****

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Documentazione**  Easy Pass   |  |  | | --- | --- | |  |  | | Versione | 1.0 | | Data | 14/02/2022 | | Destinatario | Prof. F. Palomba | | Presentato da | Montefusco Alberto  Rinaldi Viviana  Spina Gennaro | |  |  | |

Sommario

[**Sommario** 2](#_Toc530825397)

[**Team members**](#_Toc530825396) 3

[**Repository GitHub**](#_Toc530825396) 3

1. [**Introduzione**](#_Toc530825398) 4

1.1 Sistema corrente 5

1.2 Sistema proposto 5

1. [**Definizione del problema**](#_Toc530825398) 6

2.1 Obiettivi 6

2.2 Specifica PEAS 6

2.2.1 Caratteristiche dell'ambiente 6

2.3 Analisi del problema ............................................................................................................................ 7

**3.**   **Algoritmo Genetico** ...................................................................................................................................... 8

3.1 Parametri …........................................................................................................................................... 9

3.1.1 Encoding degli individui........................................................................................................ 10

3.1.2 Crossover ............................................................................................................................... 11

3.1.3 Mutazione ............................................................................................................................. 16

3.1.4 Funzione di fitness ............................................................................................................... 23

3.1.5 NSGAII multi-obiettivo ...................................................................................................... 33

3.1.6 Criteri di arresto .................................................................................................................... 38

3.1.7 Passi dell'algoritmo ............................................................................................................... 47

**4.** **Testing dei parametri scelti** ....................................................................................................................... 9

4.1 Numero di iterazioni ............................................................................................................................ 9

4.2 Durata delle iterazioni ........................................................................................................................ 10

4.3 Size della popolazione ....................................................................................................................... 15

4.4 Dimensione dell'individuo ................................................................................................................ 15

**5. Glossario** ....................................................................................................................................................... 56

Team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Ruolo** | **Acronimo** | **Informazioni di contatto** |
| Montefusco Alberto | Team member | MA | a.montefusco28@studenti.unisa.it |
| Viviana Rinaldi | Team member | VR | v.rinaldi26@studenti.unisa.it |
| Spina Gennaro | Team member | SG | g.spina5@studenti.unisa.it |

Repository GitHub

|  |
| --- |
| **Link** |
| https://github.com/Alberto-00/EasyPass-AI |

**CAPITOLO 1**

|  |
| --- |
| **Introduzione** |

* 1. Sistema corrente

Per applicare le regole anti-Covid in ambito universitario, imposte dal Governo a seguito dell’emergenza sanitaria, è previsto un meccanismo di controllo del Green Pass degli Studenti presenti in aula, effettuato da parte dei Docenti che sostengono la lezione.

Attualmente, il Docente è tenuto a scegliere a campione un numero di Studenti pari al 20% del numero dei presenti e, tramite l’applicazione “Verifica C19” installata sul proprio dispositivo, distribuita e approvata dal Ministero della Salute, scannerizzare i Green Pass di tali Studenti.

La procedura prevista è una procedura “manuale”, nel senso che il Docente deve spostarsi fra i banchi per raggiungere lo Studente selezionato. Inoltre, al termine delle operazioni, il Docente è tenuto a registrare alcuni dati riguardanti l’andamento del controllo appena effettuato (ad esempio, il numero di Green Pass esaminati) e riportare tale informazione al Direttore del Dipartimento a cui appartiene.

Sebbene la validazione di per sé è attuata dall’applicazione, la mancata informatizzazione di tutto l’iter di controllo comporta alcuni disagi a danno degli stessi Docenti e Studenti, sia da un punto di vista di risorse, quindi del tempo sottratto alla lezione, ma anche dal punto di vista della sicurezza, poiché diventa difficile garantire il mantenimento della distanza di sicurezza di un metro nel momento in cui il Docente deve effettuare la scansione del certificato dello Studente.

Inoltre, nella classe sono presenti posti contrassegnati che lo studente può occupare in modo tale da rispettare il distanziamento però questo non permette sempre di avere una buona visione della lavagna o del proiettore.

* 1. Sistema proposto

Per eliminare i disagi esposti nel precedente paragrafo, Easy Pass viene sviluppato come una Web Application, accessibile da Internet, mirata all’informatizzazione della procedura stessa. Tramite Easy Pass sarà possibile effettuare le dovute validazioni senza che ciò richieda un elevato impiego di tempo prezioso per la lezione e rispettando a pieno le misure di sicurezza per il distanziamento interpersonale. Il Docente potrà adempiere al suo ruolo di vigilante utilizzando soltanto gli strumenti messi a disposizione dall’Università, quali un computer con cui accedere al sito e un proiettore per condividere lo schermo con la classe.

Il Sistema è basato sull’utilizzo di sessioni di validazione identificate da un codice QR che, una volta condiviso dal Docente (tramite il proiettore) e scansionato dagli Studenti, permetterà a questi ultimi di inserire il proprio Green Pass per sottoporlo alla verifica. La procedura, pertanto, sarà a carico di Easy Pass, che effettuerà i controlli sui certificati ricevuti e invierà un report della sessione di controllo direttamente al Direttore di Dipartimento.

Easy Pass prevede anche delle funzionalità dedicate al Direttore di Dipartimento, in merito alla gestione di tutti i report risultanti dalle sessioni di validazione effettuate nel Dipartimento di cui è a capo.

Inoltre, il Sistema prevede anche un modulo di intelligenza artificiale che permette di selezionare i posti, che gli studenti potranno occupare, in modo da rispettare le norme di sicurezza imposte dal Ministero della Salute ma anche di posizionare lo studente in un posto ritenuto migliore per la visualizzazione della lavagna (o del proiettore). Tramite questo modulo intelligente, il Docente, dopo che si è autenticato con le sue credenziali, ha inserito il numero di studenti presenti quel giorno e l’aula in cui si terrà la lezione, mostrerà alla classe una mappa contenente i posti migliori e sicuri (in termini di norme anti Covid-19).

**CAPITOLO 2**

|  |
| --- |
| **Definizione del problema** |

* 1. Obiettivi

Lo scopo del progetto è quello di realizzare un agente intelligente che sia in grado di:

* disporre gli studenti in aula in modo da rispettare il metro di distanza tra due individui, imposto dalle norme sulla sicurezza del Ministero della Salute, e di scegliere un posto ritenuto migliore per la visualizzazione della lavagna (o del proiettore).

Una volta realizzato l’agente intelligente, quest’ultimo dovrà essere integrato all’interno di Easy Pass.

**Nota**: attenendoci alle norme Universitarie, il metro di distanza sarà rispettato se lo studente dista da un altro di almeno un posto.

* 1. Specifica PEAS

|  |  |
| --- | --- |
| ***PEAS*** | |
| **Performance** | La misura di **performance** dell’agente si basa sull’accuratezza di assegnare un posto ad uno studente che non abbia intorno, di un posto, un altro studente e che sia posizionato nel miglior posto per vedere la lavagna o proiettore. |
| **Enviroment** | L’**ambiente** dell’agente è formato dall’aula, i singoli posti e gli studenti. |
| **Actuators** | Gli **attuatori** consistono nel mostrare a video le disposizioni consigliate degli N studenti. |
| **Sensors** | I **sensori** tramite il quale l’agente reperisce gli stimoli dall’ambiente sono rappresentati dai bottoni per richiedere la visualizzazione della *seating* *map* e per la configurazione di quest’ultima tramite l’inserimento della dimensione dell’aula e del numero di studenti. |

* + 1. Caratteristiche dell’ambiente

L’ambiente è:

* **Completamente osservabile**: l’agente ha sempre accesso a tutte le informazioni relative alla disposizione dei posti.
* **Deterministico**: lo stato successivo dell’ambiente è completamente determinato dallo stato corrente e dall’azione eseguita dall’agente;
* **Sequenziale**: la posizione che lo studente può occupare è influenzata dalla sequenza di posti occupati precedentemente.
* **Statico**: durante la ricerca dei posti, il Docente attende l’esecuzione dell’algoritmo.
* **Discreto**: l’insieme delle possibili percezioni e azioni dell’agente sono distinte e definite.
* **Agente singolo**: l’ambiente consente la presenza di un unico agente.
  1. Analisi del problema

**Scopo**. Disporre N studenti in un’aula in modo tale che nessuno possa avere intorno (almeno di un posto) uno studente e che tali studenti siano disposti in modo da avere un’ottima visuale per la lavagna (o proiettore).

**Formulazione --- del problema**:

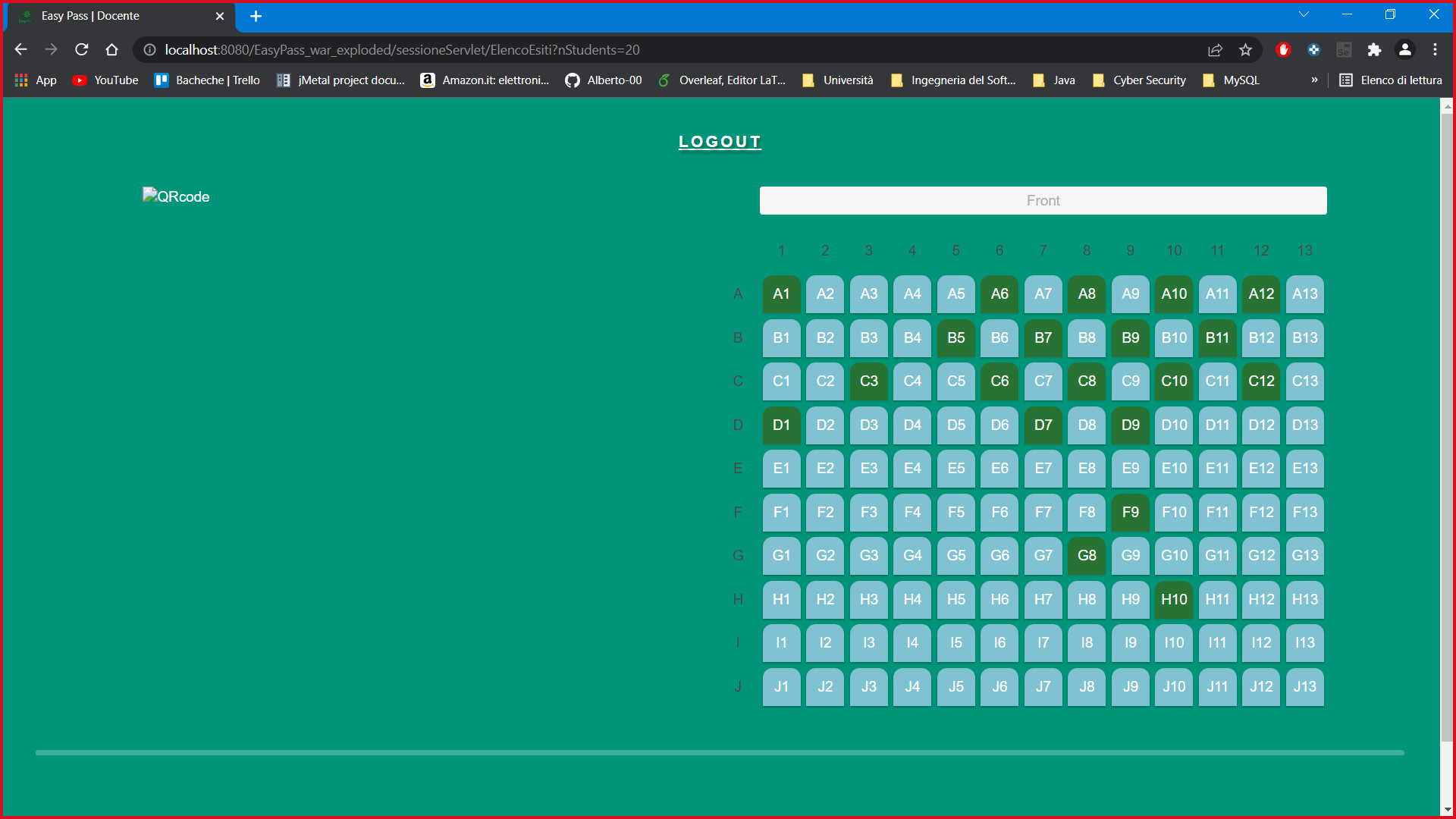
* **Stati**. Ogni disposizione nell’aula degli N studenti è uno stato.
* **Stato iniziale**. Aula vuota senza posti assegnati.
* **Azioni**. Aggiungere una regina in una casella vuota.
* **Modello di transizione**. Restituisce l’aula con una nuova disposizioni di studenti.
* **Test obiettivo**. Sulla scacchiera ci sono N studenti con nessun conflitto (spiega vivi….).

**Strategia**. Formulato il problema, era chiara la necessità di utilizzare un algoritmo di ottimizzazione. In particolare, abbiamo implementato un algoritmo genetico multi-obiettivo in quanto le

sue caratteristiche sono simili a quelle del problema. In particolare, è stata utilizzata una soluzione NSGA poiché tiene conto del trade-off tra i vari obiettivi contrastanti e, inoltre, creare un insieme di soluzioni non dominate favorendo gli individui più vicini al vero Fronte di Pareto.

**Esempio**:

* Aula 13 x 10;
* 20 studenti da disporre.



**CAPITOLO 3**

|  |
| --- |
| **Algoritmo Genetico** |

* 1. Parametri

Durante la progettazione dell’algoritmo genetico si è cercato un modo per stabilire la bontà di una

configurazione di parametri. Di seguito vengono elencati i parametri utilizzati per il setup dell’algoritmo genetico multi-obiettivo.

* + 1. Encoding degli individui

Un individuo è una disposizione di N studenti in un’aula di m x n posti (dove m rappresenta il numero di righe e n il numero di colonne), mentre un gene è inteso come le coordinate (x e y) di un posto per uno studente.

La codifica utilizzata è una codifica reale in quanto più compatta e più vicina al problema in questione (perché prende degli input numerici).

La soluzione ottima è composta da un individuo che avrà il numero minimo di conflitti e il numero massimo della funzione di fitness. Un conflitto si ha nel momento in cui un individuo (nel nostro caso lo studente), situato in un certo posto della matrice, ha adiacente un altro individuo.

La funzione di fitness calcola il punteggio della disposizione sulla base dei posti occupati.

* + 1. Crossover

Il crossover

* + 1. Mutazione
    2. Calcolo dei conflitti
    3. Funzione di fitness
    4. NSGAII multi-obiettivo
    5. Criteri di arresto
    6. Passi dell’algoritmo

**CAPITOLO 4**

|  |
| --- |
| **Testing dei parametri scelti** |

* 1. Numero di iterazioni
  2. Durata delle iterazioni
  3. Size della popolazione
  4. Dimensione dell’individuo

**CAPITOLO 5**

|  |
| --- |
| **Glossario** |