Min-conflicts e CSP: applicazione min-conflicts sul problema del grafo 3-colorabilità

Jacopo Sabatino

15/06/2020

Anno accademico 2019/2020

1) Introduzione

Seguendo l'esperienza svolta da *Minton et al.* ho applicato l'algoritmo *min-conflicts* sul **CSP** del grafo 3-colorabilità.
Il grafo 3-colorabilità è un grafo in cui ogni nodo deve avere un colore (con una scelta fra 3 possibili) diverso dai suoi vicini.

Min-conflicts è un algoritmo di ricerca locale che usa una formulazione a stato completo: inizialmente ogni variabile ha un valore assegnato che la ricerca può cambiare considerando una variabile alla volta. L'euristica del *min-conflicts* sceglie, per ogni variabile, il valore che minimizza il numero di conflitti con le sue vicine.

I test sono stati svolti su grafi di grandezza variabile (numero nodi **n** = 30, 60, 90, 120, 150, 180).
In ogni grafo, durante l'inizializzazione, i nodi vengono divisi in tre gruppi di grandezza **n/3.**
Dopodiché i nodi sono connessi fra loro con archi che possono collegare solo nodi appartenenti a gruppi diversi.
Il numero di archi dipende dal tipo di grafo che si desidera. Se vogliamo un grafo scarsamente connesso, avremo **m = 2n**, dove **m** è il numero di archi, mentre se vogliamo un grafo densamente connesso, allora gli archi saranno **m = n(n – 1)/4**. Tutti i nodi, dopo l'inizializzazione, risulteranno non colorati.

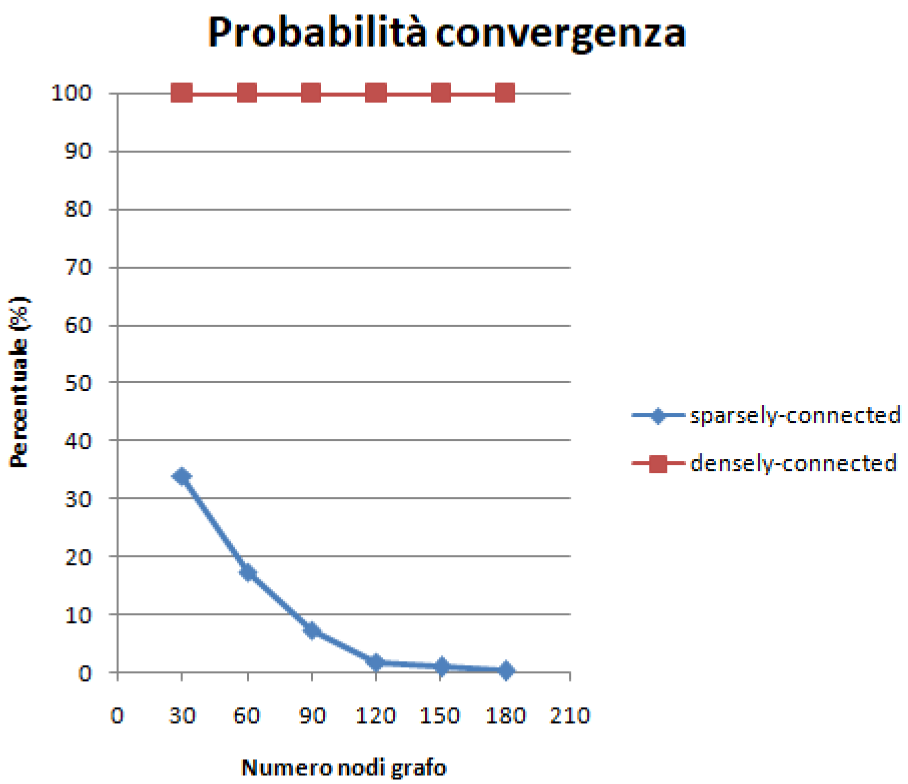
Per ogni grandezza (numero nodi) e tipo di grafo (scarsamente o densamente connesso) sono stati istanziati 3 problemi, su ognuno dei quali è stato applicato l'algoritmo *min-conflicts* per 1000 volte. *Minton et al.* hanno applicato l'algoritmo per 3000 volte, ma per limitazioni tecniche ho dovuto ridurre le iterazioni.

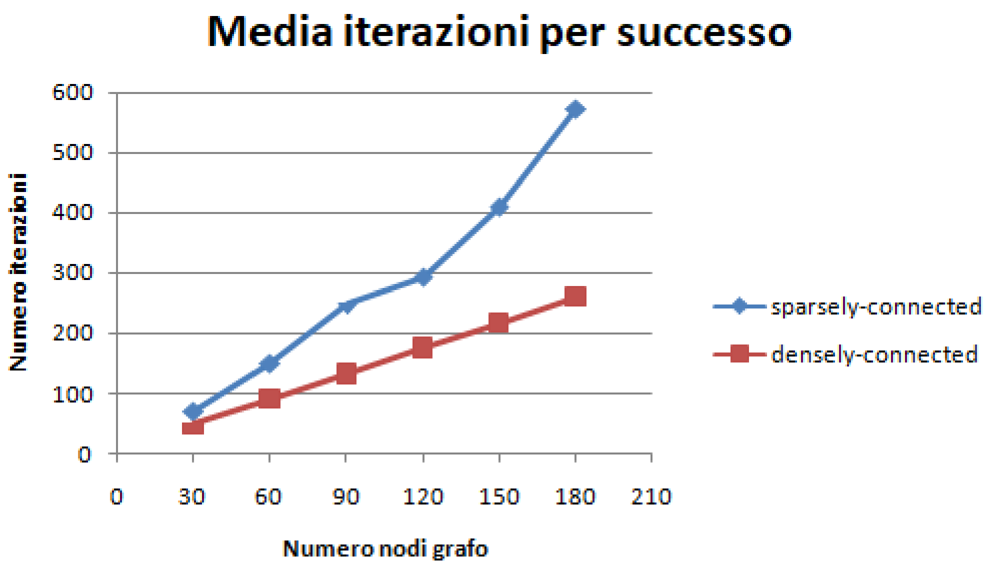
Dopodiché ho svolto altri test nei quali utilizzo il metodo di Brelaz per assegnare un colore a ogni nodo durante l'inizializzazione. In questa fase sono state svolte 100 iterazioni per 8 istanze di ogni grandezza, nel caso dei grafi scarsamente connessi, e 10 iterazioni per 8 istanze di ogni grandezza nel caso dei grafi densamente connessi. Le esecuzioni per i grafi densamente connessi sono in minor numero perché servono solo a dimostrare che in questi casi l'inizializzazione con Brelaz basta a risolvere il problema; infatti, per ogni istanza, la probabilità di convergenza dopo l'applicazione di Brelaz risulta massima. Per questo motivo non mostrerò i grafici riguardanti i grafi densamente connessi.

Per tutti i test è stato fissato un tetto di **9n** iterazioni per il quale l'algoritmo *min-conflicts* s'interrompe, dichiarando così il problema irrisolto.

2) *Min-conflicts* senza inizializzazione dei colori dei nodi

Durante questi test, come specificato sopra, è stato applicato l'algoritmo *min-conflicts* sui grafi con nodi non colorati. Iniziare ogni esecuzione del *min-conflicts* con i nodi non colorati comporta che ognuno di essi è in conflitto con tutti i suoi vicini. Per i grafi scarsamente connessi, all'aumentare di **n** nodi, la probabilità di convergenza cala esponenzialmente, mentre il numero medio d'iterazioni per i problemi risolti aumenta esponenzialmente. Per i grafi densamente connessi la probabilità di convergenza è massima, cioè il **CSP** è sempre risolto, e il numero d'iterazioni medie cresce linearmente rispetto a **n**.

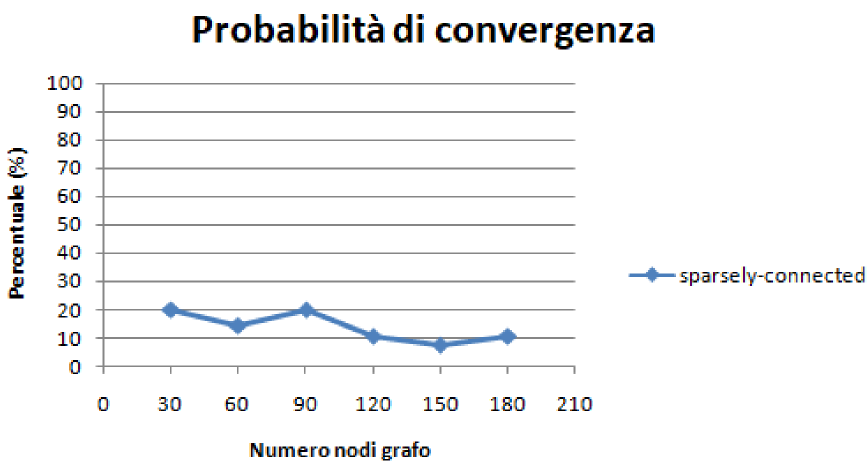


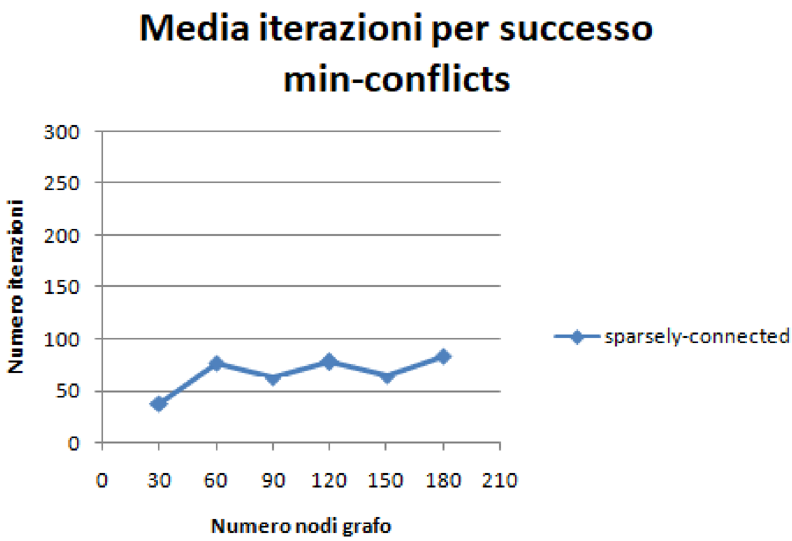


3) *Min-conflicts* con metodo di Brelaz

Il metodo di Brelaz è una regola di ordinamento di variabile che si basa su due criteri. Il primo è la scelta del nodo con il minor numero di colori disponibili (“most-constrained”). Nel caso in cui il primo criterio non sia risolutivo, cioè più nodi hanno lo stesso minimo numero di colori disponibili, viene applicato il secondo criterio. Quest’ultimo consiste nella scelta del nodo di grado massimo fra quelli che causano indecisione al primo criterio (“most-costraining”). Nel caso di un’ulteriore pluralità, viene scelto casualmente un nodo fra i rimanenti. Al momento della scelta del colore, se non ce ne fossero di disponibili, è assegnato quello che minimizza i conflitti, proprio come fa *min-conflicts.*
Brelaz in supporto di un algoritmo di backtracking è il miglior modo di risolvere i problemi di *graph-coloring*. *Minton et al.,* volendo testare l’efficacia di *min-conflicts* sul problema di *graph-coloring,* hanno confrontato il metodo di Brelaz con il *min-conflicts.* Per rendere equo il confronto, prima di applicare *min-conflicts* al **CSP***,* a ogni nodo è assegnato un colore seguendo come criterio il metodo di Brelaz stesso. Per evitare che Brelaz risolva il problema ancor prima di applicare *min-conflicts,* a ogni iterazione si limita la scelta del nodo fra quelli senza colore.

Come si può notare dai risultati, il metodo di Brelaz non incrementa significativamente la probabilità di convergenza del *min-conflicts,* ma il numero d’iterazioni medie per i problemi risolti con successo diminuisce. Naturalmente nei grafi sottostanti non sono stati inseriti i casi banali, cioè quelli in cui il metodo di Brelaz risolve il problema ancor prima di applicare il *min-conflicts*.





Infine riporto la probabilità con la quale il metodo di Brelaz risolve il **CSP** prima dell'applicazione del *min-conflicts*.

Probabilità di risolvere il problema con solo Brelaz		
Grandezza grafo	Sparsely-connected (100 esecuzioni per 8 istanze)	Densely-connected (10 esecuzioni per 8 istanze)
30	71,88%	100,00%
60	67,50%	100,00%
90	52,38%	100,00%
120	42,25%	100,00%
150	47,25%	100,00%
180	39,00%	100,00%

4) Conclusione

Così come dimostrato da *Minton et al.*, dai risultati si può evincere che le prestazioni dell'algoritmo *min-conflicts* applicato al problema del grafo 3-colorabilità dipendono principalmente dalla connettività del grafo, cioè dal numero di archi che connettono i nodi.

Seguendo gli studi di *Cheeseman et al.,* per un grafo di dimensione fissata, esiste, all'aumentare dei vincoli (archi del grafo), una "zona di transizione" per la quale la difficoltà di risoluzione è maggiore a quella di qualsiasi altro problema avente lo stesso grafo sottovincolato o sopravincolato che non risiede dentro questa zona.

Si può concludere che i grafi scarsamente connessi di questo esperimento, data la loro bassa probabilità di convergenza, risiedono nella suddetta zona, mentre quelli densamente connessi risiedono all'esterno di questa nella parte dei grafi sopravincolati.