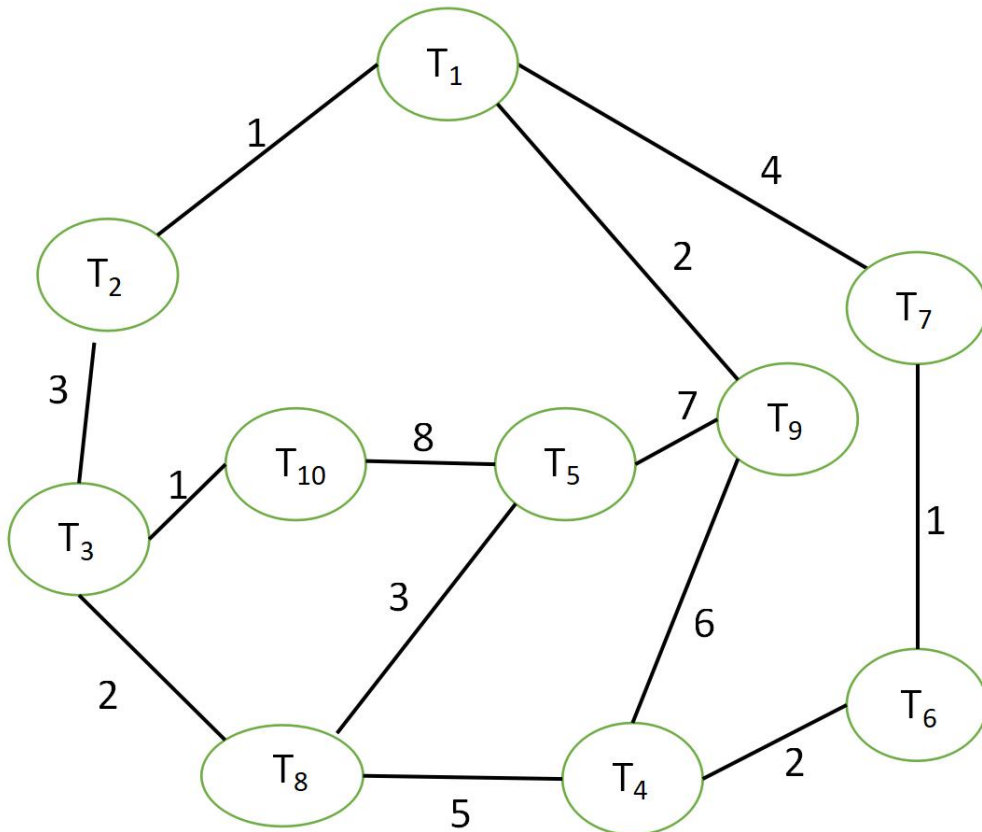


1 实验目的

- 编程实现多模块化划分算法MMM（单链接、全链接、均链接），输入是通信代价、模块数、每个模块最大任务数，输出为划分好的模块，并给出划分代价

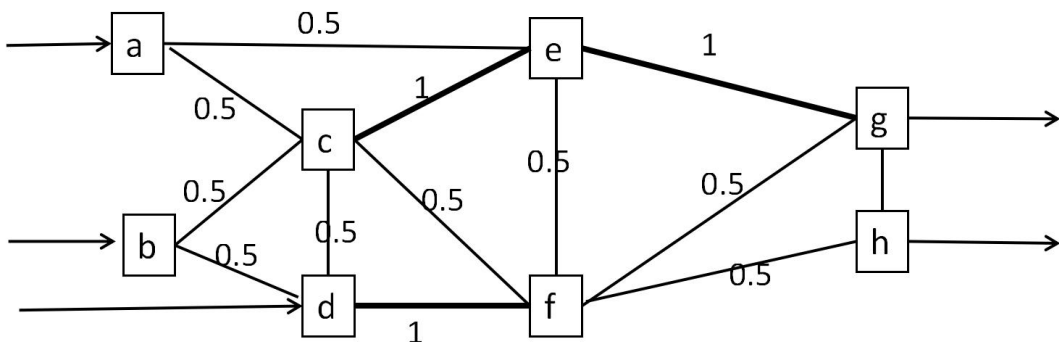
1.1 任务1

- 使用实现的程序将下图的10个任务划分成4个模块，每个块内任务个数不超过3



1.2 任务2

- 使用实现的程序将下图中的8个任务a, b, c, d, f, g, h分成2个模块，每个模块含有4个任务，使得模块间通信代价最小，任务间通信代价见下表



Cxy	a	b	c	d	e	f	g	h
a	0	0	0.5	0	0.5	0	0	0
b	0	0	0.5	0.5	0	0	0	0
c	0.5	0.5	0	0.5	1	0.5	0	0
d	0	0.5	0.5	0	0	1	0	0
e	0.5	0	1	0	0	0.5	1	0
f	0	0	0.5	1	0.5	0	0.5	0.5
g	0	0	0	0	1	0.5	0	0.5
h	0	0	0	0	0	0.5	0.5	0

2 算法设计

- 针对单链接、全链接、均链接实现对应的多模块化划分算法

2.1 单链接

- ```
map["T1"+"T2"] = q

<vector<vector>> v --> {{1,2},{2},{3}...}

v[0].size > 2 --> {"T5","T6","T7"}...}
map["T5"+"T6"+"T7"] = q --> max(q) min(q) average(q)
```

•

### 2.2 全链接

---

•

•

### 2.3 均链接

---

•

•

## 3 实验过程

- 定义数据输入格式
- 执行算法
- 获得结果

### 3.1 数据输入格式

- |         |                               |
|---------|-------------------------------|
| n       | n: 任务数                        |
| loop n: |                               |
| Ti k    | Ti: 任务i; k: k条边               |
| loop k: |                               |
| Tj q    | Tj: 与任务i存在关系的任务j; q: 两个任务边的权值 |

#### 3.1.1 表征任务1输入

- ```
10
T1 3
T2 1
T7 4
T9 2
T2 2
T1 1
T3 3
T3 3
T2 3
T8 2
T10 1
T4 3
T6 2
T8 5
T9 6
T5 3
T8 3
T9 7
T10 8
T6 2
T4 2
T7 1
T7 2
T1 4
T6 1
T8 3
T3 2
T5 3
T4 5
T9 3
T4 6
T5 7
T1 2
T10 2
T3 1
T5 8
```

3.1.2 表征任务2输入

- 8
a 2
e 0.5
c 0.5
b 2
c 0.5
d 0.5
c 5
a 0.5
b 0.5
d 0.5
f 0.5
e 1
d 3
b 0.5
c 0.5
f 1
e 4
a 0.5
c 1
f 0.5
g 1
f 5
d 1
c 0.5
e 0.5
g 0.5
h 0.5
g 3
e 1
f 0.5
h 0.5
h 2
f 0.5
g 0.5

3.2 实验结果

3.2.1 任务1

3.2.2 任务2

4 实验结论
