Nature-Inspired Computing

Example Problems

Dr. Şima Uyar
September 2006

Problem: Boolean Logic Minimization (1)

Definition: Assume you are given the set of all n prime implicants pi (i=1,2,...,n) (asal çarpan / temel içeren) for a boolean function f of k variables. You wish to find the circuit with the lowest cost implementing this function.

(Assume that the cost of the circuit corresponding to the minimized function depends only on the individual costs of the used prime implicants. The cost of each prime implicant pi is given as ci i=1,2,..,n).

Problem: Boolean Logic Minimization (2)

Example problem: Assume you have a boolean function of 4 variables f(x1, x2, x3, x4). The set of prime implicants is given below (xi' is the complement of xi).

p1: x3' x4

p2: x2' x4'

p3: x2' x3'

p4: x1' x3 x4'

p5: x1' x2 x3

p6: x1' x2 x4

Problem: Boolean Logic Minimization (3)

Hint: Think of how you would minimize the example boolean function given above f(x1, x2, x3, x4) using a prime implicants table based on the cost of each prime implicant (basit seçenekler tablosu). Assume that the cost of each prime implicant is calculated as follows: add 1 for each input variable and ½ for each complement, e.g. the costs of prime implicants p1 and p3 are 2.5 and 3 respectively).

Problem: Optimal Scheduling of Matrix Multiplications (1)

Given a sequence of n matrices M_i, i=1,2,...n with dimensions given as a one dimensional array D where each matrix M_i, i=1,2,...n has dimensions (D[i-1]*D[i]), find an optimal sequence of matrix multiplications which minimizes the total number of operations.

Problem: Optimal Scheduling of Matrix Multiplications (2)

Hint: Matrices can be multiplied by the standard formulas from linear algebra: if AxB=C then

$$C[ik] = \sum_j A[ij] * B[jk]$$

If A has dimensions (x*y) and B has dimensions (y*z) then the time for performing this multiplication in the straightforward way is O(x*y*z).

Problem: Optimal Scheduling of Matrix Multiplications (3)

Suppose there are more than two matrices to multiply. Matrix multiplication obeys the associative law A*(B*C)=(A*B)*C, so there is more than one order in which to do the multiplication. However the amount of time to do it may be very different.

Problem: Optimal Scheduling of Matrix Multiplications (4)

If for example A has dimensions (10*1000), B has dimensions (1000*4) and C has dimensions (4*200), then multiplying A*(B*C) would take roughly (10*1000*200+1000*4*200) operations, while multiplying as (A*B)*C would take roughly (10*1000*4+10*4*200) operations which is smaller than the first.

Problem: Diet Problem

This is a variation of the classical diet problem. The objective is to maximize Alice's satisfaction with her daily diet while adhering to the constraints on maximum daily intake amounts of specific nutritional ingredients and calories.

There are 20 different types of food. The amounts of each ingredient and the calorie amount for one serving of each food category is given. The maximum allowed amount for each nutritional item and maximum allowed calorie intake for Alice are also given. Alice was asked to grade each food type according to how much she likes them. Assuming that she can either eat one serving or none of each food type, recommend Alice her optimal diet.

Problem: Keyboard Layout

We wish to design an optimal layout (placement of keys on the keyboard) for a computer keyboard.

Problem: Map Coloring

Assume we have a map of N cities. We wish to color these cities using C colors in such a way that no neighbor cities have the same color. Find the assignment of colors to each city. (N and C are given constants.)

Problem: Fire Station Location

Let G = (V , E) be an undirected graph in which the vertices represent small towns and the edges represent roads between those towns. Each edge e has a positive integer weight d(e) associated with it, indicating the length of that road. The distance between two vertices (towns) in a graph is defined to be the length of the shortest weighted path between those two vertices. Each vertex v also has a positive integer c(v) associated with it, indicating the cost to build a fire station in that town. In addition, a positive integer parameter D is given. The objective is to minimize the total cost of building fire stations while ensuring that the distance from any town to a fire station does not exceed D.

Problem: Hiring Employees

An international company is hiring translators for its relations with the EU. Assume that a total of K translators applied for this job. In the application form, each applicant declares the languages he/she knows and the monthly salary he/she requires. The objective of the company is to hire translators such that the total monthly salary it pays to the translators is minimized while ensuring that all the languages spoken in the EU is known at least by one of the translators.

Problem: Query Optimization – Join Ordering (1)

The query optimizer is the component of a database management system that attempts to determine the most efficient way to execute a query. The optimizer considers the possible query plans for a given input query, and attempts to determine which of those plans will be the most efficient. Cost based query optimizers assign an estimated "cost" to each possible query plan, and choose the plan with the least cost. Costs are used to estimate the runtime cost of evaluating the query, in terms of the number of I/O operations required, the CPU requirements, and other factors.

Problem: Query Optimization – Join Ordering (2)

The set of query plans examined is formed by examining the possible access paths (e.g. index scan, sequential scan) and join algorithms (e.g. sort-merge join, hash join, nested loops). The search space can become quite large depending on the complexity of the SQL query.

Problem: Query Optimization – Join Ordering (3)

The performance of a query plan is determined largely by the order in which the tables are joined. For example, when joining 3 tables A, B, C of size 10 rows, 1000000 rows, and 1000000 rows, respectively, a query plan that joins B and C first can take several orders-of-magnitude more time to execute than one that joins A and C first.