

АНАЛИЗ АЛГОРИТМОВ АВТОМАТИЧЕСКОЙ РАССТАНОВКИ СУЩНОСТЕЙ И ПОСТРОЕНИЯ СВЯЗЕЙ

Е.А. Киряков

3iceh9i@gmail.com

SPIN-код: 9374-2244

А.Т. Левинский

adam.levinskiy@yandex.ru

SPIN-код: 2301-6960

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Большие системы часто удобно изучать с помощью концептуальных диаграмм (например, диаграмм сущность-связь или UML). Для автоматического построения таких диаграмм требуются алгоритмы автоматической расстановки сущностей и автоматического построения связей. В связи с этим был выполнен анализ существующих алгоритмов автоматической расстановки сущностей и автоматического построения связей. Приведены три основные группы алгоритмов: алгоритмы на основе физической модели, аналитические алгоритмы и генетические алгоритмы. По каждой группе алгоритмов передана основная информация, перечислены плюсы и минусы каждой группы алгоритмов. Помимо этого для каждой группы алгоритмов разобрано несколько конкретных алгоритмов. Изучение материалов данной статьи даст читателям базовые знания об основных группах алгоритмов и поможет понять принцип их работы.

Ключевые слова

Алгоритм, автоматизация, связи, построения, генетический алгоритм, модель, генетический, аналитический, мутация, расстановка сущностей, построения связей

Поступила в редакцию 04.04.2019

© МГТУ им. Н.Э. Баумана, 2019

Введение. UML (Unified Modeling Language) — это унифицированный язык моделирования, предназначенный для объектно-ориентированного анализа и проектирования. Его можно использовать для визуализации, спецификации, конструирования и документирования программных систем. Давно признано, что полезность концептуальных диаграмм (например, диаграммы сущность — связь, или UML) Разобраться в большой схеме без помощи лиц, уже знакомых с ней, может оказаться сложным и почти невозможным. К сожалению, крупные концептуальные схемы становятся все более частым явлением. Интеграция информационных систем, разработка или реинжиниринг крупных систем, использование систем ERP (Enterprise Resource Planning — планирование ресурсов предприятия) естественным образом приводят к появлению очень больших схем (база данных SAP включает 30 000 таблиц). Хорошая визуализация концептуальной схемы может помочь в ее понимании. Для визуализации схем предложены алгоритмы автоматической расстановки. Все предложенные алгоритмы подразделяют на три основных типа, каждый из которых будет рассмотрен в данной статье.

Алгоритмы на основе физической модели. Алгоритмы данной группы ставят в соответствие графу некоторую физическую модель и вводят функцию, которую впоследствии минимизируют (например, функцию потенциальной энергии системы). Популярным среди физических алгоритмов является силовой алгоритм.

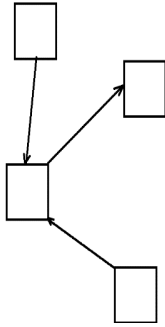


Рис. 1. Пример результата работы силового алгоритма

В силовом алгоритме сущности представляются как заряженные частицы, а связи — как пружины. Сущности отталкиваются друг от друга, подчиняясь закону Кулона, а связи стремятся к нулевому относительному удлинению, подчиняясь закону Гука.

Воздействия указанных выше сил достаточно, чтобы расположить граф на плоскости. Благодаря воздействию сил отталкивания сущности не будут сливаться в одну точку, а связи не будут слишком короткими и слишком длинными благодаря закону Гука. Но у результирующей диаграммы есть один большой недостаток — сущности и связи будут беспорядочно раскиданы по диаграмме, не будет никакой симметричности (рис. 1).

Данный алгоритм можно улучшить, добавив в систему магнитное поле, направленное в какую-либо сторону. Связи будут взаимодействовать с магнитным полем, выстраиваясь по его направлению (рис. 2).

Алгоритмы на основе физической модели обладают следующими основными преимуществами:

- легко реализуемы;
- легко поддаются модификации;
- хорошо изучены;
- могут расставить любой граф;
- их можно использовать для расстановки в режиме реального времени;

К их недостаткам можно отнести следующее:

- непостоянный результат (для одинаковых входных данных каждый раз будет получаться новый результат);
- не учитываются размеры вершин;
- связи — только прямые линии.

Аналитические алгоритмы. Аналитические алгоритмы представляют собой последовательность преобразований графа, приводящих к укладке графа. Главное преимущество данной группы алгоритмов состоит в том, что с их помощью получаем гарантированный результат, в отличие от алгоритмов на базе физической модели (т. е. для одинаковых входных данных всегда будем получать одинаковый итоговый граф).

Один из алгоритмов, которые можно отнести к данной группе, — гамма-алгоритм. Гамма-алгоритм основан на выделении сегментов в графе и их определенной укладке в нужном образом выбранных гранях графа.

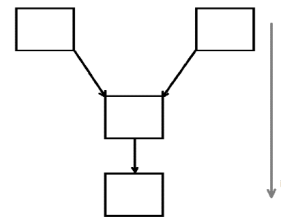


Рис. 2. Пример результата работы силового алгоритма при использовании магнитного поля

Работы гамма-алгоритма включает следующие этапы.

1. Инициализация. Выбирают любой простой цикл C исходного графа G ; изображают его на плоскости в виде грани, которую принимают за уже уложенную часть G^* ; формируют сегменты S_i ; если множество сегментов пусто, переходят к п. 3.

2. Общий шаг. Пока множество сегментов не пусто:

а) для каждого сегмента S находят множество S^* . Если существует сегмент S , для которого $|(S^*)| = 0$, граф не планарный, конец работы алгоритма;

б) выбирают один из сегментов с минимальным числом вмещающих его граней;

в) выбирают одну из подходящих граней для выбранного сегмента;

г) в данном сегменте выбирают цепь между двумя контактными вершинами и укладывают ее в выбранной грани. Учитывают изменения в структуре сегментов и переходят к подпункту а).

3. Завершение. Построена плоская укладка G^* исходного графа G , конец работы алгоритма.

Помимо гамма-алгоритма одним из популярных алгоритмов автоматической расстановки сущностей и построения связей является иерархический алгоритм. Данный алгоритм позволяет расставить сущности по слоям, а затем задать связи между сущностями. Основные этапы работы алгоритма показаны на рис. 3.

Достоинствами иерархических алгоритмов являются:

– одинаковый результат при одинаковых входных данных;

– очень хороший результат при правильной реализации;

– возможность осуществлять укладку графа, используя для соединения вершин криволинейные ребра.

Недостатками являются:

– тяжелая реализация;

– трудности модификации.

Генетические алгоритмы. Генетические алгоритмы, так же, как и физические, являются универсальными и способны расставить любой граф, который будет подан им на вход.

Генетический алгоритм схож с физическим не только своей универсальностью, но и происхождением — оба алгоритма взяты из природы. Генетический алгоритм основан на теории эволюции. Подобно эволюции живых организмов, граф эволюционирует из случайно разбросанных по диаграмме вершин и ребер в конечный граф, который удобно просматривать человеку.

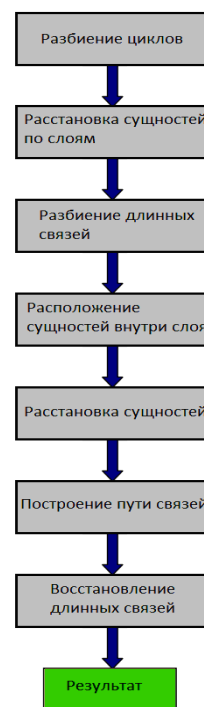


Рис. 3. Этапы работы иерархического алгоритма

1 0 1
 $|x_1| |x_2| \dots |x_n|$

Рис. 4. Битовая строка (ДНК)

Для генетического алгоритма стоит определить некоторые понятия:

– ДНК (рис. 4) — закодированное состояние графа. Обычно граф с помощью функции

кодируют в битовое представление. В закодированном состоянии графа должна содержаться вся информация, необходимая для отображения графа. Например, граф из двух вершин (максимальная координата вершины — 2) и ребра можно закодировать так: 0000 1111 10. 0000 — закодированная вершина первого графа (0, 0), 1111 — закодированная вершина второго графа (2, 2), 10 — ребро идет из вершины 0 в вершину 1, но не идет из вершины 1 в вершину 0;

- особь — один граф;
- популяция — набор особей.

В начале работы генетический алгоритм генерирует популяцию, особи которого будут скрещиваться и мутировать поколение за поколением. Для оценки поколения необходимо ввести функцию, которую в генетических алгоритмах называют фитнес-функцией или функция приспособленности. Меньшее значение функции означает лучшее поколение.

Фитнес-функцию необходимо реализовывать в зависимости от требований задачи. Фитнес-функция должна быть основана на критериях задачи и штрафовать за их несоблюдение. Например, единственный критерий задачи — нельзя, чтобы ребра пересекались. Тогда $F(x) = \text{CONST} \cdot N$, где CONST — константа, а N — количество пересечений.

Каждый параметр функции должен кодироваться битовой строкой, которая будет отображена в ДНК. Особью называется строка, являющаяся конкатенацией строк всего упорядоченного набора параметров.

В классическом генетическом алгоритме начальная популяция формируется случайным образом. Фиксируется размер популяции (количество особей N), который не изменяется в течение работы всего алгоритма. Каждая особь генерируется как случайная L -битная строка, где L — длина кодировки особи, она тоже фиксирована и одинакова для всех особей.

Как уже было сказано раньше, каждая особь — это отдельный граф (решение поставленной задачи). Чем больше приспособлена особь (меньше значение функции приспособленности), тем более подходящим является решение. Этим генетический алгоритм отличается от большинства других алгоритмов оптимизации, которые оперируют лишь с одним решением, улучшая его.

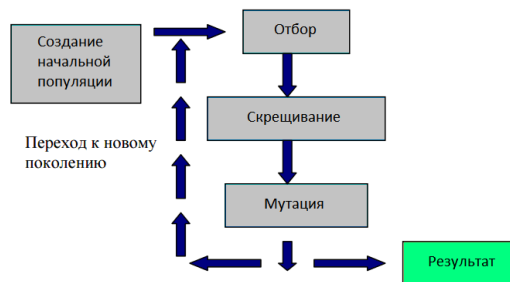


Рис. 5. Этапы генетического алгоритма

Генетический алгоритм предусматривает следующие этапы (рис. 5).

1. Генерация начальной популяции (в классическом алгоритме — случайная).

2. Генерация новых поколений (обычно генерация новых поколений происходит либо пока не будет достигнуто пороговое значение функции приспособленности, либо пока не будет достигнуто N -е поколение). В процессе генерации нового поколения происходят:

- а) отбор лучших особей среди пришедшей на вход промежуточной популяции;
- б) скрещивание особей для получения дочерних особей;
- в) мутация полученных дочерних особей;
- г) оценка полученных дочерних особей при помощи фитнес-функции.

3. Выбор лучшей особи из полученного поколения.

Промежуточная популяция — это набор особей, которые получили право скрещиваться. В промежуточную популяцию попадают наиболее приспособленные особи. Хорошо приспособленные особи могут «проскочить» через несколько поколений, а плохо приспособленные вылетят при первом отборе. Вероятность того, что особь попадет в промежуточную популяцию, пропорциональна ее приспособленности, т. е. работает пропорциональный отбор.

После отбора особи промежуточной популяции случайным образом разбиваются на пары. Каждая пара скрещивается, в результате скрещивания получаются две родительских и две дочерних особи.

Скрещиванием называют операцию кроссовера (обмена частями ДНК). В классическом генетическом алгоритме применяется односточный оператор кроссовера (1-point crossover): для родительских хромосом случайным образом выбирается точка раздела, и они обмениваются отсеченными частями. Полученные две строки являются потомками (рис. 6).

До: 110011,010010

После: 010011,110010

Рис. 6. Односточный кроссовер

До: 110101
После: 111101

К полученному в результате скрещивания новому поколению применяется оператор мутации (рис. 7). Каждый бит каждой особи популяции с вероятностью P_m инвертируется. Эта вероятность обычно очень мала и составляет менее 1 %.

Рис. 7. Мутация

Скрестив промежуточное поколение и применив мутацию, можно получить новое поколение. На этом промежуточный шаг генетического алгоритма завершается. Далее все действия повторяются. Если не вводить ограничений по возможному количеству поколений и/или по пороговому значению фитнес-функции, то генетический алгоритм будет продолжать работать вечно. Если ограничение введено, то после выполнения одного из условий на выход будет передано последнее полученное поколение, среди которого нужно выбрать наиболее приспособленную особь.

Выделим следующие преимущества генетических алгоритмов:

- возможность применения к любым входным графам;
- гибкость;
- гарантированно хороший результат.

Недостатки:

- тяжелая реализация функции кодирования особи;

- тяжелая реализация фитнес-функции;
- длительное время работы;
- непостоянный результат (для одинаковых входных данных каждый раз будет получаться разный итоговый граф).

Выводы. Проанализированы и подробно описаны три основных типа алгоритмов автоматической расстановки сущностей и построения связей: алгоритмы на основе физической модели, аналитические алгоритмы и генетические алгоритмы. Для каждого типа алгоритмов были рассмотрены примеры: силовой метод для алгоритмов на основе физической модели, гамма-алгоритм и иерархический алгоритм для аналитических алгоритмов и классический генетический алгоритм для генетических алгоритмов. Каждая группа алгоритмов имеет свои достоинства и свои недостатки, которые перечислены в конце описания каждой группы алгоритмов. Ниже приведена таблица, в которой сравниваются достоинства и недостатки каждого алгоритма. С помощью этой таблицы будет легче выбрать группу алгоритмов, подходящую для решения конкретной поставленной задачи.

Сравнение характеристик алгоритмов

Характеристика	Алгоритмы на основе физической модели	Аналитические алгоритмы	Генетические алгоритмы
Легкая реализация	+	–	–
Универсальность (подходят для любых входных графов)	+	–	+
Постоянный результат	–	+	–
Гарантированно высокое качество конечного графа	–	+	+
Возможность использовать в реальном времени	+	–	–

Например, необходимо реализовать алгоритм расстановки диаграммы, которая будет перестраиваться по мере того как пользователь взаимодействует с ней (перетаскивает сущности, изменяет связи). С учетом данных приведенных в таблице, для решения задачи подойдет алгоритм на физической основе, поскольку он позволяет строить диаграмму в режиме реального времени.

Литература

- [1] Соколов Г.В. Анализ алгоритмов автоматической укладки графов на плоскости в рамках задачи визуализации моделей на графах. *Вестник Пермского Государственного Технического Университета. Прикладная математика и механика*, 2008, № 15, с. 162–171.
- [2] Dickerson M., Eppstein D., Goodrich M.T., et al. Confluent drawings: visualizing non-planar diagrams in a planar way. *JGAA*, 2005, vol. 9, no. 1, pp. 31–52. DOI: 10.7155/jgaa.00099 URL: <http://jgaa.info/getPaper?id=99>
- [3] Гладков Л.А. Решение задачи планаризации графов на основе бионических технологий. *Вестник южного научного центра РАН*, 2005, т. 1, № 2, с. 51–57.

- [4] Davis L.D. Handbook of genetic algorithms. Van Nostrand Reinold, 1991.
- [5] Nikolov N.S., Tarassov A. Graph layering by promotion of nodes. *Discrete Appl. Math.*, 2006, vol. 154, no. 5, pp. 848–860. DOI: 10.1016/j.dam.2005.05.023
URL: <https://www.sciencedirect.com/science/article/pii/S0166218X05003100>
- [6] Siirtola H., Makinen E. The Barycenter heuristic and the reorderable matrix. *Informatica*, 2005, vol. 29, no. 3, p. 357–364.
- [7] Carstens J.J. Node and label placement in a layered layout algorithm. Master Thesis. University of Kiel, 2012.
- [8] Patarasuk P., Whalley D. Crossing reduction for layered hierarchical graph drawing. Master Thesis. Florida State University, 2004.
- [9] Genc B., Dogrusoz U. An algorithm for automated layout of process description maps drawn in SBGN. *Bioinformatics*, 2016, vol. 32, no. 1, pp. 77–84. DOI: 10.1093/bioinformatics/btv516
URL: <https://academic.oup.com/bioinformatics/article/32/1/77/1742665>
- [10] Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. М., Вильямс, 2000.

Киряков Евгений Александрович — студент кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Левинский Адам Тагирович — студент кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Иванов Алексей Михайлович, преподаватель кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

ANALYSIS OF ALGORITHMS FOR AUTOMATIC ARRANGEMENT OF ENTITIES AND CONSTRUCTION OF COMMUNICATIONS

E.A. Kiriakov

3iceh9i@gmail.com

SPIN-code: 9374-2244

A.T. Levinsky

adam.levinskiy@yandex.ru

SPIN-code: 2301-6960

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

It is often convenient to study large systems using conceptual diagrams (for example, entity-relationship diagrams or UML). For the automatic construction of such diagrams, algorithms for the automatic arrangement of entities and automatic construction of links are required. In this connection, an analysis was made of the existing algorithms for the automatic arrangement of entities and the automatic construction of links. There are three main groups of algorithms: algorithms based on the physical model, analytical algorithms and genetic algorithms. For each group of algorithms, basic information was transmitted; the pros and cons of each group of algorithms are listed. In addition, several specific algorithms are analyzed for each group of algorithms. The study of the materials of this article will give readers a basic knowledge of the main groups of algorithms and help to understand the principle of their work.

Keywords

Algorithm, automation, connections, constructions, genetic algorithm, model, genetic, analytical, mutation, arrangement of entities, construction of connections

Received 04.04.2019

© Bauman Moscow State Technical University, 2019

References

- [1] Sokolov G.V. Analysis of automatic graph drawing on a plane within the scope of model visualization problem at the graphs. *Vestnik Permskogo Gosudarstvennogo Tekhnicheskogo Universiteta. Prikladnaya matematika i mekhanika* [Perm State Technical University Bulletin. Applied Mathematics and Mechanics], 2008, no. 15, pp. 162–171 (in Russ.).
- [2] Dickerson M., Eppstein D., Goodrich M.T., et al. Confluent drawings: visualizing non-planar diagrams in a planar way. *JGAA*, 2005, vol.9, no. 1, pp. 31–52. DOI: 10.7155/jgaa.00099 URL: <http://jgaa.info/getPaper?id=99>
- [3] Gladkov L.A. Solution of the problem of graph planarization on basis bionic technologies. *Vestnik yuzhnogo nauchnogo tsentra RAN* [Vestnik SSC RAS], 2005, vol. 1, no. 2, pp. 51–57 (in Russ.).
- [4] Davis L.D. Handbook of genetic algorithms. Van Nostrand Reinold, 1991.
- [5] Nikolov N.S., Tarassov A. Graph layering by promotion of nodes. *Discrete Appl. Math.*, 2006, vol. 154, no. 5, pp. 848–860. DOI: 10.1016/j.dam.2005.05.023 URL: <https://www.sciencedirect.com/science/article/pii/S0166218X05003100>
- [6] Siirtola H., Makinen E. The Barycenter heuristic and the reorderable matrix. *Informatica*, 2005, vol. 29, no. 3, p. 357–364.

- [7] Carstens J.J. Node and label placement in a layered layout algorithm. Master Thesis. University of Kiel, 2012.
- [8] Patarasuk P., Whalley D. Crossing reduction for layered hierarchical graph drawing. Master Thesis. Florida State University, 2004.
- [9] Genc B., Dogrusoz U. An algorithm for automated layout of process description maps drawn in SBGN. *Bioinformatics*, 2016, vol. 32, no. 1, pp. 77–84. DOI: 10.1093/bioinformatics/btv516 URL: <https://academic.oup.com/bioinformatics/article/32/1/77/1742665>
- [10] Aho A.V., Ullman J.D., Hopcroft J.E. Data structures and algorithms. Pearson, 1983. (Russ. ed.: *Struktury dannykh i algoritmy*. Moscow, Vil'yams Publ., 2000.)

Kiriakov E.A. — Student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Levinsky A.T. — Student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Ivanov A.M., Assist. Professor, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.