# Latex代码

Li Wentao

2024 年 7 月 17 日

## 1 竞争权重

**算法 4-1:** 竞争权重

```
输入: 利润需求集合$\xi_j^i$,移动距离集合$len_j^i$,承诺数据质量集合$R_j^i$,质量
      贡献$\alpha$,报价贡献$\beta$,质价比例系数$\gamma$
输出: 竞争权重集合$W_j^i$
for $\tau_j^i \in \tau_i$ do
    for $\omega_a \in \omega$ do
        $Eb_{j,a}^i = e^{\lambda_i * R_{j,a}^i} + e^{\mu_i * len_{j,a}^i} - 2$
        $P_{j,a}^i = Eb_{j,a}^i + \xi_{j,a}^i$
        $W_j^i[a] \leftarrow \frac{\alpha * R_{j,a}^i}{\beta * P_{j,a}^i} = \gamma * \frac{R_{j,a}^i}{e^{\lambda_i * R_{j,a}^i} + e^{\mu_i * len_{j,a}^i} - 2 + \xi_{j,a}^i}$
    end
end
```

## 2 TWPM算法

**算法 4-2:** TWPM算法(Task Worker Perfect Match)

```
输入: 参与者$\omega_u$
输出: 匹配任务$\tau_v^i$ (无匹配则为0)
for $\tau_v^i \in G^i[\omega_u]$ do
    if $\tau_v^i$ has been visit then
        continue
    end
    set $\tau_v^i$ is visited
    if $\tau_v^i$ has not be matched or TWPM($match[\tau_v^i]$) then
        $match[\tau_v^i] \leftarrow \omega_u$
        return $\tau_v^i$
    end
end
```

## 3 信任工人分配算法

**算法 4-3:** 信任工人分配算法

```
输入: 初始任务图$G^i$,信任工人集合$\omega^t$
输出: 信任工人任务分配图$G_t^i$
辅助数组: 记录数组$vis$,匹配数组$match$
for $\omega_u^t \in \omega^t$ do
    $initial\ vis$ , $the\ elements\ are\ false$
    $\tau_v \leftarrow TWPM(\omega_u^t)$
    if $\tau_v$ is not zero then
        $G_t^i[\omega_u^t] \leftarrow \tau_v$
    end
end
```

# 4 剩余工人分配算法

**算法 4-4:** 剩余工人分配算法

**输入:** 初始任务图$G^i$,信任工人分配图$G_t^i$,第i次任务$T^i$,成本因子$Cf_i$,分辨强度$Re_i$,竞争权重集合$W^i$

**输出:** 工人任务分配图$G_f^i$

**辅助数组:** 匹配数组$match$

```
$G_f^i \leftarrow G_t^i$
for $\tau_u^i \in T^i$ do
    $initial\ \omega_u^i\ as\ an\ empty\ set$
    for $\omega_v^i \in G^i[\tau_u^i]$ $is\ not\ matched$ do
        $add\ \omega_v^i\ to\ \omega_u^i$
    end
    if $G_t^i[\tau_u^i]$ $is\ empty$ then
        for $\omega_v^i \in \omega_u^i$ do
            $G_f^i[\omega_v^i] \leftarrow \tau_u^i$
        end
    else
        $sort\ \omega_u^i\ by\ W^i\ from\ largest\ to\ smallest$
        $let\ L\ equals\ to\ the\ length\ of\ T_u$
        for $\omega_v^i \in \omega_u^i$ do
            if $cnt > \frac{Re_i}{Re_i+Cf_i} * L$ then
                $break$
            end
            $G_f^i[\omega_v^i] \leftarrow \tau_u^i$
            $cnt = cnt + 1$
        end
    end
end
```

# 5 RDD算法

**算法 4-5:** RDD算法(Real Data Discovery)

```
输入: 数据集合$Da^i_j$,迭代阈值$e$
输出: 标准数据$Ds^i$
辅助数组: 权重数组$w^i$
$Ds^i = Da^i_1$
$Ds^i_f = Ds^i + e$
while $|Ds^i - Ds^i_f| \geq e$ do
    $S^i \leftarrow \sum d(Da^i_{j,a}, Ds^i)$
    for $Da^i_{j,a} \in Da^i_j$ do
        $w^i_a = \log S^i - \log d(Da^i_{j,a}, Ds^i)$
    end
    $Ds^i_f \leftarrow Ds^i$
    $Ds^i = \frac{\sum w^i_a * Da^i_{j,a}}{\sum w^i_a}$
end
```

# 6 信任度更新算法

**算法 4-6:** 信任度更新算法

```
输入: 个人提交数据$Da^i_{j,a}$,偏差阈值$D^i_m$,低强度因子$f_l$,高强度因
      子$f_h$(,标准数据$Ds^i_j$)
输出: 信任度更新$tr_a$
if $Ds^i_j$ $doesn't\ exist$ then
    $Ds^i_j \leftarrow RDD(Da^i_j)$
    $D^i_a = |\frac{Da^i_{j,a} - Ds^i_j}{Ds^i_j}|$
    $tr_a = f_l * e^{D^i_m - D^i_a}$
else
    $D^i_a = |\frac{Da^i_{j,a} - Ds^i_j}{Ds^i_j}|$
    $tr_a = f_h * e^{D^i_m - D^i_a}$
end
```