УДК 681.3

МУРАВЬИНЫЙ АЛГОРИТМ ДЛЯ РАЗБИЕНИЯ ГРАФОВ

Ю.В. Ладыженский , Р.А. Родригес Залепинос
Донецкий национальный технический университет

Реализован алгоритм с использованием парадигмы колонии муравьёв для поиска и стягивания подмножеств вершин (кластеров) при укрупнении графа в многоуровневой схеме разбиения графов. Описана методика проведения экспериментов и представлены их результаты.

Проблема разбиения графов возникает в различных формах в data mining, робототехнике, искусственном интеллекте, переупорядочивании разреженных матриц, обработке изображений, компьютерных сетях, параллельных вычислениях, оптимизации кэширования, проектировании СБИС, базах данных, операционных системах и других областях.

Пусть дан неориентированный граф $G = G(V, E)$, где V − множество вершин, имеющие веса, $|V'|$ сумма весов вершин из $V' \subseteq V$, а E − множество рёбер (v, w) с весами $w(v, w) \in \Re$, $w(V_1, V_2) = \sum_{v_1 \in V_1, v_2 \in V_2} w(v_1, v_2)$, $V_1, V_2 \in V$. Для $(v, w) \notin E$ $w(v, w) = 0$, Проблема разбиения графа на k ($k \in N$, $k \geq 2$) частей и заданного баланса $\Theta \in \Re$, $\Theta > 0$ состоит в поиске набора попарно непересекающихся подмножеств $V_1 \cup V_2 \cup ... \cup V_k = V$ таких, что $\max(k |V_i| / |V|, 1) - 1 \leq \Theta$, и минимизации $w(V_i, V_j)$ $\forall i, j$.

В практических задачах возникают графы с большим числом вершин ($\approx 2 \cdot 10^3 \div 7 \cdot 10^6$), поэтому для их разбиения успешно применяется многоуровневая парадигма [1-5]. Она позволяет получать «хорошие» решения за «разумное» время. Вначале исходный граф G обрабатывается рекурсивной функцией, результатом работы которой является последовательность графов $G, G_1, G_2, ..., G_n, G'$, где каждый последующий граф получается укрупнением предыдущего графа, путём попарного стягивания вершин. Граф G' разбивается на требуемое число подмножеств. Затем выполняются обратные действия, назначая вершинам v и w, стянутым в вершину u то же подмножество разбиения, что и для u. При укрупнении стягиваются вершины, вероятность которых быть помещёнными в разные подмножества разбиения мала. Процесс попарного стягивания имеет некоторую природу случайности и локальность, поэтому зависит от нумерации вершин. Предлагается вместо этого стягивать сразу несколько вершин (кластеры), в надежде, что это устранит зависимость от нумерации вершин и повысит качество решения.

Реализованный алгоритм состоит из двух этапов. Этап 1 является муравьиным алгоритмом, цель которого изменить веса рёбер так, чтобы потом легко отличить кластеры. Этап 2, используя новые веса рёбер, выделяет кластеры вершин и стягивает их.

Муравьи являются социальными насекомыми, живущими в высокоорганизованных колониях. Муравьи могут находить кратчайшие пути между источниками пищи и своим гнездом. Когда муравьи перемещаются от источника пищи к гнезду и в обратном направлении, они выделяют химическое вещество феромон, которое остаётся на земле, формируя след. Двигаясь, муравьи выбирают, с некоторой вероятностью, пути с сильными концентрациями феромона. След феромона помогает муравьям находить путь обратно к гнезду и достигать источников пищи, найденных своими собратьями. Феромон имеет свойство накапливаться и его концентрация на пути пропорциональна числу прошедших по нему муравьёв. Со временем феромон испаряется с редко посещаемых путей. Искусственные муравьи обладают некоторым объёмом памяти, функционируют в дискретном времени, перемещаются по рёбрам графа и способны анализировать списки смежности вершин графа [6-8].

Пусть на вход Этапа 1 подан граф $G = G(V, E)$. Допустим, муравей выбирает вершину для своего следующего шага случайно, причём вероятность выбрать $w \in V$ пропорциональна весу $w(v, w)$, если муравей находится в $v \in V$. Можно предположить, что если v принадлежит кластеру $C \subseteq V$, тогда вероятность из v попасть в $w \in C$ должна быть высока $p = w(v, C \cap Adj[v]) / w(v, Adj[v])$, где $Adj[v]$ множество вершин, смежных с v. Следовательно, C захватывает муравьёв: попав в кластер, они будут долго циркулировать в нём; за продолжительное время число покинувших кластер муравьёв много меньше прибывших и находившихся в нём за это время.

Будем использовать новую функцию $w'(v, w) = w(v, w) * pher(v, w)$, определяющую вес ребра, где $pher(v, w) \in \Re, pher(v, w) \geq 1$ − концентрация феромона на ребре (v, w). Муравей помнит последние Δt_2 вершин своего пути, для чего сохраняет их в очереди $q = < v_1, v_2, ..., v_{\Delta t_2 - \Delta t_1}, v_{\Delta t_2 - \Delta t_1 + 1}, ..., v_{\Delta t_2} >$, и ему запрещено возвращаться в вершины, посещённые за последние Δt_1 шагов, $\Delta t_1 < \Delta t_2$. Вероятность перехода муравья в вершину, смежную с $v_{\|q\|}$ пропорциональна её связности с ранее посещёнными $\sum_{w\in Adj[v_{\|q\|}],v\in q\cap Adj[w]} w'(v,w)$, где $\|q\|$ число вершин в q. Когда q заполнена Δt_2 вершинами, то при занесении новой вершины её индекс равен Δt_2, и $v_i = v_{i+1}, i = 1,...,\Delta t_2 - 1$. Когда муравей переходит в вершину $v_t \in q$, то фиксируется факт обнаружения кластера. На основании вершин $v_t,...,v_{\|q\|}$ выделяется подграф $G' = G'(V',E')$, $V' \subseteq V$, $v_t,...,v_{\|q\|} \in V'$, $E' = \{(v,w): v,w \in V', (v,w) \in E\}$. Очевидно, в общем случае $\|E\| > \|V'\|$, т.е. E' состоит не только из рёбер цикла.

Когда все муравьи сделают шаг, происходит изменение значения феромона во всех найденных кластерах на этом шаг: новое значение $pher(v,w)$ полагается равным старому значению $pher(v,w) + \varepsilon \quad \forall (v,w) \in E'$, что является косвенной мерой качества найденного решения. Из очереди удаляются вершины с индексом большим t, при этом муравей остаётся в вершине v_t. За всё время работы алгоритма феромон не испаряется.

Этап 2 рассматриваемого алгоритма в модифицированном графе $G = G(V,E)$ классифицирует вершины на граничные (B), внутренние (I) и не принадлежащие какому-либо кластеру (C). Множества $B, I, C \subseteq V$, попарно не пересекаются. Если

$$\max_k w'(v, Adj[v][k]) / w'(v, Adj[v][k+1]) \geq \varphi_{\max} \tag{1}$$

то $v \in B$ где $\varphi_{\max}$ – некоторый параметр, обычно $\varphi_{\max} \geq 2$, а $Adj[v][i]$ индекс i-ой смежной вершины с v, причём $w'(v, Adj[v][i]) \leq w'(v, Adj[v][i+1])$ для $\forall v \in V$. Предполагается, что индекс k, найденный из неравенства (1) для вершины v отделит особо тяжёлые от резко отличающихся от них лёгких рёбер. Считается, что если $v \in B$, то v принадлежит пока что неизвестному кластеру Q, и $Adj[v][i] \in Q$, $i = \overline{0,k}$. Если для v неравенство (1) не может быть удовлетворено ни при каком k, то $v \in I \cup C$. Пусть для $v \in B$ $v^+ = \{Adj[v][i] : i = \overline{0,k}\}$, а $v^- = Adj[v] \setminus v^+$. Для $v \in I \cup C$ $v^+ = Adj[v]$.

Путь $< x_1, x_2, ..., x_n >$ в графе G положителен ($x_i \in V, i = \overline{1,n}$), если $x_{i+1} \in x_i^+$, $i = \overline{1,n-1}$. Множество $Q \subseteq V$ максимально, если для $\forall v \in V \setminus Q$ не существует положительного пути от v к w, если $v \in B$ либо от z к w через v, если $v \notin B$, где $w \in Q$, а $z \in B$ (т.е. $\forall v \notin Q$ нельзя присоединить к Q). Также, если $v \in Q$ и $w \in v^+$, то $w \in Q$. Если

$\exists Q' \subset Q$, $Q' \neq \varnothing$ такое, что Q' максимально, то Q избыточно. Подмножество $Q \subseteq V$ называется кластером (с точки зрения Этапа 2), если оно максимально и не избыточно. Этап 2 выделяет кластеры из V (можно доказать, если Q_1 и Q_2 – различные кластеры, то $Q_1 \cap Q_2 = \varnothing$). В одном кластере не может быть рёбер, сильно отличающихся друг от друга по весу. Пусть величина me равна весу максимального ребра в найденном кластере Q. Если $v \in Q$ и $me / Adj[v][0] \geq \varphi_{\max}$, то v удаляется из Q. В общем случае, подграф из вершин Q графа G становится несвязным и распадается на части $Q_1, Q_2, ..., Q_n$, которые стягиваются по отдельности.

Реализованы оба этапа описанного выше алгоритма. Для задания графа в пакетах разбиения графов используются текстовые файлы. В них построчно помещаются списки смежности вершин графа. Если для изображённого на бумаге графа выполнять это вручную, то легко допустить ошибку. Существуют утилиты, позволяющие задать граф визуально, однако они хранят описание графа в файлах, форматы которых специфичны для каждой утилиты.

Для проведения экспериментов, визуального наблюдения за процессом работы алгоритма и снятия результатов в наглядной форме используется PowerPoint. Он является COM-сервером, которым можно управлять непосредственно из собственного приложения и получать доступ к изображённому на слайде графу (с помощью эллипсов и соединительных линий) через соответствующие классы (как если бы граф был уже загружен из файла). При работе Этапа 1 после выполнения заданного числа итераций, можно визуализировать интенсивность феромона на рёбрах через толщину соединительных линий. Если граф обрабатывает один муравей, то можно проследить пошагово его перемещение из вершины в вершину (текущая вершина и вершины очереди выделяются особыми цветами).

Проведенные эксперименты на небольших графах (в PowerPoint) показывают, что муравьи ошибаются (в силу своей стохастической природы), наращивая феромон на рёбрах, не входящих в кластеры: некоторые вершины становятся граничными, а некоторые перестают быть таковыми. При этом вершины кластеров становятся неотличимыми от соседних, сливаясь на Этапе 2 в огромный кластер, вызывая большой разброс в весах попавших в него рёбер, приводя к удалению практически всех вершин из кластера. Находится несколько огромных кластеров, поэтому стягивается ничтожное число вершин по сравнению с их общим количеством, а укрупнённый граф по числу вершин получается не намного меньше исходного. Для уменьшения числа ошибок возможна модификация правил движения муравья. Также следует разработать более чувствительные схемы построения кластеров.

Проведены эксперименты с использованием METIS [9] на графах из архива C. Walshaw [10]. После Этапа 1 обработки графа он посылался в METIS для нахождения бисекции. Находилась также другая бисекция, последовательно обрабатывая тот же граф Этапом 1, Этапом 2 и METIS. Лучшая из этих двух бисекций сравнивалась с бисекцией, полученной непосредственным применением METIS к исходному графу. Для графа «data» получено разбиение на 5% лучше, но с ухудшением баланса на 2%. Для графа «add32» разбиение улучшено на 16%, а баланс на 2.78%. Это разбиение совпало с лучшим из известных на сегодняшний день.

Предложенный метод по качеству получаемого решения сравним с существующими. Он может применяться для негарантированного улучшения качества разбиений, находимых METIS, дополнительно затрачивая несколько минут на предварительную обработку графа описанным алгоритмом.

Литература

1. Shloegel K., Karypis G., Kumar V., Parallel multilevel algorithms for multi-constraint graph partitioning, 1999
2. Soper A.J, Walshaw C., Cross M., A combined evolutionary search and multilevel optimization approach to graph partitioining, 2004
3. Karypis G., Kumar V., A fast and high quality multilevel scheme for partitioning irregular graphs, 1998
4. Karypis G., Kumar V., Multilevel k-way partitioning scheme for irregular graphs, 1998
5. C. Walshaw, Multilevel refinement for combinatorial optimization problems, 2004
6. Dorigo M., Real ants inspire ant algorithms, 2004
7. Dorigo M., Maniezzo V., Colorni A., The ant system: optimization by a colony of cooperative agents, 1996
8. Dorigo M., Caro G., Gambardella L., Ant algorithms for discrete optimization, 1999
9. http://www.cs.umn.edu/~karypis/metis
10. http://staffweb.cms.gre.ac.uk/~c.walshaw

Получено 01.06.07