчитать приспособленность (fitness) каждого алгоритма. В нашем случае алгоритм является тем более приспособленным, чем быстрее робот сможет убрать территорию, используя данный алгоритм. Правила такие: если робот поднял мусор то +5 очков, если ударился в стенку, то -1 очко. Таким образом, приспособленность стратегии равна среднему количеству очков, заработанных роботом при использовании данного алгоритма за 100 сессий уборки.
4. Выбрать определенное количество наиболее приспособленных алгоритмов, которые станут родителями нового поколения алгоритмов.
5. Скрестить родителей. Каждая пара алгоритмов создает потомка, путем скрещивания. В нашем случае, один из 2 потомков будет наследовать часть правил от отца, а вторую часть – от матери. Второй потомок – наоборот: вторую часть от отца, и первую – от матери.
6. Мутация – с малой вероятностью нужно мутировать некоторые правила из стратегии потомка. Например, заменить действие “Шаг вправо” на “Шаг влево” для одного из правил.
7. 1000 раз повторить процесс с шага 2.

Весь фокус в том, что в процессе такой эволюции за 1000 поколений получается алгоритм, который почти оптимален! И от вас не требуется почти никакой работы! Вот вся сила и простота генетических алгоритмов. Я считаю, что будущее программирования именно за ними, программист еще выше поднимается по иерархии абстракций и начинает программировать алгоритмы нахождения оптимальных алгоритмов. Следует заметить, что генетические алгоритмы, несмотря на их относительную новизну, уже активно используются для решения реальных задач: от General Electric (автоматизация проектирования комплектующих к самолетам) до John Deere (планирование работы сборочных конвейеров).

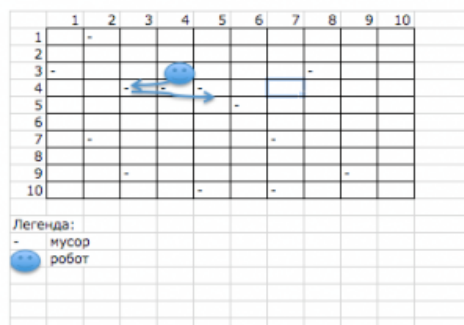
## Оптимальная стратегия уборки

Оптимальная стратегия уборки во многом напоминает оптимальную игру в змейку на телефонах Nokia: в случае отсутствия мусора в ближайших клетках робот двигается кругообразно (до упора вправо, потом до упора наверх, и против часовой стрелки) пока не найдет мусор.



**Рис. 4: Оптимальная стратегия**

Но замечено еще необычное поведение: в случае если робот находит 3 последовательно лежащие кучки мусора, то он сначала находит самую левую из кучек мусора, и потом двигаясь направо собирает все три. Мы бы навряд ли сами догадались до такого поведения: с высокой вероятностью мы бы запрограммировали поведение, что если в текущей клетке есть мусор, то следует его поднять. Более того, даже в начале работы генетического алгоритма “выживали” бы стратегии, которые именно так и поступали. Но что нам помогло? Именно! Мутация! Мутация помогла сгенерировать правило, которое ЛОКАЛЬНО неоптимально, но ГЛОБАЛЬНО ведет к оптимальному результату. Вот вам и теория ограничений :)))



**Рис. 5: Необычное поведение**

Вот, собственно и все, что я хотел рассказать о генетических алгоритмах. На семинаре мы еще поговорили о том, для решения каких задач можно применить генетические алгоритмы и придумали пару идей: от task management-а до управления светофорами. Есть еще идеи?

Для более подробного изучения генетических алгоритмов рекомендую книгу Джона Холланда “Адаптация в естественных и искусственных системах”, а сам я почерпнул этот материал из книги Melanie Mitchell “Complexity: a guided tour”.

Кстати, кто-нибудь нашел ошибку в расчетах количества возможных ситуаций (243)? :) Пишите в комментариях (кроме тех, кто был на семинаре).

Оригинал поста: <http://www.empatika.com/blog/agent-modelling-seminars-3>

# Показатели эффективности



Когда я писал магистерскую диссертацию о системе показателей оценки деятельности компании, я столкнулся с книгой Маршала Мейера "Оценка эффективности бизнеса. Что будет после Balanced Scorecard?". В первой части книги автор рассуждает о том, почему оценка эффективности столь неудовлетворительна и поясняет, что одна из причин заключается в том, что некоторые компании в анализе деятельности опираются лишь на финансовые показатели, которые, по сути, являются запаздывающими (лаговыми), так как фиксируют то, что уже случилось. Поэтому в оценке эффективности необходимо выделять и опережающие (определяющие) показатели.

В этом посте я продемонстрирую проблему запаздывающих показателей и важность опережающих на простейшей модели СД. Допустим, у нас есть некоторый производственный процесс, в результате выполнения которого появляется продукт. Этот производственный процесс состоит из 5 стадий, причем эффективность каждой стадии оценивается показателем  $P(n)$ . Например, эффективность первой стадии оценивается  $P(1)$ , эффективность второго –  $P(2)$  и так до  $P(5)$ . Весь

производственный процесс оценивается показателем Себестоимость продукта, который является арифметической суммой показателей каждой стадии ( $Сб = P(1) + P(2) + P(3) + P(4) + P(5)$ ).

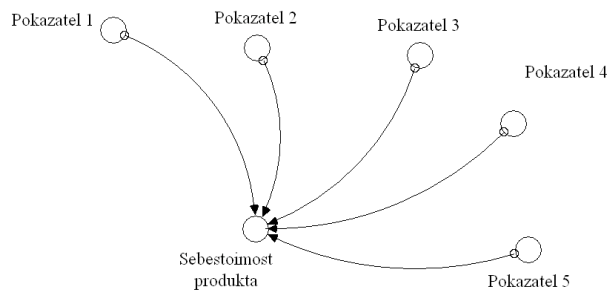


Рис. 6: Расчет себестоимости продукта

Каждый показатель стадии производственного процесса подвержен колебаниям, так как процесс выполняют живые люди, которые не могут всегда работать с заданной производительностью. Поэтому каждый показатель описывается формулой  $NORMAL(1;0.05)$  – то есть значение показателя стадии в момент времени будет  $1 \pm 0,05$ .

Также в нашу модель введем возможность увеличения показателя стадии 1 на заданное значение в заданный момент времени. То есть в определенный момент времени (мы пока не знаем какой) показатель стадии 1 будет колебаться не в пределах  $1 \pm 0,05$ , а в пределах  $(1 \pm 0,05) + (\text{величина изменения показателя})$ .

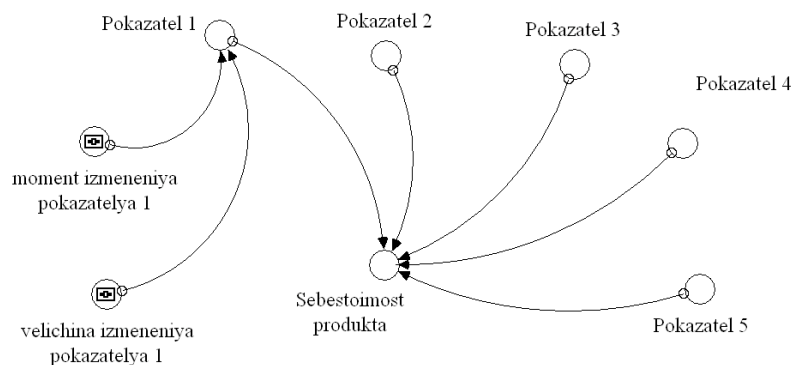
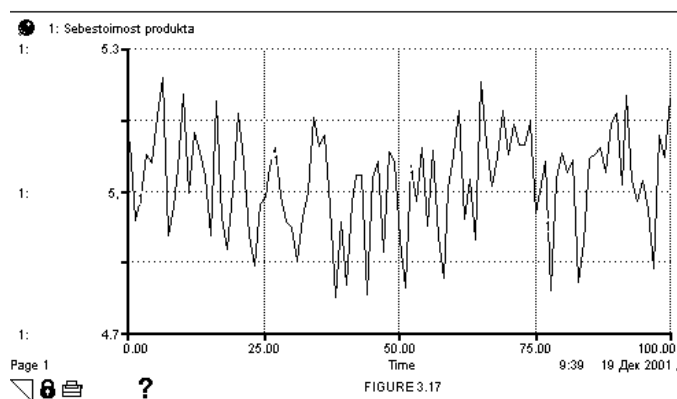


Рис. 7: Изменение показателя стадии 1

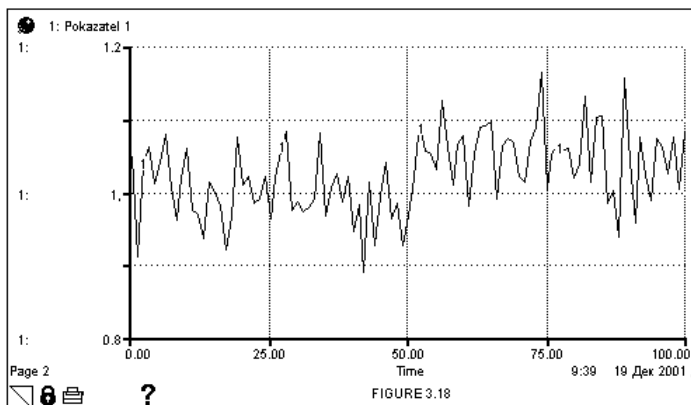
Ниже представлен график изменения себестоимости продукта – можете ли вы определить, в какой именно момент произошло изменение показателя стадии 1?



**Рис. 8: График изменения себестоимости продукта**

Бьюсь об заклад, что вы затрудняетесь это сделать. Почему? Потому что колебания других показателей “мешают” нам разглядеть, когда именно произошло изменения показателя 1. Теперь представьте реальную ситуацию: вы видите, что у вас себестоимость продукта постоянно скачет, но вы не можете понять являются ли колебания, присущими данному производственному процессу (люди не роботы), или же произошло заметное изменение на одном из этапов производственного процесса.... Какой же из вас менеджер тогда?! :)

Но, если мы взглянем на график изменения каждого показателя (ниже представлен график изменения показателя 1), то мы с легкостью определим, какой именно показатель и в какой момент времени заметно изменился.



**Рис. 8: График изменения показателя стадии 1**

Взглянув на этот график мы легко определяем заметное “структурное” изменение колебаний показателя 1, происшедшее где-то в момент времени = 50. Получается, что наблюдая за показателями процесса мы быстрее можем среагировать на изменение, нежели ориентируясь исключительно на выходные параметры процесса...

Причем, хочу заметить, что это актуально не только для производственных компаний (модель может натолкнуть на такие размышления). Возьмем, работу отдела продаж в сервисной компании: выходным процессом работы sales manager-ов является заключение контракта. Но если мы будем ориентироваться лишь на этот показатель, то не сможем среагировать на его изменения с точки зрения планирования ресурсов для выполнения этого контракта, что неблагоприятно повлияет на сроки выполнения заказа и удовлетворенность клиента. Работа получается реактивной, нежели проактивной. Отсюда и появляется инструмент “воронка продаж”, который предлагает систему показателей процесса продаж и помогает прогнозировать выходные параметры этого процесса.. Кстати, для одного из наших консалтинговых проектов мы создали системнодинамическую модель отдела продаж, в основу которой и заложили инструмент воронки продаж. Но об этом, в следующем посте :)

Можете привести еще примеры, когда ориентация исключительно на выходные параметры приводилась к неправильным или несвоевременным решениям? А может кто-то приведет контрпримеры?

Оригинал поста и модель для скачивания: <http://www.empatika.com/blog/input-output-measures>