ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Corso di Laurea in Ingegneria e Scienze Informatiche

Visualizzatore Grafico per Soluzioni di un Problema di Caricamento

Studente
Mattia Forti

Relatore
Prof. Marco Antonio Boschetti

Marzo 2025

Indice

Capitolo 1

Introduzione

Uno dei più famosi problemi di ottimizzazione conosciuti e studiati è il **Problema dello Zaino**, o **Knapsack Problem**.

Enunciato 1. *Sia dato uno zaino che possa sopportare un determinato peso e siano dati N oggetti, ognuno dei quali caratterizzato da un peso e un valore. Il problema si propone di scegliere quali di questi oggetti mettere nello zaino per ottenere il maggiore valore senza eccedere il peso sostenibile dallo zaino stesso.*

Il Knapsack Problem viene studiato da più di un secolo, con i primi studi risalenti al 1897. Si tratta di uno degli studi cardini del settore della Ricerca Operativa, in quanto è un problema molto comune e di grande interesse per diversi settori.

Tra le tecniche più utilizzate per la soluzione di questo problema troviamo:

- Programmazione Dinamica
- Algoritmo Greedy
- Branch and Bound
- Rilassamento Continuo

Essendo un problema strettamente di carattere matematico, questa tesi si pone l'obbiettivo di sviluppare un'applicazione open source che servirà non solo ad applicare gli algoritmi di risoluzione, ma anche a visualizzare i risultati con un'interfaccia grafica, per rendere più intuitiva la comprensione dell'argomento.

Capitolo 2

Descrizione del Problema

2.1 Definizione

2.2 Varianti

2.2.1 0/1

2.2.2 Frazionario

2.2.3 Con Limiti

2.2.4 Senza Limiti

2.2.5 Multi-Dimensionale

2.3 Complessità Computazionale

2.4 Soluzione del Problema

2.4.1 Programmazione Dinamica

2.4.2 Algoritmo Greedy

2.4.3 Branch And Bound

2.4.4 Rilassamento Continuo

Capitolo 3

Tecnologie

3.1 Tecnologie Proposte

3.1.1 OpenGL e GLUT

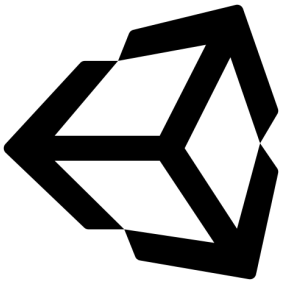
OpenGL (Open Graphics Library) è un'API molto popolare utilizzata per sviluppare applicazioni grafiche in 2D e 3D. È una libreria flessibile che permette una forte personalizzazione delle componenti grafiche.

Combinato con GLUT (OpenGL Utility Toolkit), si possono creare applicazioni partendo da 0 in C++, come insegnato nel corso di Computer Graphics di questo corso di laurea.

3.1.2 JavaSwing e JavaFX

JavaSwing e JavaFX sono delle librerie grafiche per Java, contengono un loop logico e delle componenti grafiche (pulsanti, pannelli, etc.) già implementate, permettendo un facile utilizzo delle applicazioni.

3.1.3 Unity



Una delle piattaforme più popolari per lo sviluppo di applicazioni e videogiochi, inoltre contiene un loop logico e delle componenti grafiche (pulsanti, pannelli, animazioni, etc.) già implementate. Tramite Unity è possibile creare delle interfacce grafiche estremamente fluide e *user-friendly*, grazie alla sua predisposizione per lo sviluppo di videociochi.

3.2 Tecnologie Scelte

Capitolo 4

Sviluppo dell'Applicazione

4.1 Requisiti per l'Interfaccia Utente

4.2 Setup dell'Ambiente di Sviluppo

4.2.1 Librerie

4.3 Componenti

4.3.1 Raccoglitore di Dati

4.3.2 Calcolatore

4.3.3 Motore Grafico

Capitolo 5

Testing dell'Applicazione

5.1 Dati Utilizzati

5.2 Risultati Matematici

5.3 Risultato dell'Interfaccia Grafica

Capitolo 6

Commenti

Capitolo 7

Conclusioni

Capitolo 8

Ringraziamenti

Capitolo 9

Bibliografia