Графическое параллельное программирование на примере задачи выявления сообществ в графах

Аркадий Климов

Arkady.Klimov@gmail.com

Институт проблем проектирования в микроэлектронике РАН

GraphHPC-2016

МГУ им. М.В.Ломоносова, Москва

3 марта 2016

История появления доклада

На GraphHPC-2015 был представлен доклад Александра Позднеева «**Алгоритм выделения сообществ в графах для параллельных компьютеров с распределенной памятью**» – по материалам статьи X.Que, F. Checconni, F.Petrini, J.Gunnels "Scalable community detection with the Louvain algorithm" [QCPG], которая на тот момент еще не была опубликована. В нем предлагалось использование хеш-таблиц для поддержки обмена информацией.

Я увидел много сходства с моделью вычислений потока данных (ПД), где *в реализации* используются хеш-таблицы при приеме сообщений. Захотелось выразить этот алгоритм на языке потоков данных.

Летом я получил текст статьи [QCPG] и смог реализовать свое желание. Детали алгоритма уточнялись в прямой переписке с авторами. Результат предлагаю вашему вниманию.

Основная задача доклада – на примере данного алгоритма показать изобразительные средства языка программирования. Для облегчения понимания используется графическая форма языка ПД. Язык находится в процессе становления.

Сам алгоритм не существен, но нужен для понимания изображаемого в языке. Поэтому с него и начнем (очень коротко).

План доклада

- Понятия о графах
- Задача о разбиении графа и модулярность
- Лувенский метод
  A. Общая идея
  B. Последовательный алгоритм
  C. Параллельный алгоритм
- Параллельный метаязык (Pamela)
- Параллельный алгоритм выявления сообществ (Лувенским методом) на графическом языке Pamela-G (на одном слайде)

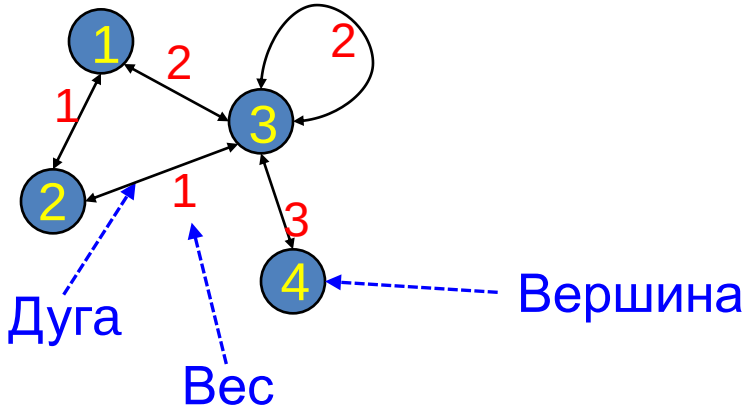
Графы. Общие сведения

Граф G = (V,E,W)

V – множество вершин (*u*,*v*,…)

E – множество дуг (*u*→*v*)

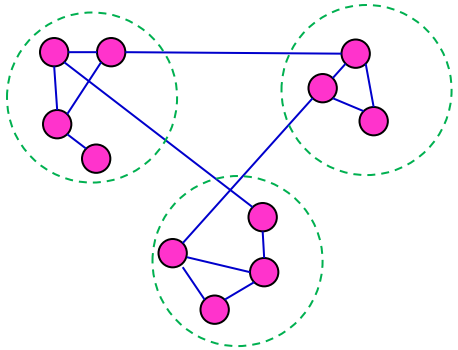
W: E → R (весовая функция)



- n=|V| (число вершин)
- Дуги имеют кратности, или веса (0, если дуги нет)
- Граф симметричный: $w_{uv} = w_{uv}$.
- Граф задан весовой матрицей W:
- На диагонали – удвоенный вес.
- Вес вершины: $w_u = \sum_v w_{uv}$

0	1	2	0
1	0	1	0
2	1	4	3
0	0	3	0

Модулярность – мера качества разбиения



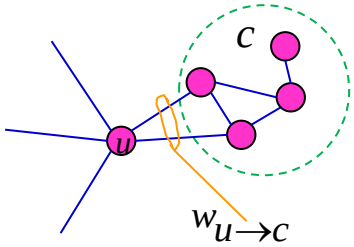
Кластеризация – разбиение на непересекающие подмножества (сообщества).
Хорошо, если внутренних связей много, внешних мало.
Модулярность – мера качества разбиения

Modularity
[M.E.J. Newman, 2004]

% внутренних дуг (по числу концов)

% внутренних дуг для случайного графа при тех же степенях вершин

$$Q = \sum_{c \in C} \left[\frac{\Sigma_{in}^{c}}{2m} - \frac{\left(\Sigma_{tot}^{c}\right)^2}{4m^2} \right]$$

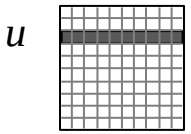


Выигрыш от присоединения изолированной вершины u к сообществу C :

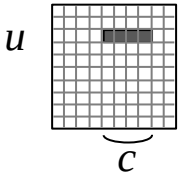
$$\Delta Q_{u \to c} = \frac{w_{u \to c}}{2m} - \frac{w_u * \Sigma_{tot}^{c}}{2m^2}$$

Используемые величины (обозначения из статьи X.Que et al):

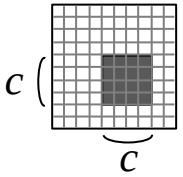
$w_u = \sum_{v \in V} w_{uv}$



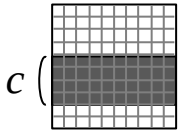
$w_{u \to c} = \sum_{v \in c} w_{uv}$



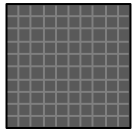
$\Sigma_{in}^{c} = \sum_{u \in c} w_{u \to c}$



$\Sigma_{tot}^{c} = \sum_{u \in c} w_u$

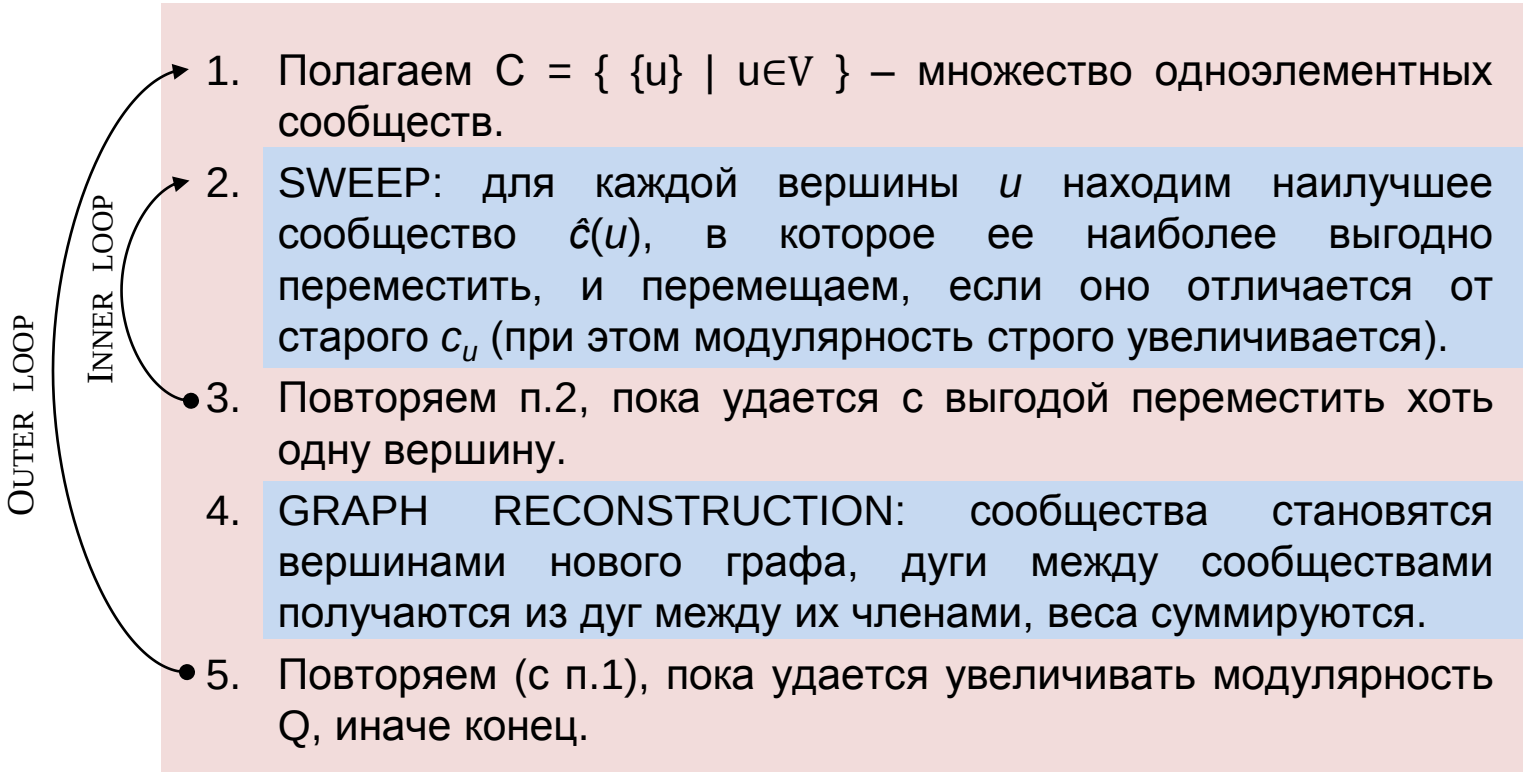


$2m = \sum_{u \in V} w_u$



Лувенский алгоритм. Идея.

Предложен группой студентов, обучавшихся в Лувенском университете (Бельгия) до 2008 года

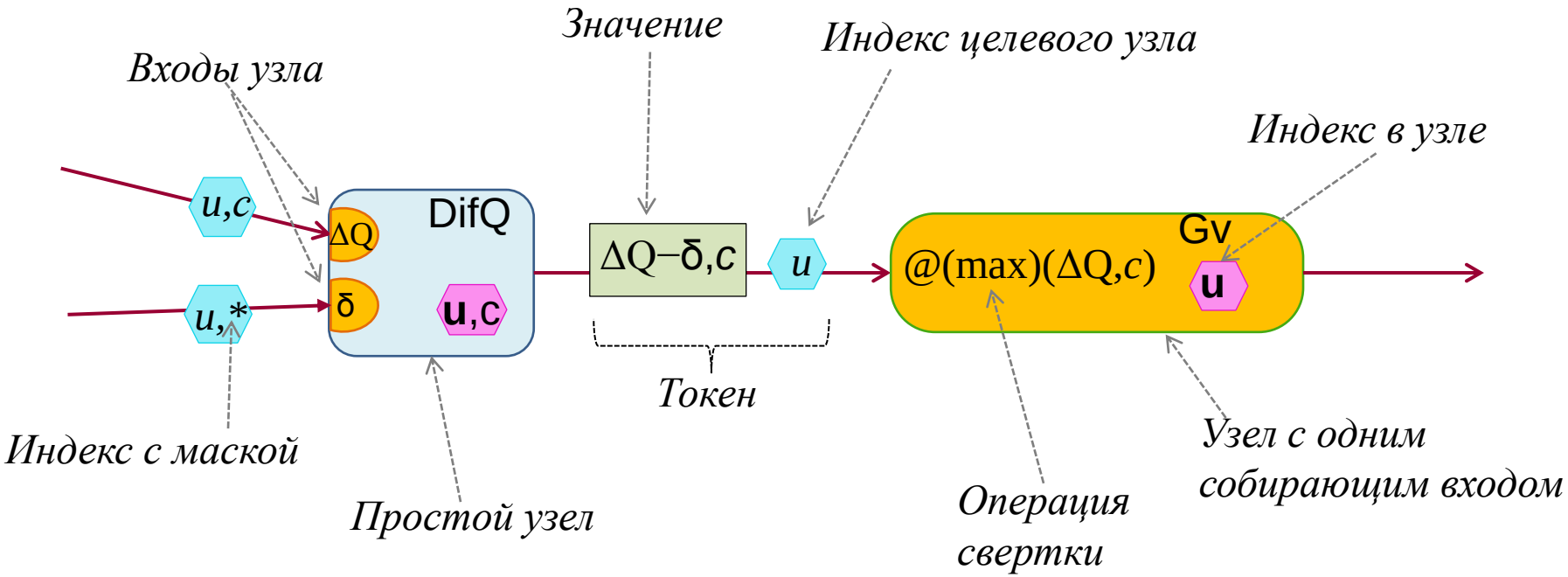


Можно распараллелить пп. 2 и 4, но без гарантии монотонности в п.2

Проблема 1. Вычисление $w_{u \to c}$ *в распределенной среде.*

Проблема 2. Конфликты: одни перемещения влияют на величину выигрышей других перемещений. Возможны проблемы со сходимостью.

Элементы языка Pamela-G

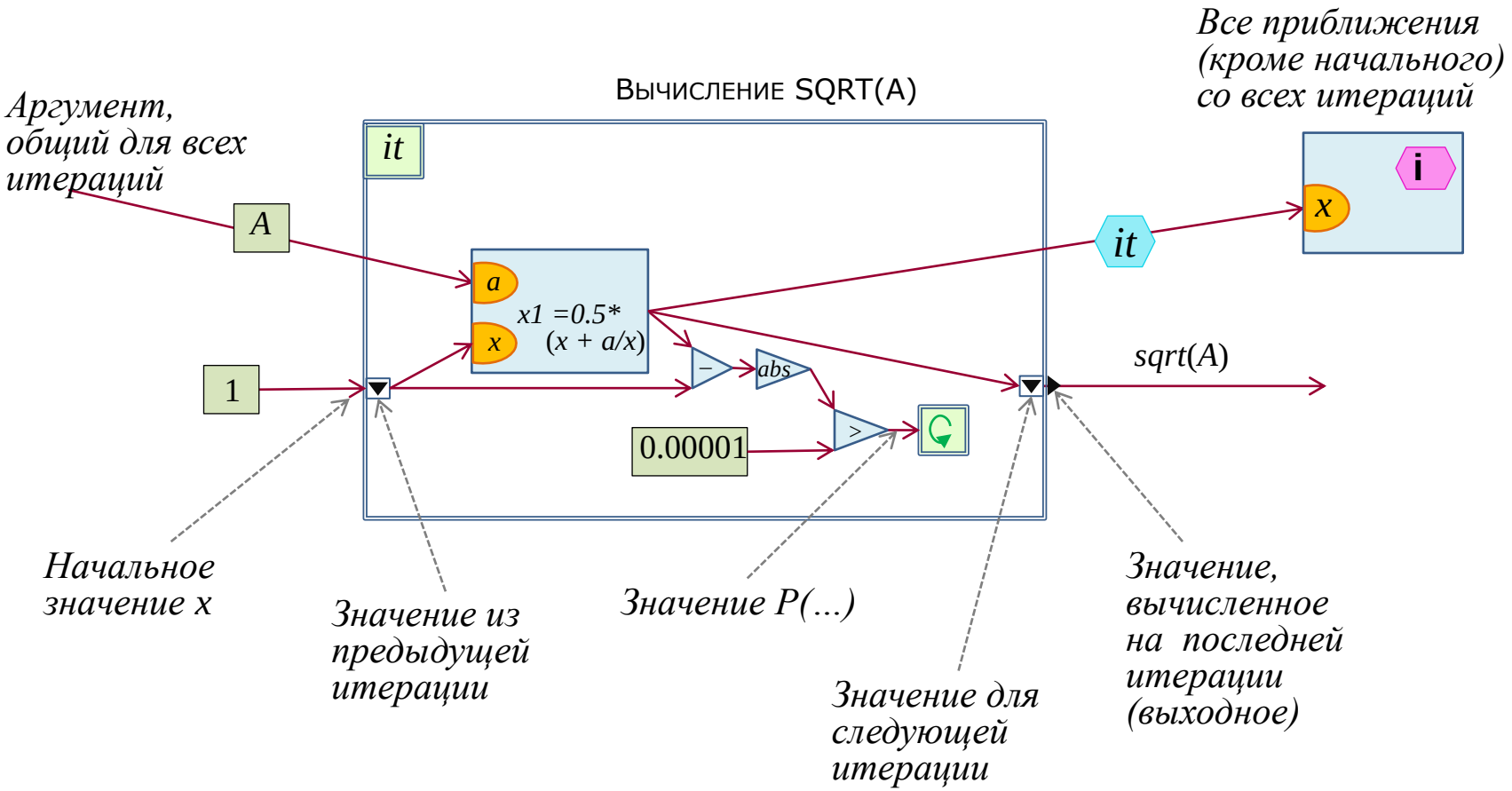


Узлы имеют индексы i и передают друг другу токены $\xrightarrow{val}\ i \rightarrow$.
Выражения в токене вычисляются в среде узла-отправителя.
Семейства однотипных узлов реализуются хеш-таблицами по индексам.
Узел (с конкретным индексом i_1) срабатывает (и выполняет предусмотренные в нем действия), когда прибывшие токены присутствуют на всех его входах x.
Элемент хеш-таблицы создается по приходу первого токена с данным индексом.
Собирающий узел (вход) @(op)name : по первому токену элемент создается, последующие к нему добавляются.

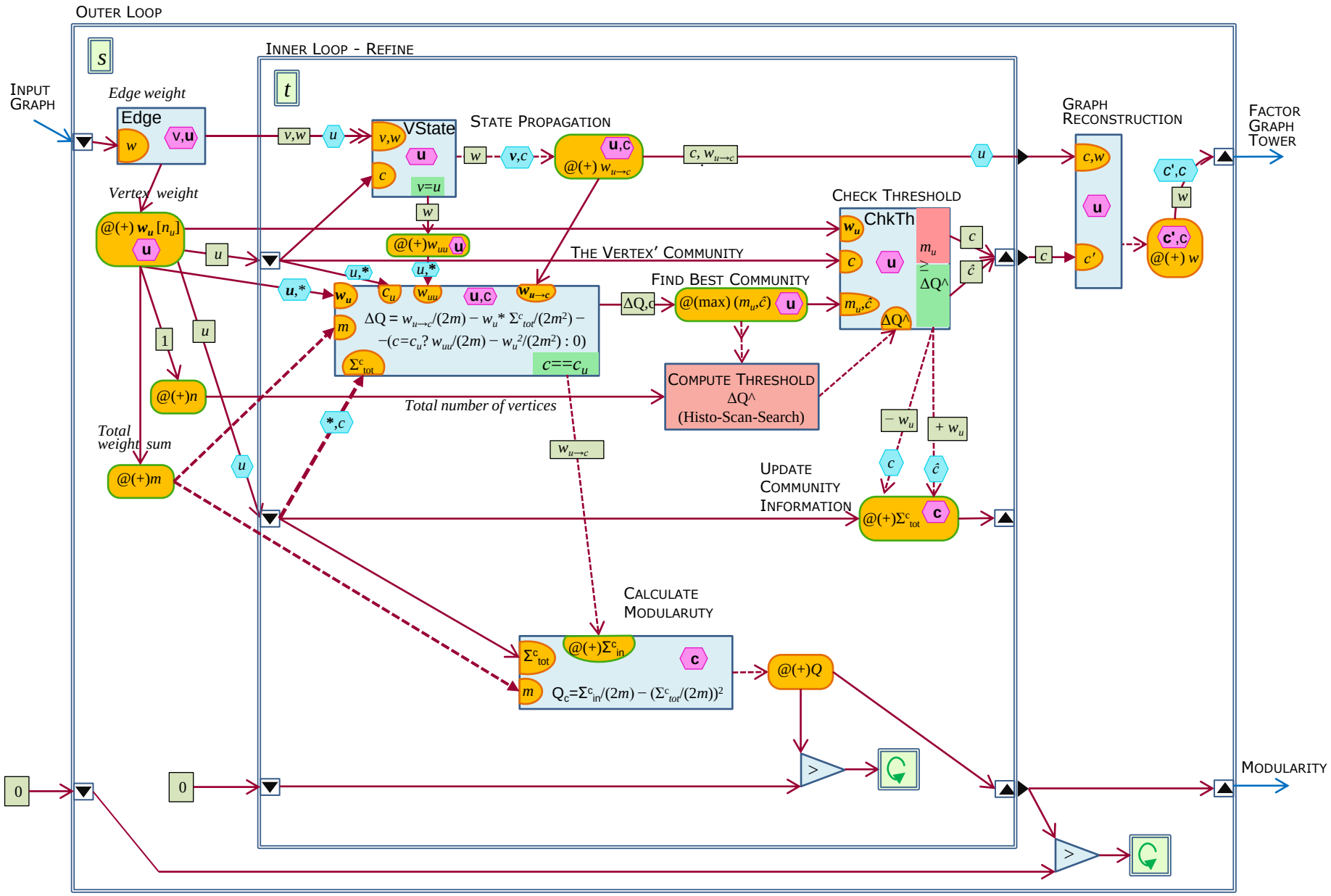
Элементы языка Pamela-G. Итерация

(средство заимствовано из системы LabVIEW от NI)

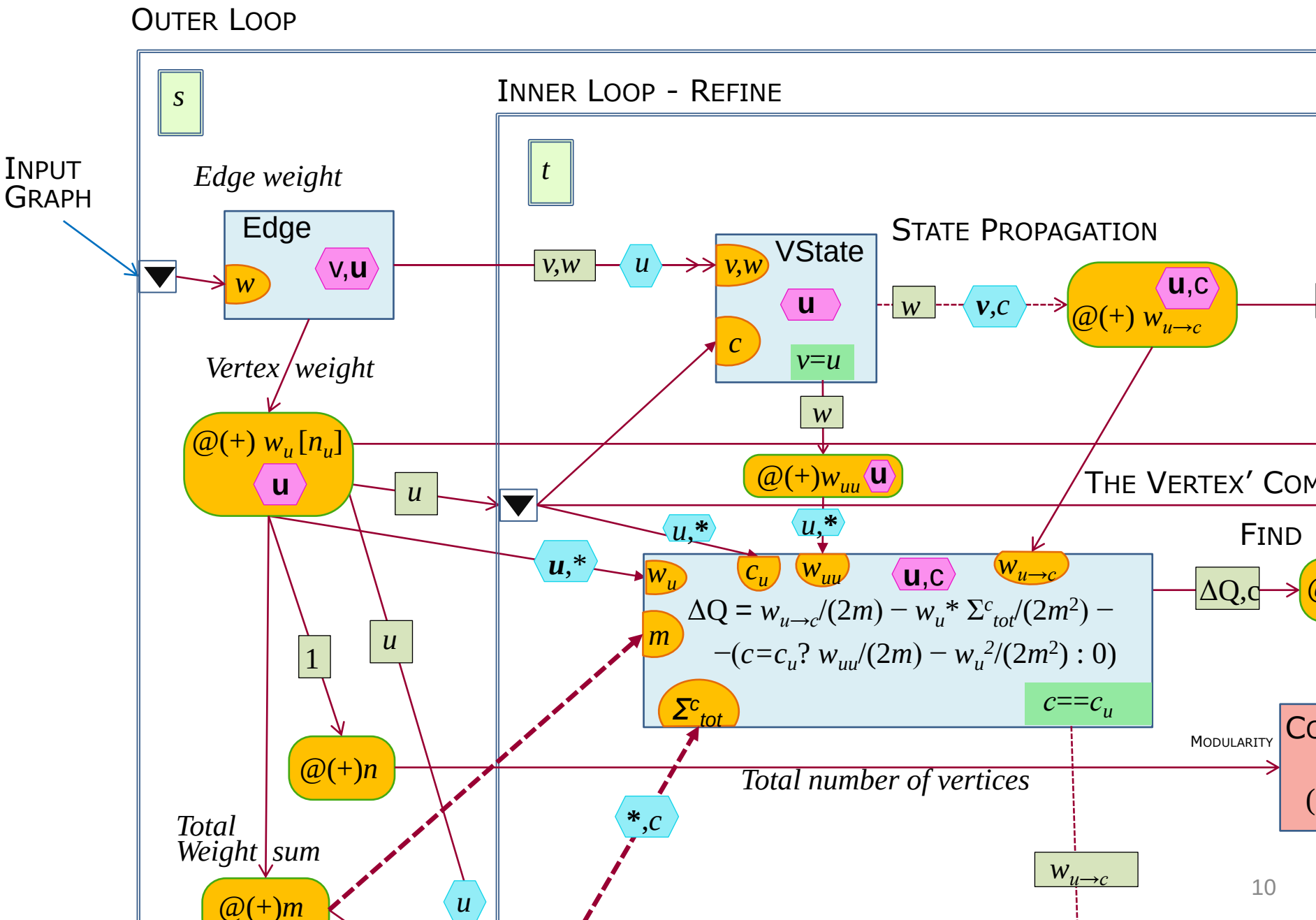
for (*it*=0; *P*(…) ; *it*++) {…}



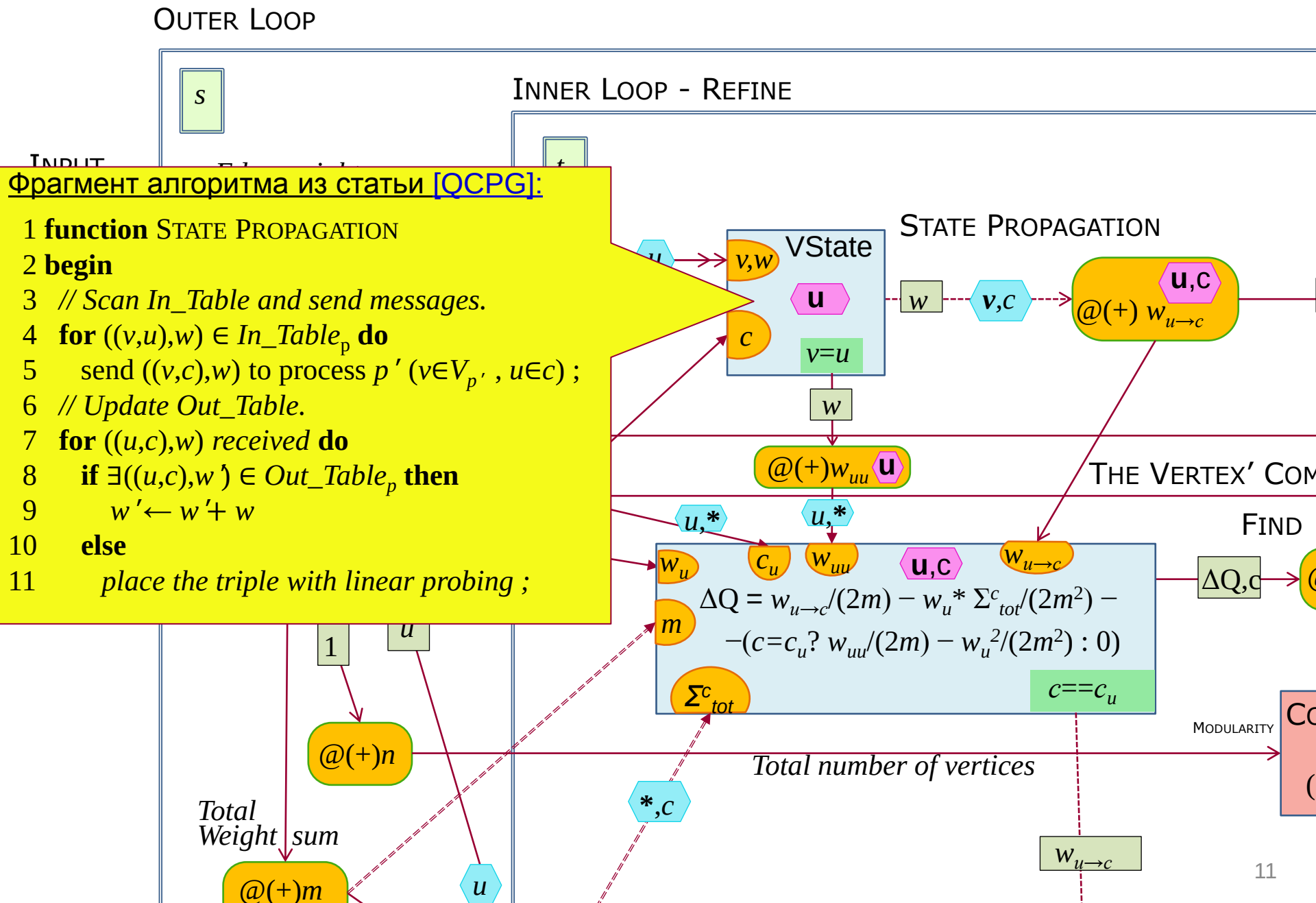
The Louvain Algorithm Flowgraph



The Louvain Algorithm Flowgraph (the previous slide zoomed)



The Louvain Algorithm Flowgraph (the previous slide zoomed)



Фрагмент алгоритма из статьи [QCPG]:

```
1 function STATE PROPAGATION
2 begin
3   // Scan In_Table and send messages.
4   for ((v,u),w) ∈ In_Table_p do
5     send ((v,c),w) to process p' (v∈V_p' , u∈c) ;
6   // Update Out_Table.
7   for ((u,c),w) received do
8     if ∃((u,c),w') ∈ Out_Table_p then
9       w' ← w' + w
10    else
11      place the triple with linear probing ;
```

Заключение

1. Представлен язык PAMELA(G) = PArallel MEta LAnguage.
2. Предназначен для выражения чистой математической структуры алгоритмов. Описание может быть исполнено!
3. Сверхзадача – уметь быстро конкретизировать до эффективных программ в разных платформах и парадигмах, дополнительно управляя распределением, представлением информации и т.п.
4. Индексирование (узлов и токенов) – основная «фича» языка, расширение старой парадигмы dataflow. Обеспечивает:
   - семантическую выразительность (ср. Map-Reduce)
   - управление распределением по пространству-времени в распределенных системах.
5. Графическая форма улучшит обзорность кода, повысит продуктивность разработки, отладки, анализа, поиска «узких мест» и т.п.
6. Надо больше экспериментировать, пробовать выражать разные алгоритмы, проводить их преобразования в традиционные языки.
7. Приглашаем к сотрудничеству! Приходите со своими задачами!
8. Только совместно сможем создать мощный инструмент!

Ссылки

1. M. E. J. Newman. Analysis of weighted networks. Physical review E, vol. 70, no. 5, p. 056131, Nov. 2004. *– вводит понятие модулярности.*
2. V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10):P10008, 2008. *– последовательный Лувенский алгоритм.*
3. S. Fortunato. Community detection in graphs. Physics Reports, 486, no. 3-5, pp. 75 – 174, 2010. *– очень хороший обзор методов.*
4. Xinyu Que, Fabio Checconi, Fabrizio Petrini, John A. Gunnels. Scalable Community Detection with the Louvain Algorithm. 2015 IEEE 29th International Parallel and Distributed Processing Symposium, pages 28–37, May 2015. *– статья, вдохновившая данный доклад [QCPG].*
5. http://labview-rus.blogspot.ru/ . *– о LabVIEW по-русски.*