

Esame di Programmazione su Architetture Parallele

Metodo del simplesso per la risoluzione della programmazione lineare

Belliato Riccardo (mat. 142652)

Simone Tomada

2022-08-02

Abstract

In questa relazione si propone una implementazione del metodo del simplesso a due fasi in CUDA per la risoluzione dei problemi di programmazione lineare in forma canonica.

Dopo una breve descrizione dell'algoritmo, seguirà la discussione su alcune scelte implementative.

Infine verranno valutate performance e scalabilità della soluzione proposta confrontando i tempi di esecuzione dell'algoritmo su istanze a dimensione crescente generate casualmente.

Contents

Introduzione al metodo del simplesso	1
Problemi di ottimizzazione lineare	1
Forma canonica e forma standard	2
Metodo del simplesso a due fasi	2
Scelte implementative e algoritmi utilizzati	2
Gestione della memoria	2
Ricerca del pivot e test di ottimalità	2
Eliminazione di Gauss	2
Aggiornamento della tabella	2
Risultati sperimentali	2

Introduzione al metodo del simplesso

Problemi di ottimizzazione lineare

Nell'ambito della Ricerca Operativa (Operations Research) uno dei principali argomenti è la cosiddetta **ottimizzazione lineare**, ossia lo studio di una classe di problemi del tipo:

$$\min / \max c^T x \text{ subject to } Ax \leq b$$

con $x \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^{1 \times m}$.

In altre parole, si vogliono trovare dei valori per le componenti del vettore x tali da massimizzare (o minimizzare) il valore di una *funzione lineare* (detta **funzione obiettivo**, mentre il vettore c è chiamato **vettore dei costi**), dati una serie di vincoli espressi nel sistema di disequazioni lineari $Ax \leq b$ (dove A è detta **matrice dei vincoli** e b **vettore dei termini noti**).

Questa classe di problemi permette di modellare un gran numero di situazioni reali in molteplici ambiti (ottimizzazione dei costi, creazione di orari, *etc.*), oltre che alcuni problemi NP-hard come il *Knapsack* o il *Vertex Cover*.

Dal punto di vista dell'algebra lineare, un problema in n variabili non è altro che uno spazio \mathbb{R}^n , la funzione obiettivo è una retta nello spazio, mentre i vincoli definiscono un *poliedro* nello spazio.

Utilizzando i teoremi e le tecniche dell'algebra lineare è stato possibile creare degli algoritmi per risolvere i problemi di ottimizzazione, come il **simpleso**, i quali si prestano molto bene ad essere parallelizzati (in quanto operano su matrici).

Problemi risolvibili, non risolvibili, illimitati

Dato un problema di ottimizzazione, questo può essere di tre tipi:

- avere una o più soluzioni ammissibili (*feasible*): ossia esistono uno o più vettori che moltiplicati per il vettore dei costi assegnano alla funzione obiettivo il valore massimo (o minimo possibile) e tutti i vincoli sono veri,
- non risolvibili (*infeasible*): se non esistono soluzioni
- illimitati (*unbounded*): se per una o più componenti della soluzione è possibile aumentarne (o ridurne) il valore all'infinito senza mai violare i vincoli

Forma canonica e forma standard

I problemi di massimizzazione possono essere convertiti in problemi di minimizzazione (e viceversa), così come è possibile manipolare le singole disequazioni. Queste operazioni servono a riportare i problemi in una forma che permetta di utilizzarli da parte dei solver. In particolare vengono utilizzate la forma *canonica* e la forma *standard*.

Un problema (di massimizzazione) è in forma canonica se è nella forma

$$\max c^T x \text{ subject to } Ax \leq b$$

Considerando che qualsiasi disequazione nella forma $\alpha x \leq y$ può essere convertita in una equazione equivalente $\alpha x + \delta = b$ definiamo la forma standard di un problema in forma canonica

$$\max d^T x \text{ subject to } A'x = b$$

con $A' = (A|I) \in \mathbb{R}^{m \times (n+m)}$ e $d = (c|0) \in \mathbb{R}^{n+m}$

In altre parole aggiungiamo una nuova variabile al problema (detta variabile *slack*) per ogni disequazione.

La forma standard è quella che viene utilizzata dagli algoritmi di soluzione.

Metodo del simpleso a due fasi

Scelte implementative e algoritmi utilizzati

Gestione della memoria

Estrazione delle colonne dalla matrice

Ricerca del pivot e test di ottimalità

Eliminazione di Gauss

Aggiornamento della tabella

Risultati sperimentali