Finite-size Facility Placement in an Existing Layout Using MPI and C

Ketan Hemant Date

Class project for CSE 633 Parallel Algorithms (Fall 2012)
Instructor: Dr. Russ Miller

Nov 13, 2012

Outline

Introduction

- Facility Location Problem: A very popular and widely studied problem in Industrial Engineering.
- Objective is to locate new facilities in a plane, minimizing the distance between interacting facilities.

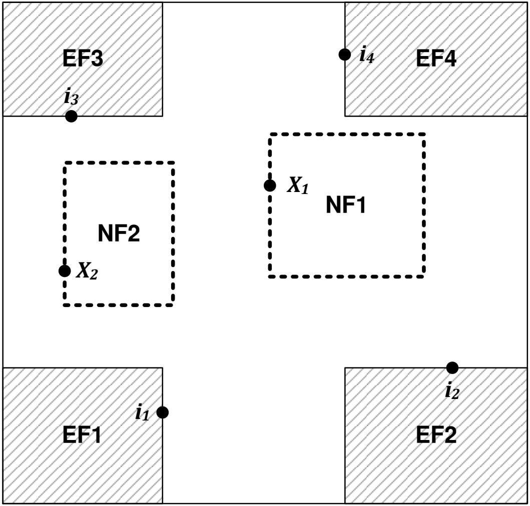
Types of objectives

- Median (or Minisum) objective.
- Center (or Minimax) objective.

Types of distance metrics

- Rectilinear (or L_1) metric $\Rightarrow |x_1 - x_2| + |y_1 - y_2|$.
- Euclidean (or L_2) metric $\Rightarrow \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Facility Placement Problem: An Example



Literature Review

Infinitesimal Facility Location (P-median Location)

The new facilities do not interact with each other.

- Location of p facilities in presence of infinitesimal facilities (Hakimi, 1964).
- Location of p facilities in presence of barriers (Larson and Sadiq, 1983).

Finite-size Facility Placement

- Placement of single arbitrarily shaped facility in presence of barriers (Savas et al., 2002).
- Placement of single rectangular GCR in presence of barriers (Sarkar et al., 2005).
- Placement of two rectangular, finite-size, interacting facilities in presence of barriers (Date and Nagi, 2012).
- Placement of single rectangular finite size NF with the help of dominance rules (Date et al., 2012)

Project Scope

1. Solving single, finite-size facility placement problem on parallel processors.
2. Solving single, finite-size facility placement problem using dominance rules on parallel processors.
3. Solving two, finite-size facility placement problem on parallel processors.

- For Fall 2012, focus will be on Item 1.
- Continue working on remaining problems over next semester.

Outline

Problem Description

Assumptions

- Layout: a rectangular, closed region with finite area.
- Finite number of *Existing Facilities* (EFs) with rectangular shapes.
- Need to locate single *New Facility* (NF) in the layout.
- Each EF has a single I/O point on boundary.
- NF has a single I/O point located at its top left corner.
- Non-negative material flow between EFs and NF; and pairs of EFs.
- Flow through any facility is not permitted.

Objective

To place NF optimally, minimizing the weighted sum of rectilinear distances between various interacting facilities.

Objective Function and Problem Statement

Notation

$\mathbf{p}$	:	Placement vector of the NF defined by coordinates of its top left corner
$u_i \geq 0$	:	Interaction between EF I/O point i and NF I/O point X
$w_{ij} \geq 0$	:	Interaction between EF I/O points i and j
$d_{\mathbf{p}}(i, X)$	:	Length of shortest feasible path between EF I/O point i and NF I/O point X
$d_{\mathbf{p}}(i, j)$	:	Length of shortest feasible path between EF I/O points i and j
$J(\mathbf{p})$	:	Total weighted travel distance between EFs and NF
$K(\mathbf{p})$	:	Total weighted travel distance between EFs

Objective Function

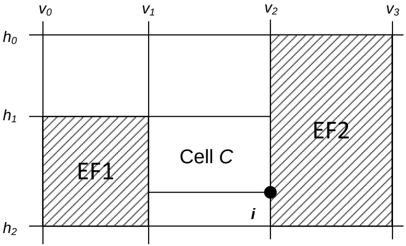
$$J(\mathbf{p}) + K(\mathbf{p}) = \sum_{i \in D} u_i d_{\mathbf{p}}(i, X) + \sum_{i \in D} \sum_{j \in D; j \neq i} w_{ij} d_{\mathbf{p}}(i, j)$$

Problem Statement

To determine optimal placement $\mathbf{p}^*$ of the NF such that:
$J(\mathbf{p}^*) + K(\mathbf{p}^*) \leq J(\mathbf{p}) + K(\mathbf{p}), \forall \mathbf{p} \in F$

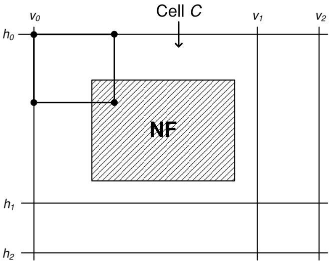
Outline

Grid Construction and Cell Formation



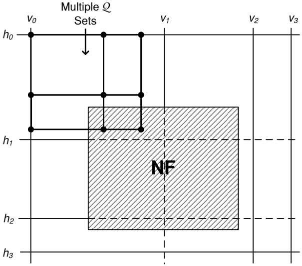
- Introduced by Larson and Sadiq (1983)
- Gridlines are constructed by passing a horizontal and vertical line through each vertex and I/O point of EFs.
- Flow between the facilities can be assumed to take place along the gridlines (without incurring any penalty).

Feasible Placement Candidates: Case 1



- NF does not cut off any existing gridlines.
- EF–EF flow not affected by NF placement.
- Optimal placement of the NF is such that one of its corners coincides with the cell corner (Sarkar et al., 2005).
- Upper bound on the number of all such candidates is $O(N^2)$.

Feasible Placement Candidates: Case 2



- NF cuts off some existing gridlines.
- EF–EF flow affected by NF placement.
- Need to construct $\mathcal{Q}$ sets for finding optimal placement candidates (Savas et al., 2002).
- Upper bound on the number of all such candidates is $O(N^4)$.

Outline

Solution Procedure

N = Number of EFs.

Step 1: Data Input and Problem Construction

- Input: Flat file containing coordinates of top left corners of EFs; dimensions of EFs; and coordinates of I/O points; facility interaction values.
- Different layouts are constructed into memory.

Step 2: Grid Construction

- Input: Coordinates of top left corners of EFs; dimensions of EFs; and coordinates of I/O points.
- Construction of horizontal and vertical gridlines passing through all EF vertices and I/O points.
- Algorithm complexity: $O(N)$.

Solution Procedure (cont.)

Step 3: Network Formation

- Input: Set of vertical and horizontal gridlines.
- Conversion of layout into network $G = (N, A)$.
- N: Set of nodes, i.e. gridline intersection points.
- A: Set of arcs, i.e. segments of horizontal or vertical gridlines.
- Algorithm complexity: $O(N^2)$.

Step 4: Cell Formation

- Input: Network $G = (N, A)$.
- Identification of various rectangular cells, which are objects bounded by four arcs.
- Algorithm complexity: $O(N^2)$.

Solution Procedure (cont.)

Step 5: Identification of Candidate Points

- Input: Network $G = (N, A)$ and set of cells $\mathcal{C}$.
- Identification of feasible placement candidates for the NF for different cells.
- Algorithm complexity: $O(N^4)$.

Step 6: Candidate Evaluation

- Input: Set of candidate points ($O(N^4)$); EF–EF interaction matrix; and EF–NF interaction vector.
- Evaluation of the objective function (sum of weighted distances) by placing NF at each candidate point.
- Finding the optimal placement(s) with the minimum overall objective function value.
- The network is reconstructed in $O(NLogN)$ time.
- Distances between different I/O points evaluated using Dijkstra's algorithm (in $O(N^3 LogN)$ time).
- Algorithm complexity: $O(N^7 LogN)$.

Why Parallelize?

- N is the number of EFs present in the layout.
- For single NF, $O(N^4)$ candidate points need to be evaluated. Complexity of overall procedure is $O(N^7 LogN)$.
- For two NFs, $O(N^8)$ feasible candidate pairs need to be evaluated. Complexity of overall procedure is $O(N^{11} LogN)$.
- As number of EFs goes on increasing, the sequential evaluation becomes cumbersome.
- Using parallel processing, each candidate can be evaluated separately and significant speedup can be achieved.

Outline

1. Introduction
2. Problem Description
3. Preliminaries
4. Solution Procedure
5. Implementation Strategy
6. Results
7. Conclusion and Future Work

Implementation using MPI and C

- Steps 1 to 4 are performed on all the MPI processes synchronously.
- In Step 1, the data is read from a flat file and layouts are constructed in the memory, as an input to the subsequent steps.
- Each process contains a local copy of the layout, grid structure, network and cell list.
- In Step 5, the cells are scattered among the processes for candidate identification (each process receives $\frac{O(N^2)}{n}$ cells).
- Individual processes identify the feasible candidate points, within the cells assigned to them.
- The partial candidate lists present at individual processes are gathered by the *root process* (rank 0) and a complete list is constructed.

Implementation using MPI and C (Cont.)

- In Step 6, the candidate list present at the *root process* is scattered among all the processes for evaluation (each process receives $\frac{O(N^4)}{n}$ candidate points).
- Individual processes calculate the objective function for all the candidate points assigned to them and identify the local minima.
- The local minimum at each process is gathered by the *root process* and the global minimum is identified, which gives the global optimal solution.

Execution Strategy

Problem Set

Computational study was conducted on randomly generated layout problems, with following specifications.

- **Data size:** 5 to 30 EFs incremented in steps of 5.
- **No. of problems per data size:** 100 (total 600 problems).
- **Layout congestion:** 30%.
- **EF area:** 10000 sq. units.
- **EF dimensions:** Randomly generated with aspect ratios $2^{U[-1,1]} = [0.5, 2]$.
- **Facility interactions:** Randomly generated from $U(0, 1)$.
- **NF dimensions:** 100×100 sq. units.
- NF I/O point located at its top-left corner.
- EF I/O point located randomly on its boundary.

Execution Strategy (Cont.)

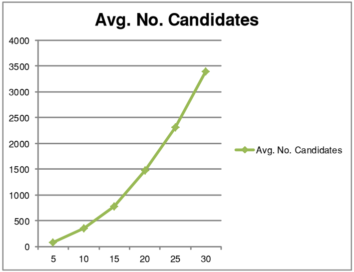
Hardware Specs.

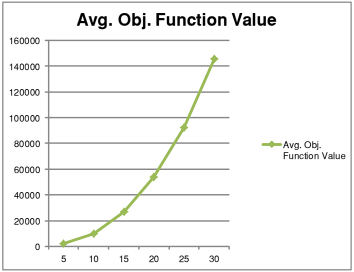
- **Number of processors:** 1 to 128, doubled at each step.
- **Type of processors:** GM Compute, 2-core nodes from CCR–U2 cluster.
- **Clock rate:** 3.00GHz.
- **Memory:** 2GB.
- **Communication network:** Myrinet.

Outline

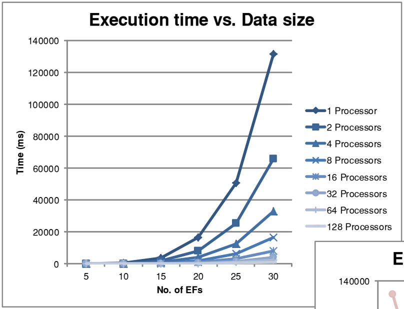
No. of Candidates/Obj. Function vs. Data Size

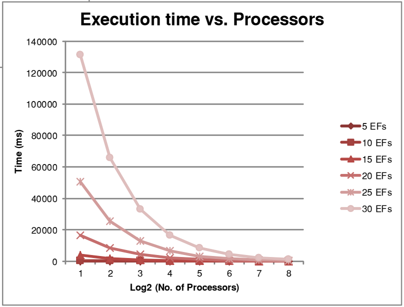
EFs	Avg. No. Candidates	Avg. Obj. Function
5	83.08	1883.70
10	346.69	9955.95
15	782.95	26798.76
20	1479.85	53637.44
25	2307.92	92519.14
30	3390.50	145949.02



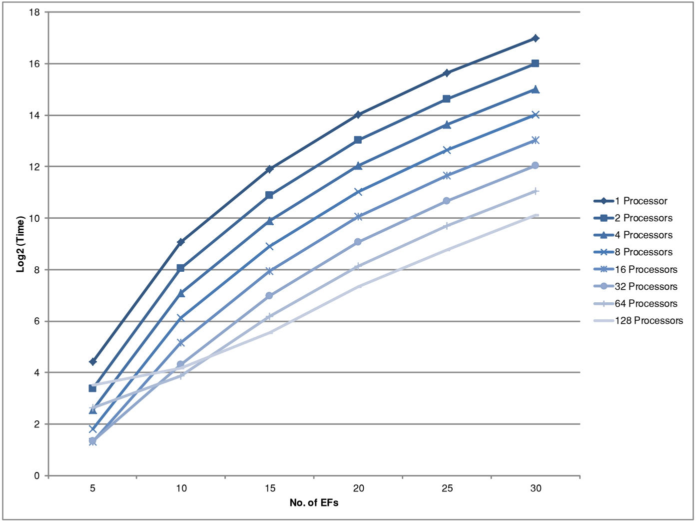


Execution Time Plots

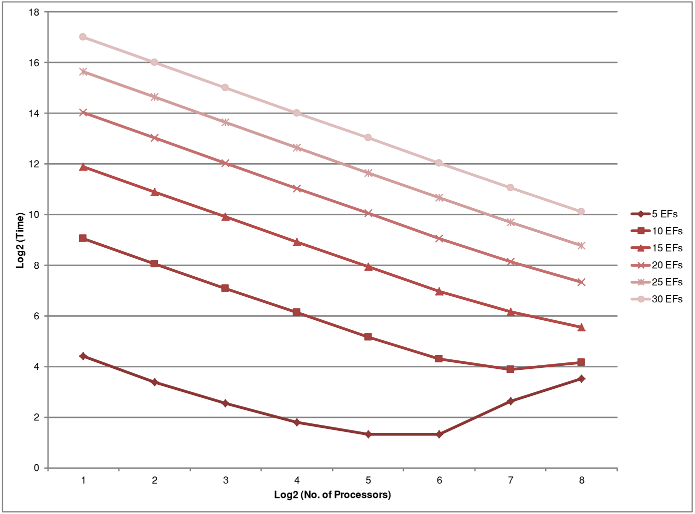




Execution Time vs. Data Size



Execution Time vs. Processors



Outline

Conclusion and Future Work

Conclusion

- Solved single facility placement problem in an existing layout, on multiple processors using MPI and C.
- Analyzed the execution time of the algorithm for various data sizes and number of processors.
- The execution time increases in polynomial order as the data size.
- Up to a fixed number of candidates per processor, execution time decreases by half as the number of processors is doubled, after which the communication time starts to dominate.
- From the graphs, the optimal number of candidates per processor is ≈ 4. The result is valid only for this particular implementation and hardware specifications.

Future Work

- Solving the one facility placement problem using dominance rules and comparing the results with parallel implementation.
- Solving the cumbersome two facility placement problem ($O(N^{11} LogN)$) on parallel processors.

Thank You

Date K., R. Nagi. 2012. Placement of two finite-size facilities in an existing layout with the rectilinear distance metric. *Submitted to Operations Research*.

Date K., S. Makked, R. Nagi. 2012. Dominance rules for the optimal placement of a finite-size facility in an existing layout. *Submitted to Computers & Operations Research*.

Hakimi S. L. 1964. Optimum locations of switching centers and the absolute centers and medians of graph. *Operations Research* **12** 450–459

Larson R. C., G. Sadiq. 1983. Facility locations with the Manhattan metric in the presence of barriers to travel. *Operations Research* **31**(4) 652–669.

Sarkar A., R. Batta, R. Nagi. 2005. Planar area location/layout problem in the presence of generalized congested regions with the rectilinear distance metric. *IIE Transactions* **37** 35–50.

Savas S., R. Batta, R. Nagi. 2002. Finite-size facility placement in the presence of barriers to rectilinear travel. *Operations Research* **50**(6) 1018–1031.