

ПРИМЕНЕНИЕ ОПТИМИЗАЦИОННЫХ ПАКЕТОВ ПРИ РЕШЕНИИ ЗАДАЧ КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Бережнов Д. Е.

Егорова Н. Г. – к-т. техн. наук

На практике часто нужно решать оптимизационные задачи. При большом количестве выходных параметров и частых уточнениях моделей целесообразно использовать оптимизационные пакеты.

Рассматривается следующая задача:

$$\max \sum_{c=1}^n (a_c - b_c p_c)(p_c - h_c) - W \sum_{c=1}^n y_c, \quad (1)$$

$$0 \leq L_c \leq p_c \leq U_c + y_c, \quad (2)$$

$$a_c - b_c p_c \geq 0, \quad (3)$$

$$p_c \geq h_c, \quad (4)$$

$$\sum_{c \in M_j} (a_c - b_c p_c) \leq R_j, R_j \geq 0, M_j \subseteq M, \bigcup_{j=1}^k M_j = M, \quad (5)$$

$$p_{c_i,j} \leq p_{c_{i+1,j}}, p_{c_i,j} \in M_j, j = \overline{1, k}, i = \overline{1, |M_j| - 1}, \quad (6)$$

$$W > 0, a_c \geq 0, b_c \geq 0. \quad (7)$$

Нужно найти $p_c, c = \overline{1, n}$. Расчет исходных данных и анализ полученных результатов производился в среде Visual Studio на языке C#. Для получения решения оптимизационной задачи использовался пакет IBM ILOG CPLEX Optimization Studio, который отличается качеством получаемых решений и имеет интерфейс для программирования на C#. В оптимизационных пакетах задачи квадратичного программирования имеют следующий вид:

$$\text{Minimize } 1/2 x^T Qx + c^T x, \quad (8)$$

$$\text{subject to } Ax \sim b, \quad (9)$$

$$\text{with these bounds } l \leq x \leq u, \quad (10)$$

Выражение $\sum_{c=1}^n (a_c - b_c p_c)(p_c - h_c)$ несложно привести к виду (8) с помощью раскрытия скобок и

группировки. Неравенства (2)-(7) также несложно записать в форме (9)-(10). Для учета штрафа $W \sum_{c=1}^n y_c$ (штраф начисляется в случаях, когда $p_c > U_c$) предлагается множество отрезков $[L_c; U_c + y_c], c = \overline{1, n}$ разбить на два подмножества: $[L_c; U_c], c = \overline{1, n}$ и $[U_c; U_c + y_c], c = \overline{1, n}$. Получается следующая задача, которая решается в среде CPLEX Optimization Studio:

$$\max \sum_{c=1}^n (a_c - b_c (p'_c + y_c))((p'_c + y_c) - h_c) - W \sum_{c=1}^n y_c,$$

$$0 \leq L_c \leq p'_c + y_c \leq U_c + y_c,$$

$$a_c - b_c (p'_c + y_c) \geq 0,$$

$$p'_c + y_c \geq h_c,$$

$$\sum_{c \in M_j} (a_c - b_c (p'_c + y_c)) \leq R_j, R_j \geq 0, M_j \subseteq M, \bigcup_{j=1}^k M_j = M,$$

$$p'_{c_i,j} + y_{c_i,j} \leq p'_{c_{i+1,j}} + y_{c_{i+1,j}}, p_{c_i,j} \in M_j, j = \overline{1, k}, i = \overline{1, |M_j| - 1},$$

$$W > 0, a_c \geq 0, b_c \geq 0.$$

Решение такой модифицированной задачи будет состоять из наборов $p'_c, p'_c \in [L_c; U_c], c = \overline{1, n}$ и $y_c \geq 0, c = \overline{1, n}$, а решение исходной задачи будет иметь следующий вид: $p_c = p'_c + y_c, c = \overline{1, n}$.

В задачах квадратичного программирования оптимизационные пакеты без проблем вычисляют максимум только для выпуклых функций. В остальных случаях решения могут быть неоптимальными или вообще не вычисляться. Такая ситуация может возникнуть, если хотя бы для одного c_i выполняется нера-

венство $b_{c_i} < 0$ (для решаемой практической задачи такая ситуация может возникнуть в случае недостаточного количества данных для вычисления параметра b_c). В этом случае неравенство (2) будет иметь вид $0 \leq L_{c_i} - y_{c_i} \leq p_{c_i} \leq U_{c_i}$. Для решения такой задачи предлагается перейти к задаче поиска приближенного решения. Основная идея состоит в следующем. Пусть $y_i = x_i^2$. Без ограничения общности рассуждений полагаем $0 \leq x_i \leq 1$. Тогда $y_i \leq x_i$, $y_i \geq 0$, $y_i \geq 2x_i - 1$ (т.к. $(x_i - 1)^2 \geq 0$), т.е. вместо x_i^2 можно использовать параметр y_i и дополнительные ограничения (рисунок 1). Для получения более точного результата можно увеличить количество ограничивающих прямых.

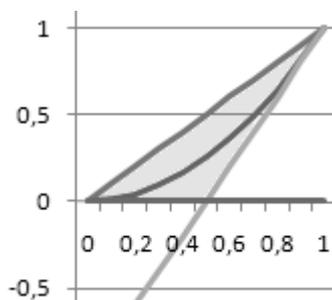


Рис. 1

Список использованных источников:

1. Химмельблау, Д. Прикладное нелинейное программирование. – М.: Мир, 1975.
2. IBM ILOG CP Optimizer V2.3. User's Manual. – 252 p.

ПОДХОД К ОПТИМИЗАЦИИ ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ В СИСТЕМАХ СБОРА СТАТИСТИКИ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Курдесов Н.Н.

*Поддубная О.Н. доцент,
кандидат физико-математических наук*

Вместе со стремительным накоплением информации развиваются и технологии анализа данных. Увеличивается количество областей, где работа с качественно большими объемами информации становится более актуальной. Постоянно происходит увеличение скорости потока данных в организационный процесс: экономика, банковская деятельность, маркетинг, веб-аналитика, и др. В современных системах хранения и обработки больших объемов информации одной из главных задач является обеспечение возможности быстрой выборки требуемых данных.

При выборе способа хранения большого объема информации необходимо учитывать специфику её выборки. Одной из черт в системах сбора статистики является то, что накапливаемые данные распределяются последовательно во времени, а выборка, как правило, делается по объектам сбора этих данных. Например: приложение собирает информацию о работе некоторого количества компьютеров в сети. Обработанные данные от каждого из компьютеров будут последовательно во времени записываться на диск. На данный момент для хранения больших объемов данных используются накопители на жестких магнитных дисках. Их особенностью является то, что скорость последовательного чтения/записи может быть в несколько раз выше, чем скорость случайного чтения/записи. При выборе информации конкретного объекта (в нашем случае статистики работы компьютера) время выборки будет сопоставимо с временем случайного чтения.

На рисунке 1 проиллюстрировано распределение данных на диске, накапливаемых последовательно во времени:

Данная проблема остаётся актуальной даже при использовании индексов в современных СУБД, вроде MySQL или Postgres. Они ускоряют поиск информации, расположенной на диске, но не могут ускорить её чтение.

Решением данной проблемы является подход перераспределения данных на диске таким образом, чтобы информация, которая в последствии будет использоваться для выборки, располагалась на нём последовательно.