

LAB 10 – ES 01

Strutture dati

Per memorizzare l'elenco dei Target e delle Risorse si usano dei vettori di struct con i relativi campi coerenti con le caratteristiche di entrambi i tipi.

Per memorizzare le posizioni in cui si vanno a piazzare le risorse si fa uso di un vettore di dimensione NR, dove NR è il numero di righe della matrice che rappresenta la mappa, dove per ogni riga si indica se è presente una risorsa e, se sì, in quale colonna (se non ci sono risorse si ha pos[i]=-1).

Per rappresentare i target sulla mappa ci si appoggia a due matrici, una per i nemici e una per gli alleati, dove ogni casella è inizializzata a 0 e in cui si assegna, per ogni target in posizione [i,j] , un numero negativo il cui valore assoluto corrisponde al valore del target (di difesa o di attacco a seconda se è un alleato o un nemico) per gestire meglio il controllo sulla validità della soluzione.

Strategie algoritmiche

C'è una lettura da File degli elenchi di Target (*leggiFileTarget*) e di Risorse (*leggiFileRisorse*) e una lettura da riga di comando delle dimensioni della matrice e del budget.

Dato il budget, e considerando che bisogna creare dei sottoinsiemi di risorse coerenti con esso ma che possano prevedere ripetizioni, si calcola per ogni risorsa quante occorrenze ci sarebbero se il budget venisse usato solo per comprare quella risorsa ($N = \text{budget} / \text{costo}$, Funzione *calcolaOccorrenze*). Di questi valori si trova il massimo *max*, e ciò permette di calcolare le combinazioni con ripetizione di n risorse distinte di classe k, con k crescente da 1 a max (funzione *calcolaElenchi*). In *comb_rip* viene applicata una condizione di pruning sulla prossima scelta: se questa comporta il superamento del budget, non viene presa.

Appena trovato un elenco di risorse che rispetta il budget, si prova a piazzarle sulla mappa (funzione *provaElenco*) applicando le disposizioni semplici di NR righe di classe t, dove t è la dimensione dell'elenco di risorse attuale, e salvando nel vettore *posizioni[]* di dimensione NR se nella riga i c'è una risorsa presente ed eventualmente in quale colonna. In *disp_semplici* si applica una condizione di pruning sulla scelta della posizione: se questa è già occupata da un target, si passa avanti.

Una volta creato un "elenco" di posizioni, si richiama *controllaSol* e si provano ad applicare le proprietà delle risorse sulle due matrici ausiliarie per i due tipi di target (nemici ed alleati, inizializzate ognuna con una chiamata di *inizializzaMappa*) e si somma, ad ogni casella compresa nella regione d'azione della risorsa, il valore di difesa o di attacco a seconda che la matrice si riferisca a alleati o nemici. Se tutte le caselle delle due matrici sono positive, allora l'elenco di risorse trovato in *comb_rip*, ognuna piazzata nella posizione individuata in *disp_semplici*, va bene e si stampa la soluzione mediante la funzione *stampaSol* e, aggiornando il flag *finito*, si "interrompono" le altre ricorsioni.