

# Pokémon

TM

GottaBattleEmAll

GA



PRESS START

© 2004 GAME FREAK Inc.



Questo progetto si concentra sull'utilizzo di algoritmi di intelligenza artificiale per prevedere l'esito degli scontri tra due squadre Pokémon.



I Pokémon rappresentano un vasto universo affascinante, dove l'impiego di strumenti di intelligenza artificiale apre nuove prospettive nell'analisi delle interazioni tra queste creature.



Attraverso l'analisi delle caratteristiche e delle abilità dei Pokémon, cerchiamo di sviluppare un modello predittivo per determinare il vincitore di un confronto.



Luigi

HP

Lu. 0512114301

30/30



Angelo

HP

Lu. 0512113929

30/30



Gianvincenzo

HP

Lu. 0512114922

30/30



Team Member



NEXT



**Una delle principali sfide è decidere quale approccio adottare per risolvere il problema. Due approcci comuni sono la ricerca locale e l'apprendimento.**



Chi sceglierà tra Ricerca Locale, Bulbasaur e Apprendimento?

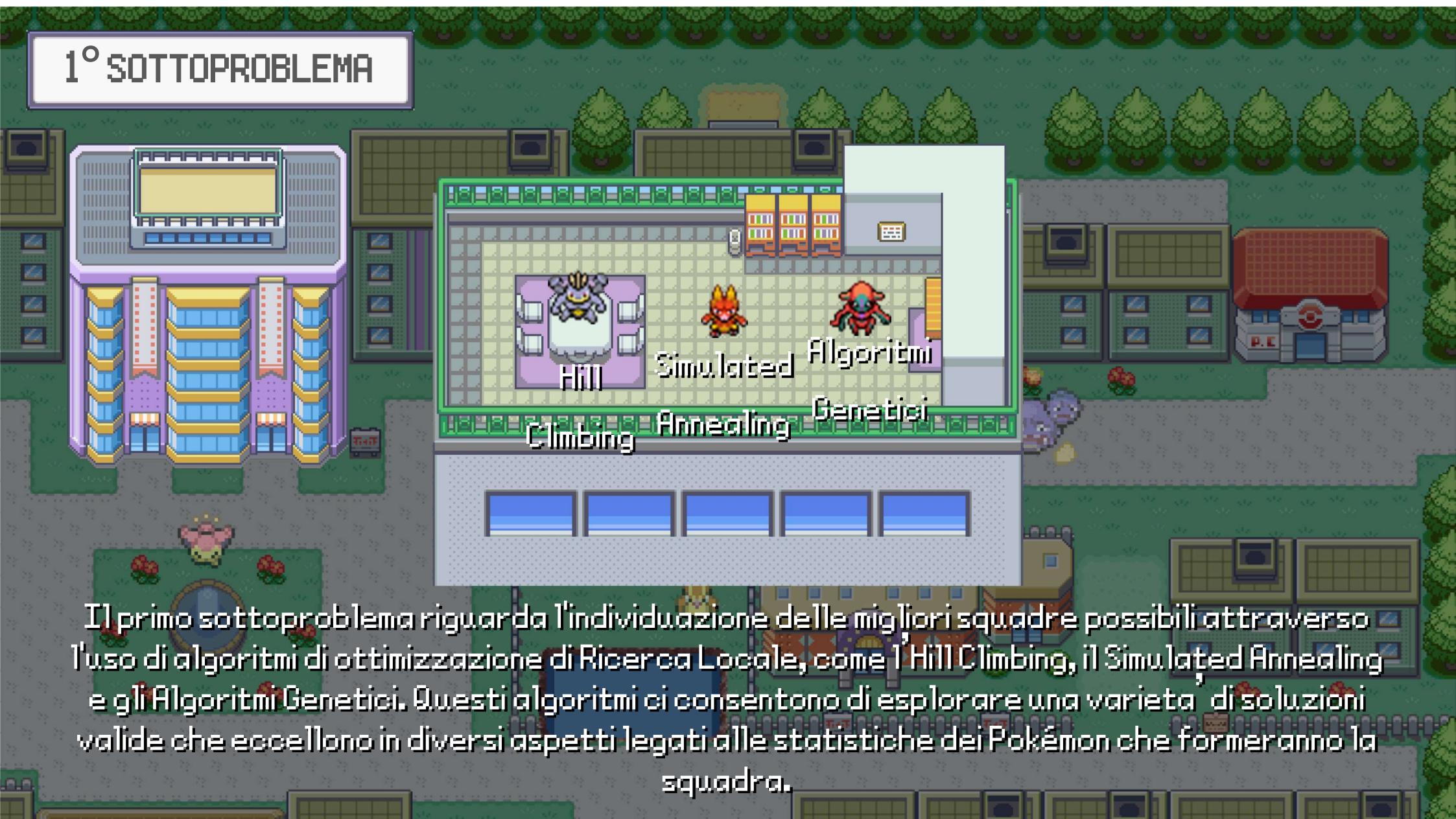


Complimenti hai scelto Ricerca Locale, nessuno sceglie Bulbasaur.

# PROBLEMA



## 1° SOTTOPROBLEMA



Il primo sottoproblema riguarda l'individuazione delle migliori squadre possibili attraverso l'uso di algoritmi di ottimizzazione di Ricerca Locale, come l'Hill Climbing, il Simulated Annealing e gli Algoritmi Genetici. Questi algoritmi ci consentono di esplorare una varietà di soluzioni valide che eccellono in diversi aspetti legati alle statistiche dei Pokémon che formeranno la squadra.

## PKH DATA



L'Hill Climbing è un algoritmo di ottimizzazione locale che esplora lo spazio delle soluzioni cercando iterativamente una soluzione migliore adiacente alla soluzione corrente. Sebbene sia efficiente e facile da implementare, l'Hill Climbing può rimanere bloccato in massimi locali e non garantisce la ricerca della soluzione ottimale.



## LEMA



## PKMN DATA



Simulated Annealing è una tecnica di ottimizzazione probabilistica ispirata al processo di ricottura dei metalli, che permette di esplorare lo spazio delle soluzioni in modo più ampio, accettando occasionalmente soluzioni peggiori per evitare minimi locali. Tuttavia, richiede la messa a punto di parametri sensibili, come la temperatura di Annealing.



# PKMN DATA

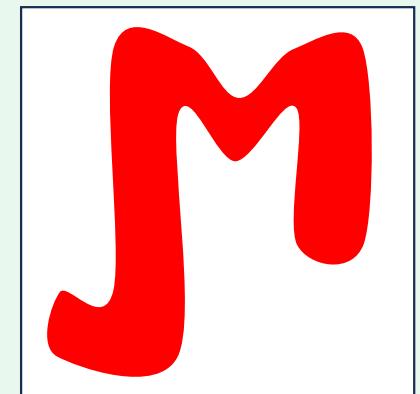


L'Algoritmo Genetico è una tecnica di ottimizzazione basata su popolazioni ispirata alla teoria dell'evoluzione biologica, che utilizza concetti come la selezione naturale, la mutazione e l'incrocio per esplorare lo spazio delle soluzioni. È particolarmente adatto a problemi con molteplici obiettivi, come la ricerca delle migliori squadre Pokémon.



## 2° SOTTOPROBLEMA





Abbiamo scelto JMetal per la sua flessibilità e la vasta gamma di funzionalità offerte per l'implementazione di algoritmi evolutivi. Inoltre, abbiamo integrato nel nostro progetto due dataset completi contenenti tutti i dati relativi ai Pokéモンster, strutturato in formato CSV. Un dataset contiene i Pokéモンster mentre l'altro contiene le mosse

# Progettazione



# Progettazione



**CODIFICA DEGLI INDIVIDUI**

Per rappresentare gli individui nel nostro algoritmo genetico, adotteremo un approccio basato su un array di interi. Ogni elemento di questo array corrisponderà alla chiave di un Pokémon all'interno di un'HashMap pre-caricata che contiene tutti i Pokémon del dataset.

# Progettazione



# Progettazione



## INIZIALIZZAZIONE

Il secondo approccio impiega un metodo di inizializzazione basato su euristiche. Qui, sfrutteremo conoscenze specifiche del dominio del problema per generare una popolazione iniziale che rifletta le caratteristiche desiderate delle soluzioni.

# Progettazione



# Progettazione



## FUNZIONI DI FITNESS

Adotteremo una funzione di fitness multiobiettivo, che considerera' simultaneamente piu' criteri di valutazione. Questo ci consentira' di valutare gli individui in base a piu' obiettivi (funzioni) e di specificare dei pesi per ognuno di essi. Le nostre quattro funzioni di fitness saranno:

# Progettazione



## FUNZIONI DI FITNESS

1. Media del Totale di un Team
2. Copertura del tipo Pokemon
3. Copertura del tipo di Mossa
4. Statistiche delle mosse

# Progettazione



## FUNZIONI DI FITNESS

Formula funzione fitness:

$$\text{Fitness} = w_1 \cdot f_1 + w_2 \cdot f_2 + \dots + w_i \cdot f_i$$

La normalizzazione è necessaria per garantire che gli obiettivi con diverse scale o range di valori abbiano un impatto equilibrato valutazione complessiva della fitness.

Min-max normalization:  $\hat{y} = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$

# Progettazione



La fase di selezione in un algoritmo genetico svolge un ruolo fondamentale nel determinare quali individui della popolazione saranno inclusi nel mating poll, i quali avranno la possibilità di riprodursi e generare la prossima generazione di individui. Per la nostra implementazione, esploreremo due strategie di selezione diverse: K-Tournament e RandomSelection.

# Progettazione



## SELEZIONE

Dopo vari test abbiamo scelto K-Tournament perché genera soluzioni migliori.

# Progettazione



# Progettazione



## Crossover

Dopo vari test abbiamo scelto K-Point perché restituisce soluzioni migliori.

# Progettazione



## MUTAZIONE

La mutazione e' un altro operatore genetico essenziale all'interno degli algoritmi genetici, il cui compito e' introdurre variazioni casuali nei genotipi degli individui della popolazione. Una delle strategie di mutazione comuni e' il Random Resetting, in cui un gene casuale all'interno del cromosoma di un individuo viene selezionato casualmente e sostituito con un valore casuale. Verra' sperimentata la probabilita' di mutazione ideale.

# Progettazione



## STOPPING CONDITION

La stopping condition, o condizione di terminazione, determina quando l'esecuzione dell'algoritmo deve essere interrotta. Serve a garantire che l'algoritmo genetico abbia sufficiente tempo per esplorare e sfruttare efficacemente lo spazio delle soluzioni, evitando allo stesso tempo di continuare indefinitamente l'ottimizzazione senza ottenere miglioramenti significativi. La stopping condition verrà implementata impostando un numero massimo di iterazioni. Tale numero verrà definito empiricamente.



SPEAR2

Lu 3

HP

NSGR-II

Lu 6

HP

30/30

EXP

Chi vincera'?

Crowding Distance

Elitismo

Mating Pool

Fronte di Pareto



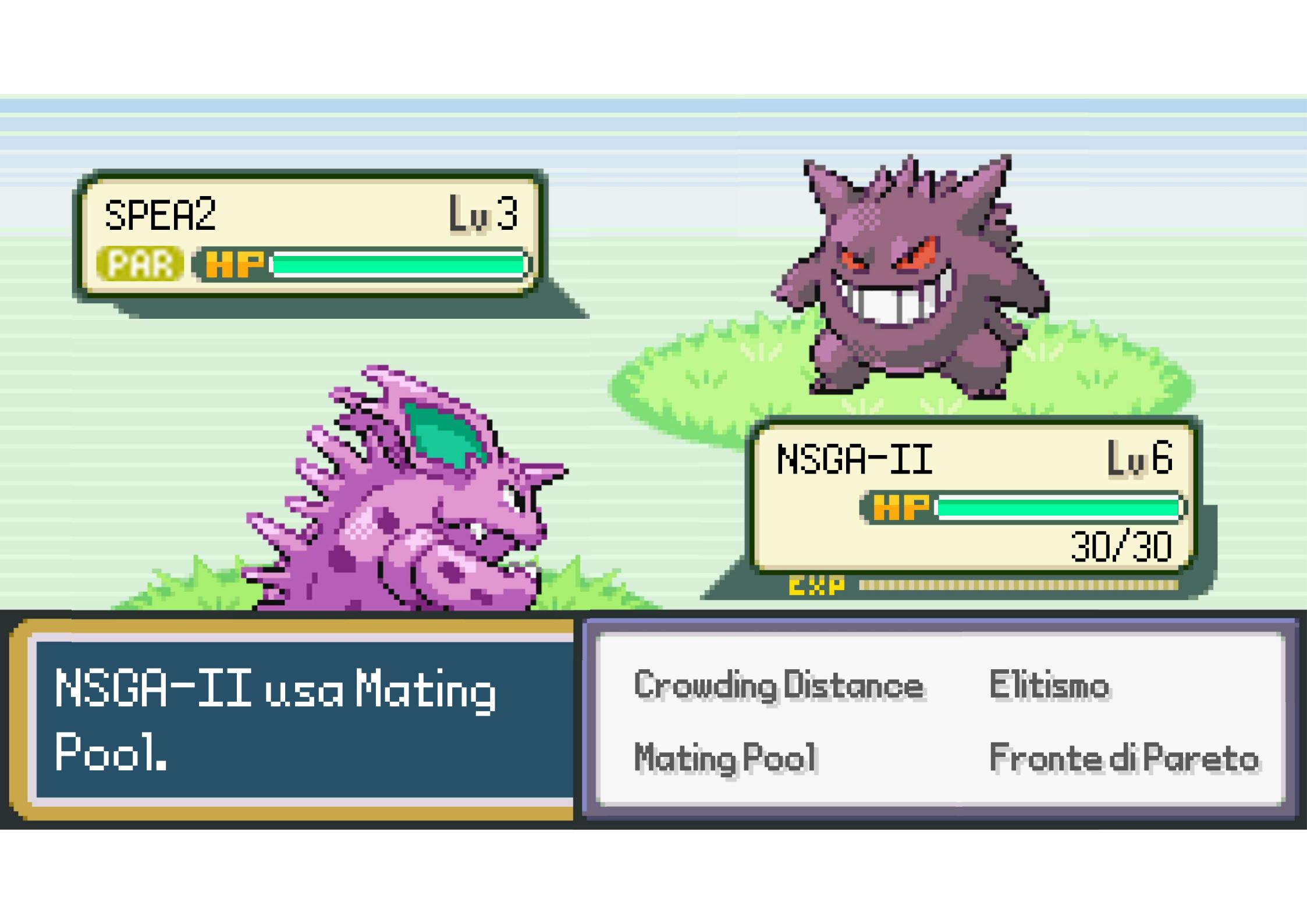
NSGA-II usa Elitismo.

L'NSGA-II è un algoritmo **elitista**, il che significa che i cromosomi migliori di una generazione saranno presenti anche nella generazione successiva.



SPEA2 e' paralizzato.

L'NSGA-II e' un algoritmo **elitista**, il che significa che i cromosomi migliori di una generazione saranno presenti anche nella generazione successiva.



SPEA2

Lv 3

PAR

HP



Lv 6

HP

30/30

EXP

NSGA-II usa Mating  
Pool.

Crowding Distance  
Mating Pool

Elitismo  
Fronte di Pareto



NSGA-II usa Mating Pool.

Per garantire l'elitismo, ogni volta che si forma il mating pool, vengono selezionati gli individui con il rango migliore, ovvero il rango più basso, e con un alto valore di crowding-distance. Questo assicura che i migliori individui vengano conservati e trasferiti alla generazione successiva, contribuendo così alla convergenza verso soluzioni di alta qualità nel tempo.



Brutto Colpo.

Per garantire l'elitismo, ogni volta che si forma il mating pool, vengono selezionati gli individui con il rango migliore, ovvero il rango più basso, e con un alto valore di crowding-distance. Questo assicura che i migliori individui vengano conservati e trasferiti alla generazione successiva, contribuendo così alla convergenza verso soluzioni di alta qualità nel tempo.

SPEA2

Lv 3

PAR

HP



NSGA-II

Lv 6

HP

30/30

EXP

NSGA-II usa Crowding Distance.

Crowding Distance

Elitismo

Mating Pool

Fronte di Pareto



NSGA-II usa Crowding Distance.

Esso utilizza l'ordinamento non-dominato per mantenere la diversità delle soluzioni e il **crowding distance** per favorire la distribuzione uniforme delle soluzioni sull'intero fronte di Pareto.



Superefficace.

Esso utilizza l'ordinamento non-dominato per mantenere la diversità delle soluzioni e il **crowding distance** per favorire la distribuzione uniforme delle soluzioni sull'intero fronte di Pareto.

SPEAR2

Lv 3

PAR

HP



NSGA-II

Lv 6

HP

30/30

EXP

NSGA-II usa Fronte  
di Pareto.

Crowding Distance

Elitismo

Mating Pool

Fronte di Pareto

SPEAR2

Lu 3

PAR

HP



NSGA-II

Lu 6

HP

50/50

EXP

NSGA-II usa Fronte  
di Pareto.

Gestisce meglio il fronte di pareto.

SPEAR2

Lu 3

PAR

HP



NSGRA-II

Lu 6

HP

30/30

EXP

Superefficace.

Gestisce meglio il fronte di pareto.

SPEAR2

Lv 3

HP



NSGA-II

Lv 6

HP

30/30

EXP

NSGA-II Vince.



Fine

