

分布式过程监控

背景与意义

• 背景

1. 随着工业规模与复杂程度逐渐变大，大多数学者采用全厂过程监控，当局部设备（节点）发生故障时，会影响到一小部分的变量，而不是整个过程。
2. 设备之间通常不是独立工作的，因此考虑变量之间的依赖关联并剔除冗余有利于提升监控性能。

• 面临的问题

1. 根据过程先验知识以及监控性能设计合适的子系统分解；
2. 通过合适的分块策略维持块之间的关联信息，避免破坏变量之间的依赖关系；
3. 建立局部的监控模型有利于故障的定位以及隔离，可以方便进一步的根因溯源。

局部-全局监控

基于图论的子系统分解

• 分布式模式识别的扩展前向选择 (Extended forward selection for distributed pattern recognition, EFSDPR)

- 第一步：选择传感器

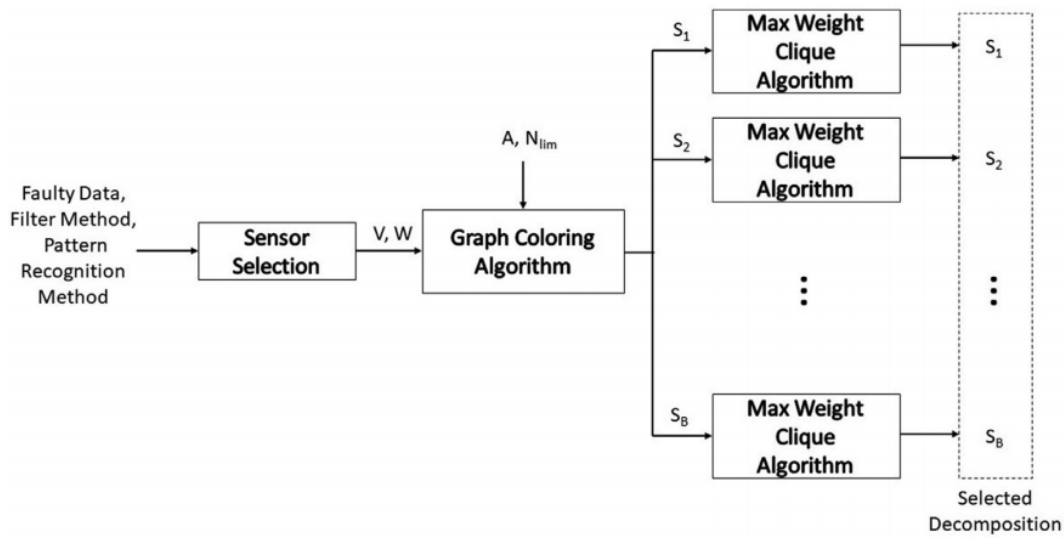
需要有标记的故障样本，用户指定一种特征选择方法，用它对故障数据进行诊断，根据传感器在区分故障类别时的**误报率**对传感器排名。

定义空集 V ，添加集 U 包含所有传感器，若误报率较低，向 V 中添加传感器，直到 U 中所有传感器被遍历；最终输出选择的传感器，还可以输出权重。

- 第二步：配置传感器

应用图着色算法找到系统的非重叠子块；

为子块选定传感器，使传感器的权重最大并满足约束条件，实质是最大权重团算法；



• 扩展子系统和传感器分配 (Extended subsystem and sensor allocation, ESASA)

- 当工厂还未运行时，需要确定监控点数量以及对应分配的传感器

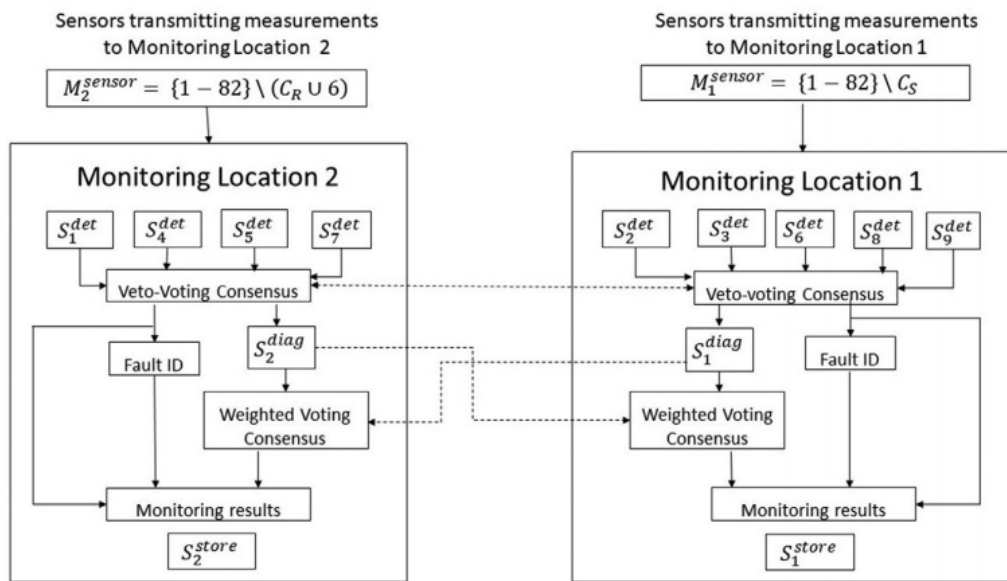
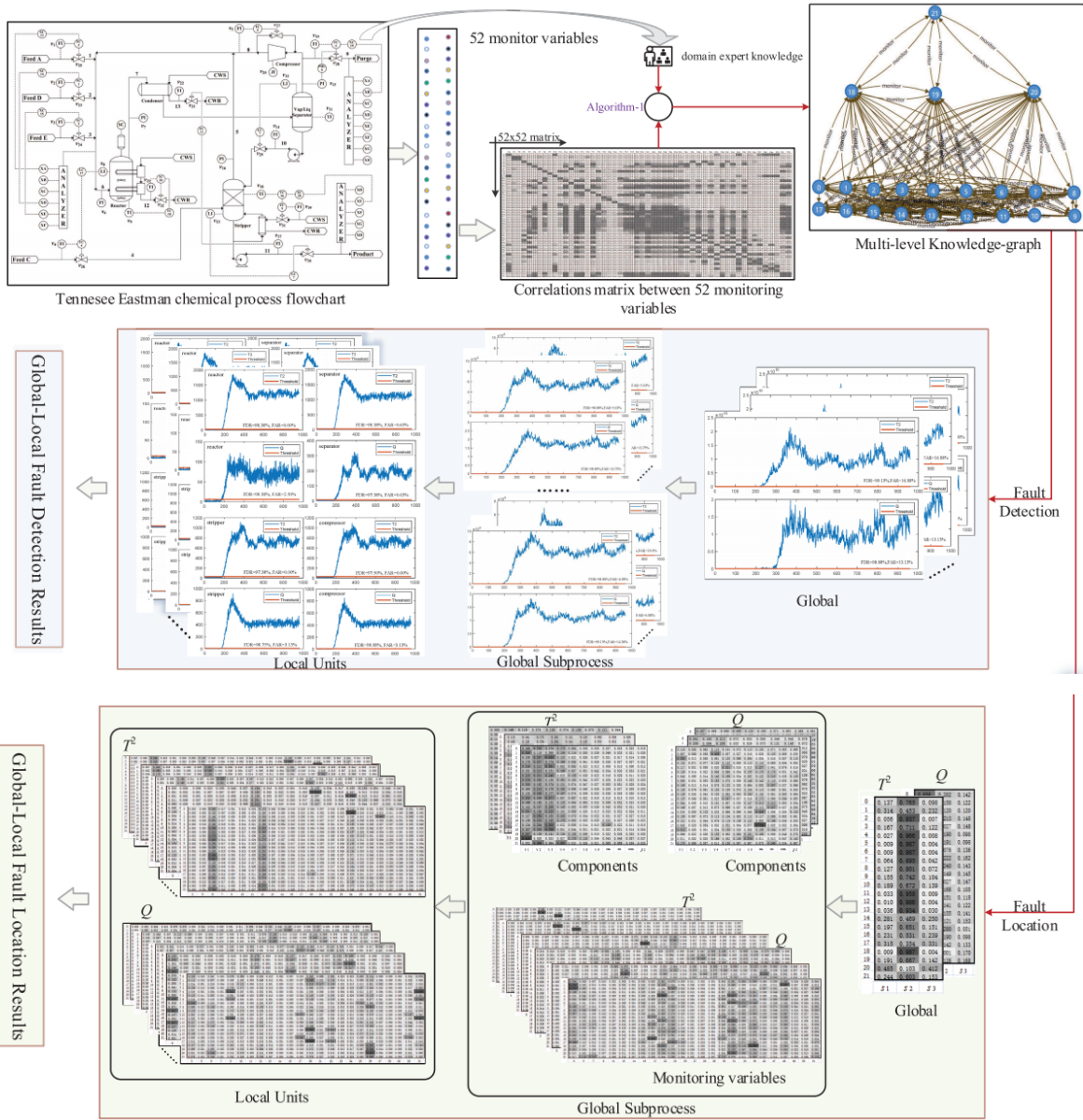


Fig. 4. Schematic of the distributed configuration used for monitoring TEP.

1. 每个子块使用测量值进行假设检验，检测到故障时，将信息传给另一个系统；
2. 使用模式识别算法诊断故障类别，并将信息传给另一个系统；
3. 使用加权投票共识机制选择要监控的子块；
4. 故障识别：找到失控传感器相对于正常状态下的偏差，对传感器进行排名，若故障并非以往诊断的类别，操作员可以根据这个信息进行诊断。

图信息聚合

多级知识图谱



$$I(S; V) = \sum_{i=1}^{l-1} \sum_{j=1}^n p(s_i, 0_j) \log_{\frac{p}{2}}(s_i) p(v_j)$$

$$NMI(S; V) = \frac{2 \times I(S; V)}{H(S)H(V)}$$

$$r_{kl} = \begin{cases} 1, & NMI_{l_1} \geq Thr \\ 0, & NMI_{l_1} < Thr \end{cases}$$

$$X_{n \times m} = TP^T + L$$

$$\begin{cases} T^2 = xPA^{-1}P^Tx^T \leq \frac{k(n-1)}{n-K} F_{\pi, n-K, \alpha} \\ Q = x(1 - PP^T)x^T < g\gamma_{h, \alpha}^2 \end{cases}$$

$$\begin{cases} X_i = Y_v \\ v = \text{arg}a_i(j) := [j | \forall t : a_{i,t} = 1] \end{cases}$$

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1l} \\ \vdots & \ddots & \vdots \\ a_{kl} & \cdots & a_{kl} \end{bmatrix}$$

1. 计算变量之间的标准化后互信息熵作为两者之间的关系；

2. 根据关系构造邻接矩阵；
3. 计算每个节点的监控变量；
4. 利用PCA进行特征分解，构造统计量、阈值，进行故障检测；
5. 计算局部变量的贡献率进行故障定位。

$$Q = ||cx|^2 = \sum_{i=1}^m Cont_i^q$$

$$Cont_i^Q = (\xi_i^r \hat{C}x)^2$$

$$T^2 = (x^T Dx) = |D^{1/2}x|^2 = \sum_{i=1}^m Cont_i^{T^2}$$

$$Cont_i^{T^2} = (\xi_i^T D^{1/2}x)^2 = x^T D^{1/2} \xi_i \xi_i^{l-1} D^{1/2} x$$