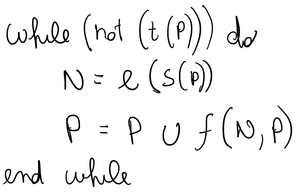
Pool Evolution Pattern

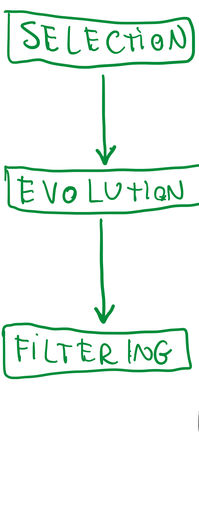
venerdì 6 settembre 2019 16:05

Il pattern pool evolution modella l'evoluzione di una popolazione soggetta a mutazioni, questa popolazione evolve in modo che una certa funzione fitness sia ottimizzata.
Il pattern può essere utilizzato sia per gli algoritmi evolutivi sia per altri algoritmi paralleli.
Inizialmente il pattern Pool evolution sembrava adatto solamente a contesti molto specifici, poi però è stato dimostrato che può essere utile in tanti altri contesti.
Questo pattern è in grado di catturare l'evoluzione iterativa di una popolazione.



P = pool di oggetti
S = funzione di selezione che ne prende alcuni da P ⇒ parallelizzabile se dividiamo in gruppi gli oggetti
e = funzione che fa l'evoluzione ⇒ parallelizzabile
f = funzione che filtra gli oggetti modificati e in caso li mette nel pool ⇒ parallelizzabile

t = funzione terminazione ad ogni iterazione Controlliamo se siamo arrivati ad una situazione in cui abbiamo fermarci.



primo fai la map e stimi quanto è buono un individuo per l'evoluzione poi riduci avendo filtri i migliori

ogni individuo evolve indipendentemente dagli altri quindi è parallelizzabile

Valutazione di una funzione fitness e i migliori vengono mantenuti.
Possiamo usare la map reduce
Map: filtering function
Reduce: filtra i risultati

Split in gruppi distinti
↓
Evoluzione di ogni gruppo
↓
Rimettiamo gli individui evoluti nella popolazione P e valutiamo la condizione di terminazione

Ci sono altri pattern che sono simili al pattern Pool evolution e possono essere ridotti a questo:
Orbit: è un pattern in cui partiamo da un set di dati iniziali S e poi abbiamo a disposizione vari generatori G. Ad ogni iterazione prendiamo i dati che abbiamo in S e li passiamo ai vari generatori in modo che vengano nuovi elementi.
Il set di dati che vengono generati poi vengono aggiunti ai dati iniziali.
Genetic algorithm: è un pattern in cui abbiamo una popolazione iniziale e ad ogni iterazione facciamo delle modifiche agli individui della popolazione apportando delle modifiche genetiche, il risultato può essere la stessa popolazione o una popolazione aumentata.
Global single population pattern: l'evoluzione in questo caso è un processo che riguarda tutta la popolazione.
MultiAgent System Pattern: abbiamo degli agenti che svolgono una certa attività e attraverso questa attività rappresentano l'evoluzione dell'ambiente circostante.
Questo è il pattern che più somiglia al pool evolution perché abbiamo una funzione di selezione che prende gli agenti che hanno ricevuto un messaggio nella loro coda di input, poi viene fatta una evoluzione basandoci su quello che c'è nella coda di input e poi dopo alla fine la funzione filter è in realtà una funzione che prende l'agente che si è evoluto e lo mette nel gruppo di agenti sostituendo quello non evoluto.
Concurrent memetization pattern: in questo caso abbiamo una selezione della popolazione che poi viene fatta evolvere e ogni volta facciamo anche un'ottimizzazione dell'evoluzione. Mutazioni e valutazioni vengono effettuate in parallelo, la migliore soluzione che otteniamo alla fine diventa il punto di partenza dell'evoluzione successiva.
Ci fermiamo dopo un certo numero di evoluzioni.

Possibili utilizzi:

Generazione di stringhe a partire da una grammatica -> In questo caso ci troviamo di fronte ad un Orbit pattern perché abbiamo delle lettere appartenenti ad una grammatica e vogliamo usare dei generatori per generare le possibili stringhe che hanno quelle lettere.

Sudoku -> Il pool evolution pattern lo possiamo usare per risolvere il sudoku, in particolare ad ogni iterazione produciamo una board che ha tutti i numeri, poi selezioniamo quelle migliori.

Trovare l'approssimazione di una funzione: abbiamo dei punti e vogliamo trovare la funzione che meglio approssima questi punti. Per farlo possiamo usare un pattern evolutivo in cui andiamo a generare le varie possibili funzioni e abbiamo una funzione che mi fa da filtro e mi controlla quale funzione è migliore.

3 possibili implementazioni parallele

- La parte dell'evoluzione la facciamo in parallelo mentre tutto il resto è sequenziale.
- tutte e 3 le fasi sono svolte in modo parallelo. La map reduce viene usata per selection e filter. Per la parte evolutiva si usa la map
- Si divide la popolazione in gruppi e si lavora in parallelo sui vari gruppi

Queste tre versioni hanno un grain differente.