$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的多重网格方法。假设 $\boldsymbol{A}$ 是一个对称矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，一个最简单的代数多重网格算法可以通过如下步骤得到：

（1） 用一个快速近似算法得到一个近似解 $\boldsymbol{x}'$，并假设残差为 $\boldsymbol{r}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}'$。

（2） 找到一个矩阵 $\boldsymbol{S}\in\mathbb{R}^{n\times m}$，计算 $\boldsymbol{S}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{S}\boldsymbol{y}=\boldsymbol{S}^{\mathrm{T}}\boldsymbol{r}$ 的解 $\boldsymbol{y}$。

（3） 得到一个更好的近似解 $\boldsymbol{x}=\boldsymbol{x}'+\boldsymbol{S}\boldsymbol{y}$。

（4） 重复上述步骤直到残差足够小。

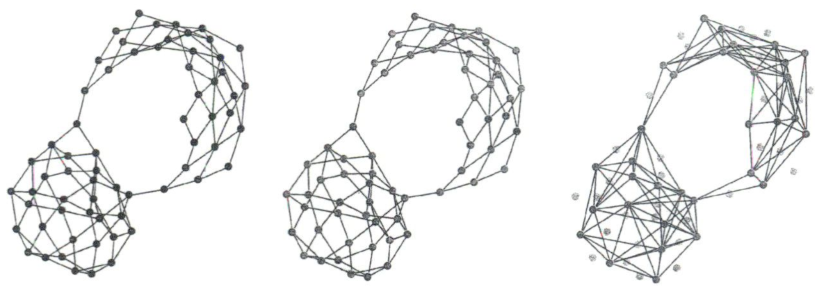
观察第二步，$\boldsymbol{S}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{S}$ 将线性系统从 n 维降到 m 维，实际上就是对图做了粗化（称为 Galerkin 粗化算子）。回到图数据中，对于一个邻接矩阵 A，它的粗化表示为

$$A_{\mathrm{c}}=\boldsymbol{S}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{S}$$

对应地，图信号 $\boldsymbol{X}$ 池化为

$$\boldsymbol{X}_{\mathrm{c}}=\boldsymbol{S}^{\mathrm{T}}\boldsymbol{X}$$

图 5.3 给出了一个例子，方便读者理解代数多重网格的粗化。



(a) 原图 G

(b) 粗化节点（红色）和其他节点（蓝色）

(c) 粗图 G_{c}

图 5.3　图的代数多重网格粗化示意图

DiffPool [72] 是一个基于代数多重网格的池化方法。不同于传统的代数多重网络，DiffPool 中的 $\boldsymbol{S}$ 可以根据上层的图信息参数化，这样我们就得到了一个端到端可学习的池化图网络。在每一层 l 中，我们定义两个不同的图神经网络层，一个图神经网络用来嵌入，另一个图神经网络用来学习粗化矩阵 $\boldsymbol{S}$。