

Backtracking e Job Scheduling Problem
Backtracking: implementazione e applicazione
al problema dell'assemblaggio nella catena di montaggio

Alessandro Bonciani
Febbraio 2024

1 Introduzione

Questa relazione descrive il lavoro svolto per l'esercizio assegnato di Intelligenza Artificiale. L'assegnazione prevede: di implementare l'algoritmo di Backtracking e MAC e di applicarlo al problema di Job Scheduling (descritto in §6.1.2 di R&N 2021), ovvero della ricerca di una serie di assegnazioni che permettano di svolgere i compiti in un dato tempo massimo.

2 Analisi dell'algoritmo

La versione dell'algoritmo a cui si fa riferimento è quella vista durante la lezione del 2023-10-4:

```
1  Backtrack(A,CSP)
2  if A is complete return success or failure
3  var = select_var(CSP,A)
4  for value in order_domain(var,A,CSP)
5      if value is consistent with A
6          A = A u {var = value}
7          inf = inferences (CSP, var, value)
8          if inf != failure
9              add inf to A
10             result = Backtrack(A,CSP)
11             if result != failure
12                 return result
13         else remove assignment and inferences from A
14  return failure
```

dove le funzioni per gestire le inferenze sono state definite come:
removeDomains() e putRemovedDomains()

Cercando quindi di propagare le inferenze e di ridurre quindi i d

3 Implementazione

L'algoritmo è stato implementato in Python, riceve in input il problema e restituisce la soluzione. Ove possibile, si è cercato di rimanere il più vicini possibile allo pseudocodice fornito a lezione. L'algoritmo utilizza il costruttore della classe per prendere in input variabili, domini e vincoli del problema. Successivamente viene effettuata la prima chiamata di

backtracking_search(CSP,A) che procede al prendere una variabile tra quelle non assegnate e tenta di assegnarle un valore fino a che non ne viene trovata una che soddisfi tutti i vincoli tramite **checkAllConstraints(CSP,A)**. Dopo aver trovato una variabile che soddisfa i vincoli, vengono ridotti i domini di tutte le altre variabili e si ripete ricorsivamente la ricerca per ogni variabile. Per la modellazione dei problemi, è stato seguito il riferimento fornito (§6.1.2 di R&N 2021). Le variabili e i loro domini sono stati inseriti usando una hash table in cui le variabili sono state inserite come chiavi e gli elementi erano i domini stessi. I vincoli, composti da disequazioni e una disgiunzione, sono stati rappresentati come una doppia hash table in cui, per ogni variabile, è possibile vedere quali siano le variabili con cui essa ha un vincolo e che tipo di vincolo sia.

4 Test e risultati

E' stato effettuato un ulteriore test sulla correttezza dell'algoritmo utilizzando un altro problema generato randomicamente di 17 variabili tutte con un dominio

$N=\{x \in \mathbb{N} \text{ t.c. } 1 \leq x \leq 50\}$

con i seguenti vincoli:

seguendo questo risultato:

$\text{Var1} + 9 \leq \text{Var3}$,	$\text{Var2} + 9 \leq \text{Var6}$
$\text{Var1} + 9 \leq \text{Var4}$,	$\text{Var2} + 9 \leq \text{Var7}$
$\text{Var1} + 9 \leq \text{Var5}$,	$\text{Var2} + 9 \leq \text{Var8}$
$\text{Var1} + 7 \leq \text{Var17}$,	$\text{Var2} + 7 \leq \text{Var17}$
$(\text{Var2} + 10 \leq \text{Var1}) \text{ or } (\text{Var1} + 10 \leq \text{Var2})$		
$\text{Var3} + 15 \leq \text{Var9}$,	$\text{Var4} + 15 \leq \text{Var10}$
$\text{Var3} + 7 \leq \text{Var17}$,	$\text{Var4} + 7 \leq \text{Var17}$
$\text{Var5} + 15 \leq \text{Var11}$,	$\text{Var6} + 15 \leq \text{Var12}$
$\text{Var5} + 7 \leq \text{Var17}$,	$\text{Var6} + 7 \leq \text{Var17}$
$\text{Var7} + 15 \leq \text{Var13}$,	$\text{Var8} + 15 \leq \text{Var14}$
$\text{Var7} + 7 \leq \text{Var17}$,	$\text{Var8} + 7 \leq \text{Var17}$
$\text{Var9} + 15 \leq \text{Var15}$,	$\text{Var10} + 15 \leq \text{Var16}$
$\text{Var9} + 7 \leq \text{Var17}$,	$\text{Var10} + 7 \leq \text{Var17}$
$\text{Var11} + 7 \leq \text{Var17}$,	$\text{Var12} + 7 \leq \text{Var17}$
$\text{Var13} + 7 \leq \text{Var17}$,	$\text{Var14} + 7 \leq \text{Var17}$
$\text{Var15} + 7 \leq \text{Var17}$,	$\text{Var16} + 7 \leq \text{Var17}$

```
{'Var1': 1, 'Var2': 11,
'Var3': 10, 'Var4': 10,
'Var5': 10, 'Var6': 20,
'Var7': 20, 'Var8': 20,
'Var9': 25, 'Var10': 25,
'Var11': 25, 'Var12': 35,
'Var13': 35, 'Var14': 35,
'Var15': 40, 'Var16': 40,
'Var17': 47}
```

Attraverso il test è stato notato che, al variare dell'ordine in cui vengono assegnate le variabili, il costo computazionale varia in maniera sensibile. Nello specifico, se si parte dalla variabile a cui successivamente viene assegnato l'ultimo valore, l'algoritmo richiede una quantità di tempo ingente per generare il risultato.

5 Bibliografia

Russell, S. J., and Norvig, P. (2020). Artificial Intelligence: A Modern Approach. Pearson education limited.