

***Эволюционный алгоритм синтеза
тестов для дискретных устройств***

Сперанский Д.В.

***Российский университет транспорта
(МИИТ)***

Постановка задачи

Пусть задано дискретное устройство (ДУ) и множество его допустимых неисправностей. Предполагается, что эти неисправности являются одиночными константными. Требуется разработать алгоритм для синтеза тестов минимальной длины, позволяющих обнаруживать все неисправности из заданного множества. Далее эту задачу будем называть задачей синтеза тестов (ЗСТ).

Для простоты изложения в качестве ДУ в докладе рассматриваются комбинационные устройства.

Ранее известные подходы к решению задачи

- случайное и псевдослучайное тестирование;
- метод критических путей;
- метод различающей функции;
- метод альтернативных графов;
- метод активизации одномерного пути;
- метод сигнатурного анализа и т.д.

Недостаток этих методов – их эффективность не всегда отвечает современным требованиям, особенно для устройств большой размерности.

Современные тенденции в решении оптимизационных задач

В последние 30-40 лет активно развиваются различные методы и технологии, инспирированные природными системами (natural computing), нашедшие широкое практическое применение и показавшие высокую эффективность.

К их числу относятся :

- генетические алгоритмы;
- роевые алгоритмы;
- муравьиные алгоритмы и т.п.

Муравьиные алгоритмы

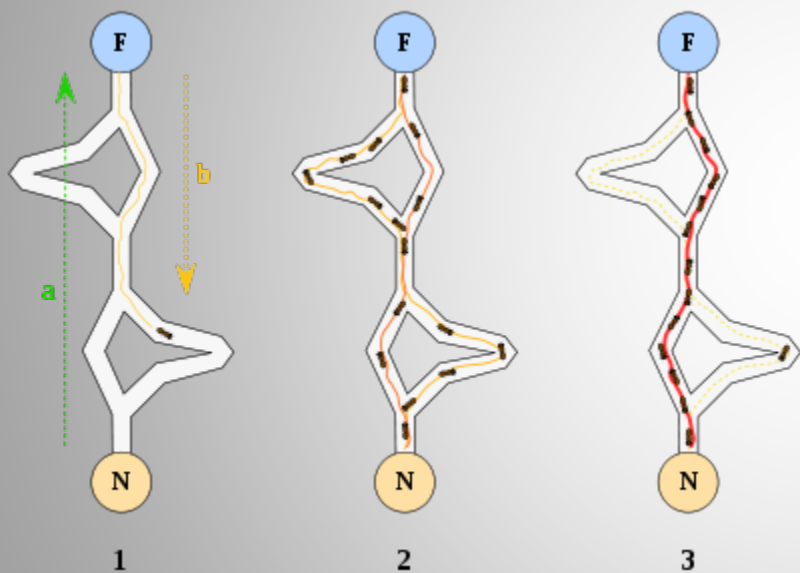




09.07.2018

Марко Дориго

Идея муравьиного алгоритма



1. Первый муравей находит источник пищи (F) любым способом (a), а затем возвращается к гнезду (N), оставив за собой тропу из феромонов (b).

2. Затем муравьи выбирают один из четырёх возможных путей, многократные проходы по которому «укрепляют» его и делают привлекательным.

3. Муравьи выбирают кратчайший маршрут, поскольку у более длинных феромоны испаряются сильнее.

Компоненты обеспечения самоорганизации поиска пищи муравьями

Самоорганизация является результатом взаимодействия следующих четырех компонентов:

- случайности;
- многократности;
- положительной обратной связи;
- отрицательной обратной связи.

Заметим, что распределение феромона по пространству передвижения муравьев является своего рода динамически изменяемой глобальной памятью муравейника. Любой муравей в фиксированный момент времени может воспринимать и изменять лишь одну локальную ячейку этой глобальной памяти.

Моделирование поведения муравьев базируется на использовании принципа обратной связи

- **Положительная обратная связь** – чем больше концентрация феромона на возможном маршруте, тем привлекательнее для муравья это направление движения. Эта обратная связь приведет к тому, что кратчайший маршрут станет единственным маршрутом движения большинства муравьев колонии.
- **Отрицательная обратная связь** – испарение феромона с течением времени с маршрута, либо редко, либо совсем не используемого муравьями. Эта обратная связь гарантирует, что найденный локально оптимальный маршрут не будет единственным, муравьи будут искать и другие пути.

Задача синтеза тестов для КУ (ЗСТ) близка к классической задаче коммивояжера (ЗК)

Сходство

- 1.Используемые математические модели – графы (в ЗК вершинами являются города, в ЗСТ – входные наборы КУ).
2. Каждая вершина посещается ровно один раз.
3. Все ребра графа имеют определенную «длину» (в ЗК – расстояние между городами, в ЗСТ – длина имеет иную интерпретацию).
- 4.Цель – найти маршрут в графе некоторой «оптимальной» длины.

Различия

- 1.В ЗК – граф не является полным, в ЗСТ- граф полный.
- 2.Содержательный смысл длины ребра в ЗК и ЗСТ различен.
- 3.Решение в ЗК - минимальный гамильтонов цикл, в ЗСТ – маршрут минимальной длины, удовлетворяющий иным заданным требованиям.
- 4.Длина ребер графа в ЗК задается заранее, в ЗСТ- вычисляется с использованием методов логического моделирования рассматриваемого КУ.

Общая схема одного из основных этапов алгоритма решения ЗСТ

Процесс построения теста каждым k -ым муравьем является пошаговым. Он начинается с включения в тест некоторого входного набора с последующим добавлением к «текущему» тесту $T_{тек}$ нового набора на каждом очередном шаге. С помощью $T_{тек}$ обнаруживается некоторое множество неисправностей ЦУ, которое исключается из первоначально заданного множества неисправностей F и в результате остается множество F_k^H еще не обнаруженных неисправностей. Добавление к «текущему» тесту нового набора продолжается до тех пор, пока множество F_k^H не окажется пустым.

Для каждого муравья переход в тесте от вход. набора i к набору j зависит от трех составляющих:

- От табу-списка –в нем наборы, запрещенные к использованию, т.к. они уже были использованы в построенном ранее фрагменте теста .
- От значения видимости.
- От значения виртуального следа феромона.

Общая схема алгоритма решения ЗСТ

Алгоритм построения теста начинается с размещения в вершинах графа КУ случайным образом группы из m муравьев, где $m \leq 2^p$ (это варьируемый параметр).

Процесс построения теста состоит в повторении, как правило, заранее заданного числа итераций алгоритма (возможны иные критерии останова).

Каждый муравей строит свой тестовый маршрут в графе. После построения теста до конца каждым муравьем завершается текущая итерация и производится подготовка к выполнению следующей. Она заключается в изменении концентрации феромона на ребрах графа. Содержательно это отражает приобретение муравьями опыта в поиске теста. При корректировке концентрации используется не только положительная, но и отрицательная обратная связь – испарение феромона. Этот принцип приводит к тому, что группа муравьев, решающая ЗСТ, одновременно исследует разные точки пространства решений.

После завершения очередной итерации выводится наилучший маршрут T^+ и его длина L^+ .

Используемые обозначения

$F = \{f_1, f_2, \dots, f_n\}$ - заданное множество неисправностей ЦУ ($|F| = n$),

\mathcal{W} - некоторое множество входных наборов ЦУ,

$F(\mathcal{W})$ - множество неисправностей из F , обнаруживаемых с использованием последовательности входных наборов \mathcal{W} ,

$F_{k,t}^N(T_{тек})$ - множество еще не обнаруженных тестом $T_{тек}$ k -ым муравьем неисправностей из множества F на итерации t ,

$D_{ij}(k,t) = |F_{k,t}^N(T_{тек}, i) \cap F_{k,t}^N(T_{тек}, j)|$ - число неисправностей, не обнаруживаемых k -ым

муравьем на итерации t входным набором i , так и следующим за ним в текущем тесте входным набором j . Интуитивно понятно, что чем меньше $D_{ij}(k,t)$, тем больший вклад внесет набор j в обнаружение неисправностей заданного множества F в синтезируемом k -ым муравьем тесте,

τ_{ij} - концентрация (количество) феромона на ребре (i,j) графа рассматриваемого КУ.

Используемые обозначения

$$d_{ij} = \begin{cases} |F(i) \cap F(j)|, & \text{если } F(i) \neq F(j), \\ C - \text{большая константа}, & \text{если } F(i) = F(j). \end{cases}$$

Это величина является длиной ребра (i, j) графа рассматриваемого КУ. Длины всех ребер вычисляются перед началом работы МА методами логического моделирования КУ. Понятно, что $d_{ij} = d_{ji}$, т.е. матрица длин графа КУ симметрична относительно главной диагонали.

Содержательный смысл d_{ij} : чем больше общих неисправностей, обнаруживаемых входными наборами i и j в отдельности, тем менее привлекательно ребро (i, j) для включения его в тестовый маршрут. Это вытекает из следующих соображений: чем больше имеется различных неисправностей из F, обнаруживаемых наборами i и j, тем больший вклад в построение теста может внести эта пара наборов.

Понятно, что вклад каждого набора из пары (i, j) в тест при распознавании неисправностей из F в этом случае будет уменьшаться с ростом величины d_{ij} . Это хорошо согласуется с используемым в МА для ЗК понятием видимости вершин графа: чем больше значение d_{ij} , тем «дальше» вершина j от вершины i (и вершина j от i), т.е. тем хуже ее «видимость».

$\eta_{ij} = 1/d_{ij}$ - - видимость, являющаяся локальной статической информацией, которая выражает естественное желание включить в качестве очередного элемента искомого теста входной набор с наибольшим вкладом в обнаружение неисправностей множества F.

Правила корректировки концентрации феромона на ребрах графа

Так же как и в ЗК, при решении ЗСТ может быть использовано аналогичное правило вычисления вероятности перехода k -го муравья от одной вершины графа к другой, формула вычисления количества откладываемого на ребре графа феромона, правило корректировки концентрации феромона за счет испарения и т.п.

Соответствующие формулы для корректировки приведены на следующих слайдах.

Заметим, что все это хорошо согласуется с очевидной аналогией перечисленных правил и формул в МА для решения ЗСТ с таковыми в МА для решения ЗК.

Формулы для вычислений при реализации муравьиных алгоритмов

Вероятность перехода k -го муравья из вершины i в вершину j на t -ой итерации имеет следующий вид:

$$\left\{ \begin{array}{l} P_{ij,k}(t) = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{i \in J_{i,k}} [\tau_{ii}(t)]^\alpha [\eta_{ii}(t)]^\beta}, \quad \text{если } j \in J_{i,k} \\ P_{ij,k}(t) = 0, \quad \text{если } j \notin J_{i,k} \end{array} \right. \quad (1)$$

где α и β - регулируемые параметры, задающие веса следа феромона и видимости при построении теста.

Понятно, что при $\alpha = 0$ концентрация феромона при выборе ребра не используется, т.е. выбор осуществляется только на основе видимости вершины j .

Если же $\beta = 0$, то для выбора используется только концентрация феромона, что может привести к сваливанию маршрутов к одному из субоптимальных.

Выбор очередного набора теста j должен осуществляться по правилу рулетки. При этом каждый город имеет в секторе рулетки площадь, пропорциональную вероятности (1).

•

Формулы для вычисления при реализации муравьиных алгоритмов

Количество феромона, откладываемого на ребре (i,j) k-м муравьем, вычисляется по формуле

- $$\Delta\tau_{ij,k}(t) = \begin{cases} Q / L_k(t), & \text{если } (i, j) \in T_k(t), \\ 0, & \text{если } (i, j) \notin T_k(t), \end{cases} \quad (2)$$

- где $T_k(t)$ - маршрут, пройденный k-м муравьем на итерации t, $L_k(t)$ - длина этого маршрута, т.е. число ребер, входящих в него, Q - регулируемый параметр.

- Пусть $\rho \in [0,1]$ - это коэффициент испарения феромона. Формула корректировки концентрации феромона на ребре (i, j) графа за счет его испарения имеет следующий вид:

- $$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t), \quad (3)$$

где $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij,k}(t)$, m - количество муравьев в группе (колонии), строящих маршрут коммивояжера (m является регулируемым параметром).

Выбор численности группы муравьев, участвующих в решении ЗСТ

В ЗК обычно число муравьев полагают равным числу городов и каждый муравей строит гамильтонов маршрут из своего города. Общее количество муравьев в колонии остается постоянным на протяжении выполнения всего алгоритма.

В ЗСТ требование равенства числа муравьев m , синтезирующих тест для ЦУ, числу всех его входных наборов не обязательно (m может быть существенно меньше и является регулируемым параметром).

Улучшение временных характеристик муравьиного алгоритма решения ЗСТ

По аналогии с ЗК для улучшения временных характеристик МА для ЗСТ можно использовать некоторое количество так называемых *элитных муравьев*.

Цель их использования состоит в усилении ребер наилучшего тестового маршрута, построенного с начала работы алгоритма. Как и в ЗК, количество феромона, откладываемого на ребрах наилучшего текущего тестового маршрута T^+ , можно принять равным Q / L^+ , где L^+ - длина маршрута T^+ . Если имеется S элитных муравьев, то ребра маршрута T^+ получают общее «усиление» $\Delta\tau_0 = S \cdot Q / L^+$, где Q – есть регулируемый параметр.

Этот феромон побуждает муравьев исследовать решения, содержащие несколько ребер наилучшего на данный момент маршрута.

Подведем некоторые итоги

1. МА для ЗСТ сведен к модификации МА для ЗК, для которой разработано несколько очень эффективных разновидностей муравьиных алгоритмов.
2. *Недостатки:*
 - а) МА дает приближенное решение задачи и время его сходимости не определено.
 - б) Эффективность сильно зависит от регулируемых параметров, т.е. требует проведения предварительных экспериментов.
3. *Достоинства:*
 - а) Статистика показывает, что МА находят хорошие решения значительно быстрее, чем точные методы оптимизации.
 - б) Эффективность МА возрастает с ростом размерности задачи.

Спасибо за внимание!