Parallel Algorithm in Computational Mathematics SS 2008
Tutorial 3 Monday, 21 April 2008, HF 136, 12.45–14.15

Graph partitioning

In dieser Übung besprechen wir, wie man am besten ein Finite-Element Netz möglichst gleichmäßig auf eine gegebene Anzahl von Prozessoren verteilt.

Dies lässt sich in ein Problem der Graphenpartitionierung übertragen. Wir führen einen Graphen $G = (V, E)$ ein, wobei

- $V \ldots$ Eckenmenge, d.h. $V = \{v_i\}_{i=1}^{n}$ (vertices),
- $E \ldots$ Kantenmenge, d.h. $E = \{e_{ij} = (v_i, v_j)\}$ (edges).

Zusätzlich wichten wir noch jede Kante e mit einem Gewicht ω_e.

Gesucht ist nun eine Unterteilung der Eckenmenge V in $V_1, \ldots, V_p$, so dass

- $\bigcup_{i=1}^{p} = V$,
- $V_i \cap V_j = \emptyset$ falls $i \neq j$,
- $|V_i|$ ist annähernd gleich.

Die Güte der Partitionierung wird durch eine Kostenfunktion beschrieben:

$$f(V_1, \ldots, V_n, E) = \sum_{\substack{e_{ij} \in E \\ v_i \in V_p, v_j \in V_q, p \neq q}} \omega_e(e_{ij}).$$

Bei einem Finite-Elemente Netz sind die Vertices die einzelnen Finiten Elemente (entspricht der Rechenlast) und zwischen zwei Elemente besteht eine Kante, falls sie benachbart sind (dann ist zwischen beiden ein Datenaustausch erforderlich). Die Kostenfunktion beschreibt den Kommunikationsaufwand.

Der Einfachheit halber setzen wir im weiteren $\omega_e = 1$.

Man erläutere folgende Verfahren und demonstriere das Ergebnis anhand der Aufteilung des Finite-Element Netzes aus Abbildung 1 auf 4 bzw. 8 Prozessoren:

1. lineare Aufteilung,
2. rekursive Bisektion,
3. Koordinatenbisektion,
4. Greedy-Algorithmus,
5. Farhat's-Algorithmus,
6. Kernighan-Lin,
7. spektrale Bisektion.

Literatur: Ulrich Elsner: Graph partitioning: A survey. December 1997. Preprint SFB 97-27, SFB 393, TU Chemnitz.

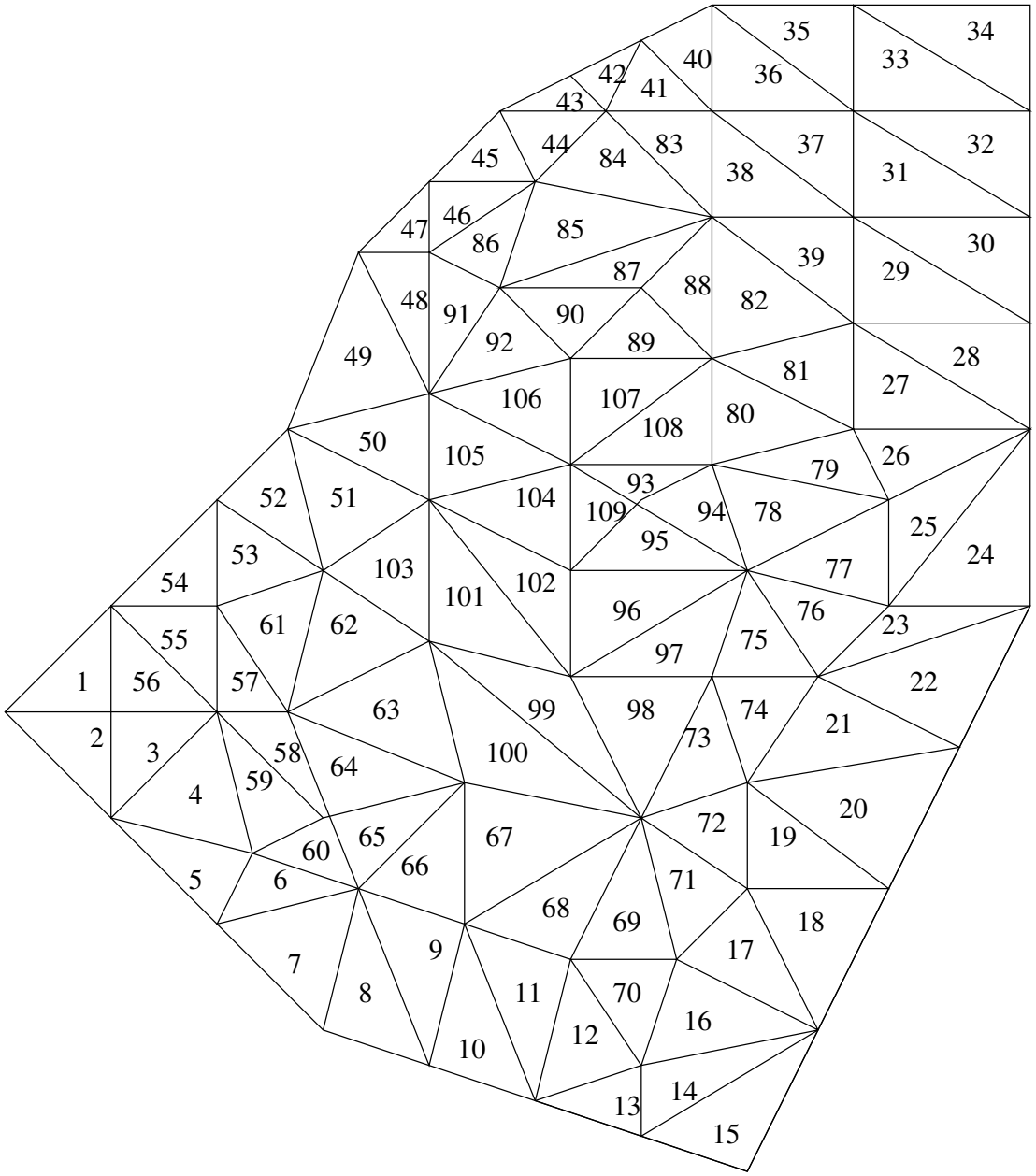

Figure 1: Finite-Element Netz

.