PROBLEMI DI SODDISFACIMENTO DI VINCOLI

I problemi visti tramite la modellazione di grafo possono essere risolti tramite i vari algoritmi di ricerca vista, in questo approccio per l’algoritmo di ricerca lo stato o nodo del grafo è atomico o indivisibile e rappresenta una sorta di scatola nera. Ora attraverso i CSP ovvero problemi di soddisfacimento di vincoli tratteremo un problema e i relativi stati utilizzando una rappresentazione fattorizzata dove ogni stato viene descritto da una serie o insieme di variabili ogni una delle quali ha un dominio nel quale può assumere valori. In questo approccio una soluzione è data da una assegnazioni sulle variabili che vadano a soddisfare i vincoli fissati su tali variabili.

L’intelligenza artificiale nei problemi visti è sinonimo di computazione.

MONDI POSSIBILI, VARIABILI E VINCOLI

Siamo partiti dal rappresentare problemi tramite spazi atomici con gli algoritmi di ricerca su grafi, mentre adesso ci sposteremo su una rappresentazione degli stati del problema basata su delle caratteristiche. Queste caratteristiche possono essere a loro volta rappresentate sotto forma di variabili e sono tali da poter instaurare delle relazioni di dipendenza. Queste caratteristiche che definiscono lo stato del problema dovranno soddisfare dei vincoli (una forma di regole che permette alle variabili di assumere solo specifici valori). I vincoli[assegnazioni lecite delle variabili] possono essere rigidi (hard constraint) e quindi devono necessariamente essere soddisfatti oppure possono essere flessibili(soft constraint, come accade nell’ottimizzazione) i vincoli flessibili permettono di stabilire criteri con i quali è possibile preferire delle assegnazioni rispetto a delle altre. Nel ragionamento da parte dell’agente intelligente si effettua una computazione che prevede l’assegnamento delle variabili che rappresentano lo stato e in seguito la verifica dei vincoli.

Text

Description automatically generated

Text

Description automatically generated

Ogni problema. Verrà caratterizzato da variabili indicate da lettere maiuscole X alle quali corrisponderà un dominio di valori possibili definito da dom(X) [la + semplice variabile è quella discreta booleana], tuttavia i domini possono essere discreti o continui, finiti o infiniti. Un assegnazione su tutte le variabili del problema prenderà il nome di MONDO POSSIBILE ovvero uno stato in cui può trovarsi il problema, chiaramente poi bisognerà verificare se tale mondo possibile è compatibile con i vincoli imposti dal problema.

Un’assegnazione è una funzione che prende l’insieme delle variabili del problema e le assegna un valore del corrispettivo dominio.

Text

Description automatically generated

I mondi possibili sono dati dal prodotto cartesiano delle cardinalità dei domini delle variabili in gioco.



Questo ci permette di ragionare più facilmente sui problemi poiché gestendo un numero accettabile di variabili è possibile gestire un numero molto elevato di stati, questo è il principale motivo del perché passiamo da una rappresentazione atomica dello stato ad una fattorizzata per mezzo di variabili che permette di gestire meglio la complessità.

DEFINIZIONE DEI PROBLEMI DI SODDISFACIMENTO DI VINCOLI

Un problema di soddisfacimento di vincoli CSP(constraint satisfaction problem) è costituito da 3 componenti:

1. X un insieme di variabili atte a codificare e rappresentare le caratteristiche di uno stato e quindi la rappresentazione del mondo X = {x1, x2, x3, …….., xn}
2. D l’insieme che rappresenta i domini associati ad ognuna delle variabili contenute in X, per cui vi è un particolare dominio per ogni variabile.
3. C l’insieme dei vincoli che vanno a specificare particolari condizioni che devono essere soddisfatte o su una singola variabile o su una tupla di più variabili.

CSP finito: è caratterizzato da un numero finito di variabili ogni una delle quali ha un dominio finito di valori.

In particola un vicolo può essere definito come una coppia<ambito, relazione> ove l’ambito è costituito dalla tupla di variabili sulle quali insiste la relazione specificata. L’ambito prende il nome di scope o anche scheme(schema) e quindi una tupla sull’ambito risulta essere un assegnamento sulle variabili coinvolte nell’ambito. I vincoli possono essere definiti con la terminologia delle basi di dati relazionali in quanto essi rappresentano relazioni tra tuple.

Un assegnamento sulle variabili del problema viene detto mondo possibile e se questo mondo soddisfa tutti i vincoli imposti allora viene detto modello.

MODELLO: è un assegnamento che soddisfa tutti i vincoli, quindi la soluzione per un CSP è un modello sui vincoli.

CSP finito: è un CPS in cui il numero delle variabili è finito così come il corrispettivo dominio di tali variabili.

Dato un CSP è possibile chiedersi :

* Se esiste o meno un modello che soddisfi il CSP
* Trovare un modello ovvero una soluzione(uno qualsiasi quindi un problema di ricerca)
* Trovare tutti i possibili modelli(soluzioni) è leggermente più complicato(nel caso di spazi infiniti serve una modalità o formula che permetta di enumerare le infinite soluzioni)
* Contare il numero delle possibili soluzioni
* Trovare la soluzione ottima, dato un criterio

di ottimalità o una misura di qualità

* Determinare se alcuni enunciati siano veri in tutti i modelli

Text

Description automatically generated

L’idea generale nell’affrontare i CSP è quella di assegnare dei valori presenti nei rispettivi domini alle variabili al fine di trovare un assegnamento che rispetti i vincoli del problema tale assegnamento è detto CONSISTENTE. Nel trovare un assegnamento consistente possiamo tentare o di dare un assegnamento parziale ovvero non tutte le variabili vengono assegnate, ma solo alcune oppure un assegnamento completo su tutte le variabili che rappresentano il problema.

Un CSP può essere astratto e visualizzato tramite una struttura a grafo che metta in evidenza i vincoli, tale struttura prende il nome di grafo dei vincoli ove i nodi sono le variabili del problema e gli archi sono le relazioni espresse dai vincoli che sussistono tra le variabili.

[Motivi per affrontare un problema con i CSP sono:

1. Costituiscono una rappresentazione naturale per molte categorie di problemi.
2. I risolutori di CSP attuali sono molto efficienti
3. Un CSP riesce a potare rapidamente grandi porzioni dello spazio degli stati cosa che con una rappresentazione a stati atomica come quella vista con i problemi di ricerca sarebbe più ostico da effettuare

]

I vincoli possono essere di diverso tipo a seconda della natura del problema che vogliamo rappresentare possono essere:

* **Vincolo di precedenza** specifica che una certa variabile deve rispettare un ordine di cardinalità rispetto ad altre.
* **Vincolo disgiuntivo** permette di effettuare una scelta sull’assegnamento delle variabili che sia mutuamente esclusiva.
* **Vincoli lineari** per trattare domini discreti o continui che siano infiniti (nei domini continui si parla di programmazione lineare).
* **Vincoli unari**
* **Vincoli binari**
* **Vincolo globale** interessa un numero arbitrario di variabili del problema.
* **Vincolo sulle risorse** detto vincolo “atmost” ovvero al massimo. Questo vincolo permette la cancellazione di alcuni valori del dominio in particolare quando questi valori non sono compatibili con la somma data dai valori minimi sulle altre variabili coinvolte nel vincolo di atmost. Permette di definire gli estremi consistenti del dominio di una variabile.

I vincoli possono essere rappresentati meglio con un ipergrafo di vincoli costituito da nodi rappresentati come cerchi e iper-nodi rappresentati come quadrati i quali rappresentano vincoli n-ari.

GENERATE-AND-TEST ALGORITHMS: questo algoritmo tenta di risolvere un CSP di forza bruta andando prima a creare(generare tutte le possibili assegnazioni) e quindi assegnare il mondo alle variabili e in seguito a testarlo per vedere se i vincoli siano soddisfatti. È molto rude come tecnica ma permette di essere esaustivo, ma vale per CSP finiti, poiché per i CSP infiniti potrebbe richiedere un tempo troppo elevato per trovare la soluzione. Se abbiamo n variabili e ogni una ha dominio di cardinalità d abbiamo che l’intero insieme dei domini ha dn elementi che dovranno essere assegnati e verificati quindi complessità esponenziale alla quale si aggiunge la complessità per la verifica dei test sui vincoli nel qual caso con un numero di vincoli pari ad e avremo O(edn) che diviene facilmente intrattabile. A seconda che serva trovare una sola soluzione accettabile o tutte il generate-and-test continuerà a ciclare sullo spazio oppure terminerà dopo aver fornito la prima soluzione.

Text

Description automatically generated

Per migliorare l’algoritmo si potrebbe pensare di iniziare con un assegnamento parziale sulle sole variabili coinvolte nei vincoli questo ridurrebbe la dimensione degli assegnamenti da generare e testare e in seguito su quelli validi si continua fornendo l’altra parte dell’assegnazione. Per fare questo sarà possibile costruire un grafo di ricerca per la soluzione a cui siamo interessati.

Text

Description automatically generated with medium confidence

Il grafo conterrà un certo numero di assegnazioni, i vicini di un nodo sono assegnazioni parziali un po' più lunghe poiché andranno ad aggiungere una variabile con il valore e il collegamento indicherà che i vincoli sono soddisfatti. In questa ricerca sul grafo non siamo interessati al cammino percorso bensì a raggiungere lo stato obiettivo.

Diagram, schematic

Description automatically generated

L’algoritmo di ricerca sul grafo può cambiare a seconda del criterio con cui espandere i vicini e costruire soluzioni parziali via via sempre più complete. La ricerca sul grafo permette di tagliare lo spazio di ricerca per trovare le soluzioni rispetto al generate-and-test.

USO ALGORITMI DI RICERCA PER RISOVERE I CSP, TIPICA RICERCA DEPTH-FIRST SEARCH NOTA COME BACKTRACKING:

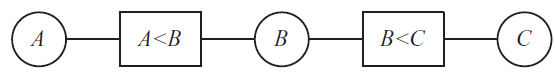
Text

Description automatically generated

Nel backtracking si riscopre l’inconsistenza delle assegnazioni più volte in maniera ripetitiva.

ALGORITMI BASATI SULLA CONSISTENZA

CONSISTENCY ALGORITHM: [idea: si parte da un grafo che rappresenta tutte le variabili con i vincoli e i rispettivi domini e via via si cerca di semplificare il grafo andando a ridurre le dimensioni dei domini delle variabili] sono algoritmi che necessitano di una particolare rappresentazione del grafo del CSP questa rappresentazione prende il nome di constraint network (o di iper-grafo). In questi grafi ci sono nodi ovali che rappresentano le variabili e l’insieme di possibili valori del dominio della variabile che sono assegnabili(inizialmente è impostato a dom(X) ovvero è selezionabile tutto il dominio) e nodi rettangolari che rappresentano i vincoli tra le variabili che sono collegate. Infine, vi è un arco per ogni variabile che risulta coinvolta nel vincolo <X,c>, ne segue che un vincolo n-ario che sarà rappresentato da un nodo rettangolare avrà n nodi collegati con le singole n variabili che partecipano al vincolo



Text

Description automatically generated

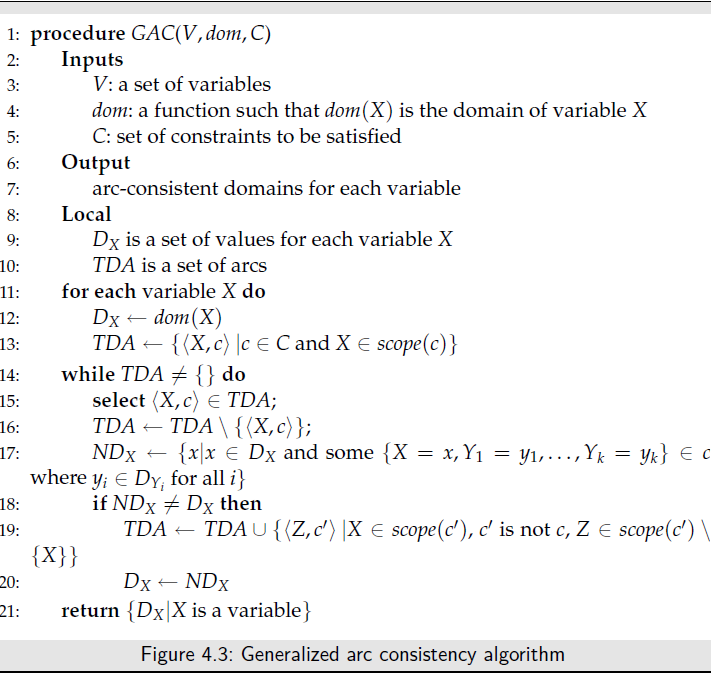
Una volta ottenuto il grafo con tutti gli archi consistenti basterà effettuare il prodotto cartesiano tra le possibili assegnazioni rimaste utili nei domini ridotti per trovare tutte le possibili soluzioni al CSP.

Questa rappresentazione permette di elaborare un algoritmo “Generalise arc consistency algorithm” GAC che permette di trovare un assegnazione per le variabili in modo tale che i vincoli nei rettangoli siano soddisfatti e quindi ogni arco sia consistente (ovvero rispetti il vincolo assegnatoli). La terminazione di un algoritmo GAC permette di capire se il CSP non ha soluzioni poiché uno dei domini associati ad una variabile è rimasto vuoto ovvero senza possibilità di associare un valore del dominio che soddisfi il vincolo sugli archi, oppure se il CSP ha una sola ed unica soluzione e questo avviene quando in ogni dominio per ogni variabile vi rimane un solo possibile valore in grado di soddisfare i vincoli ed infine è possibile capire se ci sono più soluzione nel caso i domini non siano vuoti e c’è né uno con più possibili valori. Va tuttavia evidenziato che il GAC non è un algoritmo specifico per risolvere un CSP bensì serve per semplificarlo al fine di rendere più efficiente la ricerca di una soluzione con gli algoritmi noti per la ricerca, questo perché il GAC andando a rendere tutti gli archi consistenti semplifica i rispettivi domini di ogni variabile che saranno ridotti rispetto agli iniziali domini e su questo nuovo CSP con i domini ridotti dal GAC andrà a lavorare un algoritmo di ricerca che sarà adesso avvantaggiato nel trovare la soluzione cercata.

L’idea del GAC è quella di rendere la rete consistente andando a restringere i domini:

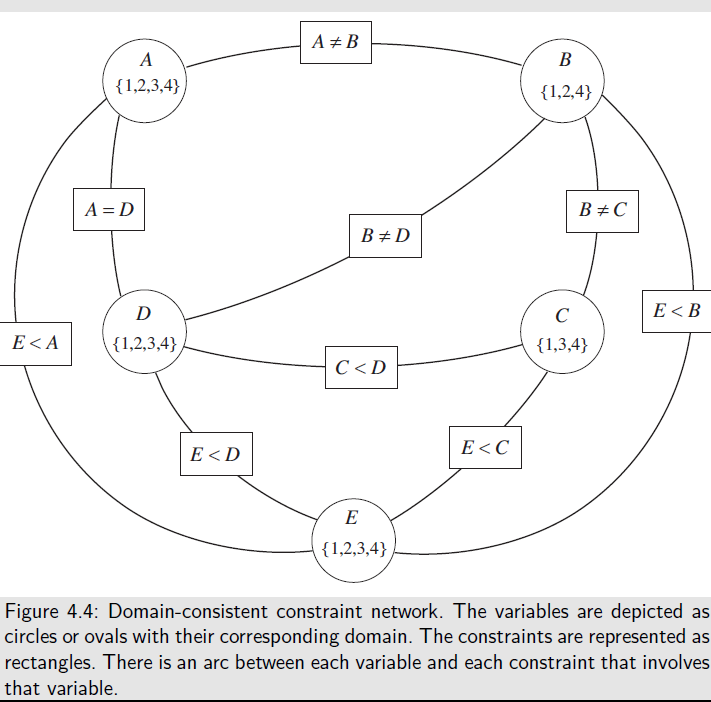
Text

Description automatically generated

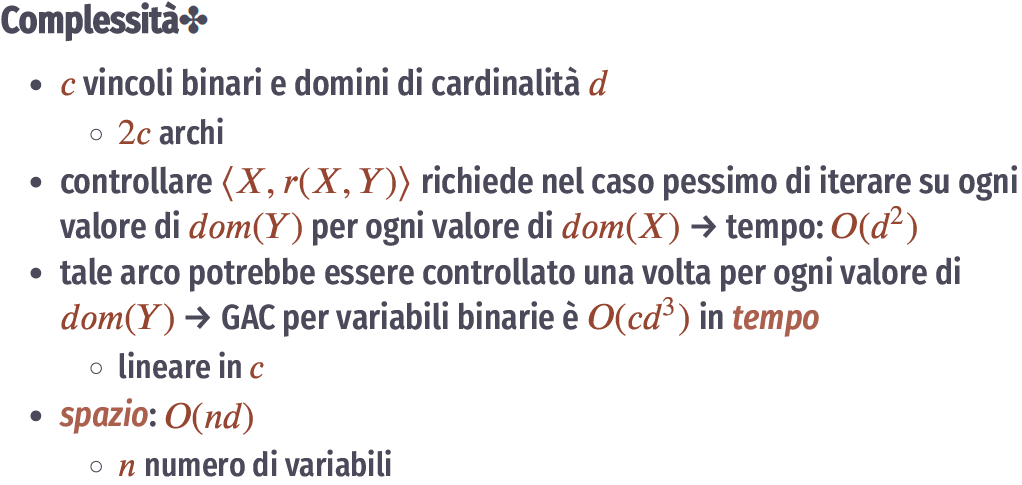


Terminazioni possibili del GAC:

1. Se un solo dominio di una variabile è vuoto allora non esiste una soluzione
2. Se tutti i domini sono stati ridotti ad un solo valore possibile allora esiste una ed una sola soluzione.
3. Il GAC porta ad una rete semplificata sulla quale è possibile eseguire altre strategie di calcolo a seconda delle nostre esigenze.



Studio della complessità del GAC

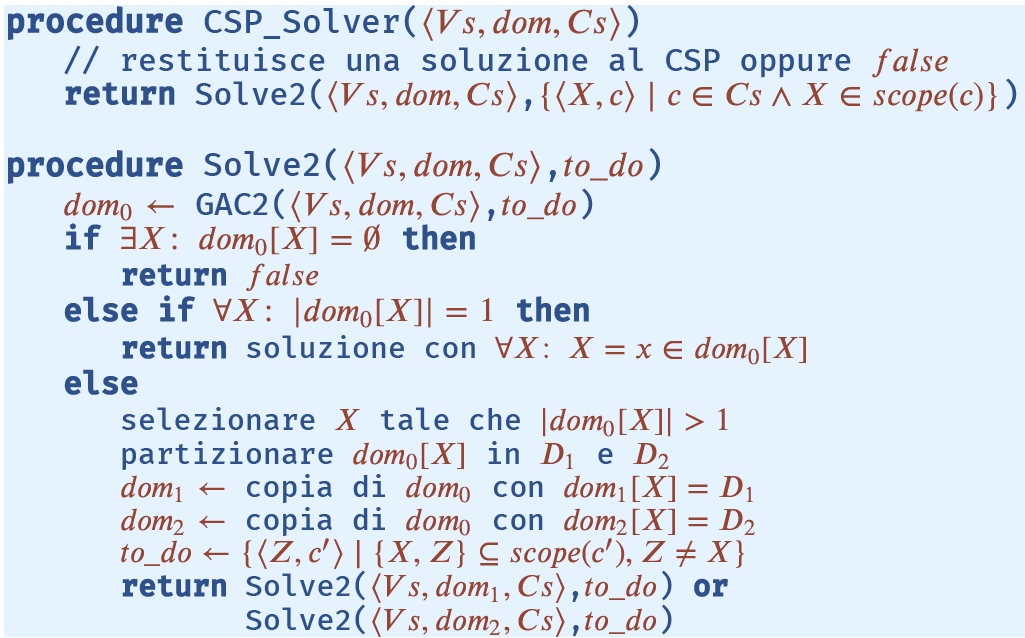


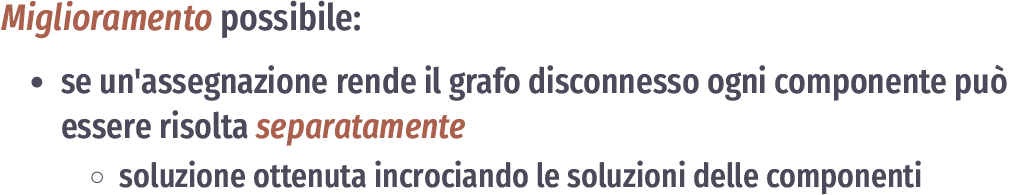
DOMAIN SPLITTING:



l’idea è quella di andare a divide il network constraint(grafo) in sottografi più piccoli e quindi più semplici da risolvere questa tecnica è chiamata domain splitting oppure case analysis. La divisione può essere applicata sui singoli valori del dominio associati ad una variabile oppure dividendo il dominio della variabile in sottoinsieme disgiunti.

Una buona strategia di risoluzione dei CSP prevede l’utilizzo prima del GAC per semplificare la network constraint e in seguito l’utilizzo del domain splitting sui domini rimasti con ancora una notevole scelta di possibilità come assegnazione.

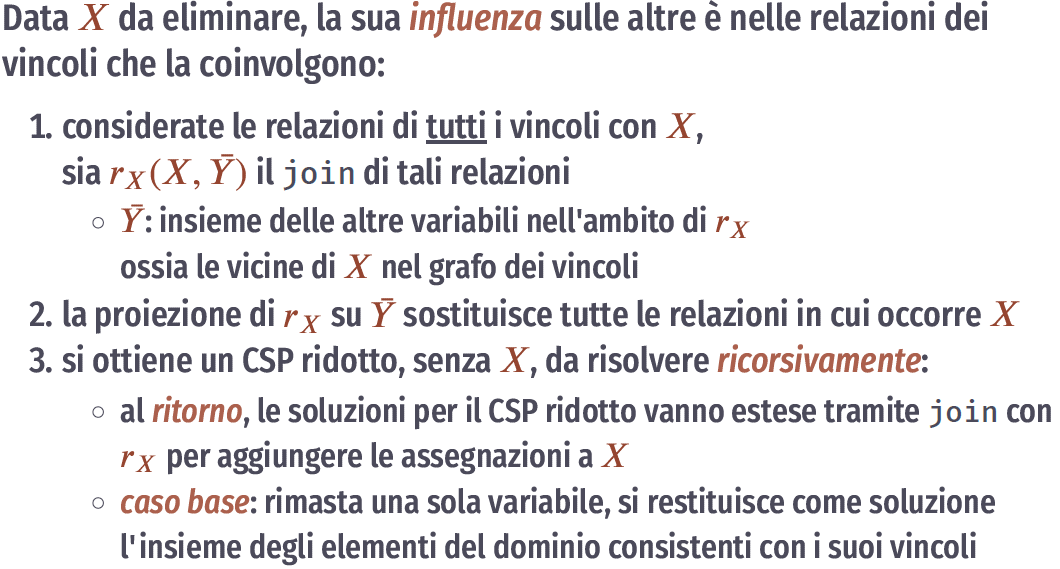


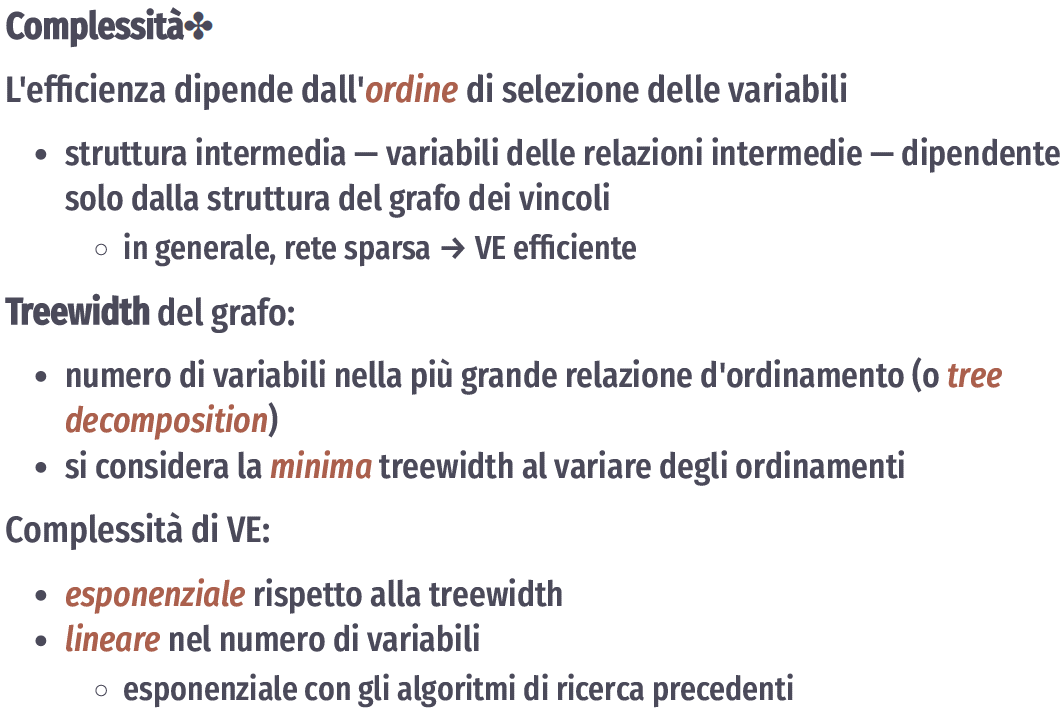


VARIABLE ELIMINATION: mentre il GAC semplifica la network constraint andando a ridurre i possibili valori che è possibile assegnare alle variabile la VE ovvero variable elimination effettua una semplificazione andando a rimuovere le variabili. L’idea e quella di rimuovere le variabili una ad una.

Text

Description automatically generated





Text

Description automatically generated

Nei CSP quando viene utilizzata la tecnica di ricerca in backtracking la quale può essere migliorata sfruttando euristiche indipendenti da i domini le quali traggono vantaggio dalla rappresentazione fattorizzata dello stato in variabili del CSP cosa che invece non succede nei problemi di ricerca non informata ove le funzioni euristiche risultano specifiche per il dominio. Un euristica particolarmente usata nei CSP è la MRV ovvero minimum remaining values o anche conosciuta come euristica della variabile più vincolata o anche fail-first, dato che avrà più vincoli sarà la prima a fallire in caso i vincoli non siano rispettati. Infatti, tale euristica si basa sulla scelta della variabile che ha più probabilità di fallire per prima. Questo fallimento rapido permetterà di potare una buona parte dell’albero di ricerca riducendo lo spazio di ricerca.

Un ‘altra euristica utilizzata è l’euristica di grado detta degree heuristic, questa cerca di ridurre il fattore di ramificazione delle scelte future scegliendo la variabile coinvolta nel maggior numero di vincoli con le altre variabili, una volta effettuata la scelta per la scelta delle successive variabili si usa l’euristica del valore meno vincolante la quale sceglie variabili che sono più libere da vincoli, questa euristica permette di lasciare massima flessibilità sulle successive scelte.

VERIFICA IN AVANTI (FORWARD CHECKING): è una delle forme più semplici di inferenza la quale permette ogni qualvolta si assegna un valore ad una variabile di controllare la consistenza dei possibili valori che le altre variabili coinvolte con quella per cui abbiamo effettuato l’assegnazione siano consistenti ovvero possono assumere in seguito valori dei loro domini che rispettino i vincoli, questa tecnica permette di andare a eliminare dai domini delle variabili coinvolte con la variabile assegnata i valori conflittuali lasciando solo quelli che non causano conflitti e per cui risultano consistenti. La verifica in avanti, tuttavia, può generare inconsistenze che emergono solo alla fine del processo è per questo che ad essa si predilige l’algoritmo del maintaining arc consistency il quale continua ricorsivamente una volta assegnato un valore ad una variabile va controllare le altre variabili implicate in vincoli e in modo ricorsivo controllo il rispetto dei valori nei domini delle successive variabili legati in altri vincoli con quelle già controllate.



BACKTRACKING NEI CSP: il primo approccio più semplice da implementare ed usare è il backtracking cronologico, il quale torna indietro e quindi risale nel albero di ricerca al nodo precedente per cambiare assegnamento ed evitare il fallimento, tuttavia questo approccio non risulta molto intelligente poiché la causa del fallimento potrebbe essersi verificata dei nodi di generazioni più alti per migliorare la tecnica si usa il backjumping il quale conserva un insieme dei conflitti per un nodo ed è capace di tornare indietro nella storia di questo assegnamento, l’insieme dei conflitti può facilmente essere costruito grazie alla verifica in avanti ed è un insieme che conterrà le variabili con il relativo assegnamento che hanno causato il fallimento su di un’altra variabile.

I domini delle variabili di un CSP possono essere discreti oppure continui, finito o infiniti.