

POLITECNICO DI MILANO  
Dipartimento di Ingegneria Meccanica  
Sezione Progettazione e Disegno di Macchine



**ANALISI DI UN APPROCCIO AUTOMATICO  
NELL'INDIVIDUAZIONE DELLE SEQUENZE  
D'ASSEMBLAGGIO**

Relatore: Prof. Roberto Viganò

Tesi di laurea di:  
Alessio Giannotti Matr. 754528

Anno accademico 2011/2012







# Indice dei contenuti

<b>INDICE DEI CONTENUTI</b> .....	<b>III</b>
<b>SOMMARIO</b> .....	<b>V</b>
<b>ABSTRACT</b> .....	<b>VI</b>
<b>CAPITOLO 1 INTRODUZIONE</b> .....	<b>1</b>
1.1 OBIETTIVI NELLA PROGETTAZIONE.....	1
1.2 TECNICHE DI PROGETTAZIONE METODICA E ASSEMBLY SEQUENCES ANALYSIS.....	10
1.2.1 <i>Soluzioni teoriche</i> .....	16
1.3 OBIETTIVO DELLA SEGUENTE RICERCA.....	25
<b>CAPITOLO 2 APPROCCIO ALLA DEFINIZIONE DELLE SEQUENZE D'ASSEMBLAGGIO</b> .....	<b>29</b>
2.1 TECNICHE DI GENERAZIONE DELLE SEQUENZE.....	29
2.1.1 <i>Strategia di implementazione</i> .....	30
2.2 IPOTESI E CONSIDERAZIONI PRELIMINARI.....	33
2.2.1 <i>Tutte le soluzioni possibili</i> .....	35
2.2.2 <i>Riduzione dei collegamenti</i> .....	37
2.2.3 <i>Considerazioni sui grafi</i> .....	39
2.3 METODOLOGIA AUTOMATICA PROPOSTA.....	42
2.3.1 <i>Definizioni</i> .....	43
2.3.2 <i>Procedura</i> .....	52
2.4 IMPLEMENTAZIONE DELL'ALGORITMO ITERATIVO.....	64
2.4.1 <i>Struttura dati</i> .....	64
2.4.2 <i>Struttura del codice sorgente</i> .....	68

<b>CAPITOLO 3 VALIDAZIONE DELL'APPROCCIO AUTOMATICO PROPOSTO .....</b>	<b>75</b>
3.1 VALIDAZIONE ATTRAVERSO CASISTICHE BASE RICORSIVE.....	75
3.1.1 <i>Esempio Valvola Europa</i> .....	80
3.2 VALIDAZIONE ATTRAVERSO IL CASO DI DE FAZIO & WHITNEY .....	85
3.2.1 <i>Tutte le sequenze possibili</i> .....	87
3.2.2 <i>Riduzione sul numero di collegamenti</i> .....	87
3.2.3 <i>Approccio bibliografico (metodo di De Fazio &amp; Whitney)</i> .	88
3.2.4 <i>Approccio tramite la teoria dei grafi</i> .....	90
3.2.5 <i>Metodologia automatica proposta</i> .....	92
3.3 VALIDAZIONE ATTRAVERSO UN CASO PRATICO .....	102
<b>CAPITOLO 4 SELEZIONE DELLE SEQUENZE GENERATE .....</b>	<b>111</b>
4.1 CONSIDERAZIONI SULLE SEQUENZE GENERATE DAL METODO PROPOSTO .....	111
4.2 TECNICHE DI SELEZIONE DELLE SEQUENZE FATTIBILI .....	113
4.2.1 <i>Valutazione della tecnica di selezione proposta</i> .....	117
<b>CAPITOLO 5 CONSIDERAZIONI FINALI.....</b>	<b>126</b>
5.1 CONCLUSIONI.....	126
5.2 ANALISI DEI RISULTATI .....	127
5.3 SVILUPPI FUTURI .....	128
<b>CAPITOLO 6 RIFERIMENTI BIBLIOGRAFICI.....</b>	<b>131</b>

## SOMMARIO

Allo stato attuale, la pianificazione della sequenza di assemblaggio di un prodotto viene svolta manualmente sulla base dell'esperienza del personale. Nonostante questa attività richieda tempi e costi non indifferenti, non sempre garantisce il raggiungimento della migliore soluzione. Avvalersi di una sequenza di montaggio efficiente è essenziale per migliorare la produttività del processo e ridurre il tempo ed i costi dovuti all'assemblaggio del prodotto. Questo lavoro propone lo sviluppo di una metodologia automatica in grado di definire più sequenze di assemblaggio, basandosi sulle sole informazioni ottenute attraverso i contatti e le interferenze estratte dal modello CAD del prodotto. In particolare, il numero di sequenze che sarà possibile ottenere per un medesimo prodotto, sarà legato alla natura combinatoria del problema mediante l'applicazione del metodo iterativo proposto. La possibilità di avere a disposizione una molteplicità di soluzioni di assemblaggio altrettanto valide può fornire così un ausilio al progettista durante la scelta di quella ottima. Inoltre, la tipologia delle informazioni richieste dal metodo consente la sua applicabilità nei primissimi stadi del processo di progettazione, dove sono note solo alcune informazioni progettuali, quali ad esempio il layout della soluzione concettuale. Inoltre, i risultati forniti dal metodo implementato non dipenderanno dalla strada di modellazione CAD intrapresa. La procedura proposta si focalizza sull'ottenimento di un numero ridotto di soluzioni, garantendo la presenza di almeno una fattibile tra esse. La generazione delle sequenze possibili si basa sull'individuazione sistematica di potenziali sottoassiemi all'interno del prodotto. Dopo una breve osservazione relativa alle metodologie correnti sviluppate per pianificare l'assemblaggio, verrà presentata l'implementazione automatica al calcolatore del metodo proposto. Successivamente verrà effettuata una validazione attraverso un esempio bibliografico ed uno pratico. Verrà infine proposta una strategia di riduzione e selezione del numero totale delle soluzioni precedentemente ricavate.

**Parole chiave:** Assiemi, Analisi delle sequenze di assemblaggio, Pianificazione delle sequenze di assemblaggio, Approccio automatico, Progettazione Metodologica.

## ABSTRACT

Nowadays the assembly sequence for the products is often carried out manually and its definition, typically, is very expensive, not guaranteeing optimal solutions. Coming up with an efficient assembly sequence is the essential step to improve process productivity and reduce the time and costs related to assembly activities. The issue related to the assembly sequence of a product depends on the total number of its components. In particular, the number of the possible sequences can be obtained through the factorial calculus regarding the number of products components. This work presents an automatic method with the aim of defining assembly sequences. This method is based on the information about contacts and interferences existing among the components, which is obtained by the assembly CAD model of the product. The level of the information required by this approach allows its implementation at early stages of design, as soon as the layout of the conceptual solution of the product is defined, without considering the method used to model the CAD assembly. The procedure proposed is focused on obtaining a reduced number of assembly sequences, guaranteeing at least one feasible assembly sequence among them. The procedure is oriented to iteratively identify independent and important subassemblies into the CAD assembly, then merge them to specific assembling nodes and generate sequences until the whole product is analyzed. After a brief review of current methodologies developed for assembly planning, will be presented the automatic implementation of the proposed method, with a computer. Afterwards will be carried out a validation through some bibliographical and practical example and will be proposed a strategy for reducing the total field of the previously obtained solutions.

**Keywords:** Assemblies, Analysis of assembly sequences, Assembly sequences planning, Automatic approach, Design methodology.







# CAPITOLO 1

## INTRODUZIONE

### 1.1 Obiettivi nella progettazione

Il progettista oggi ha il compito di cercare di ridurre i tempi ed i costi necessari per lo sviluppo del progetto di un prodotto. La stima dimensionale di un progetto è una delle prime attività cruciali da cui dipende il successo dello stesso. Infatti, è ormai una prassi comune quella di quantificare i tempi ed i costi necessari a realizzare un progetto o, se si vuole, la sua durata.

Il team di progettisti di un'impresa odierna si trova così a dover portare a termine un progetto completo di dettagli costruttivi secondo le specifiche richieste dal committente, soddisfacendo al tempo stesso eventuali vincoli progettuali quali budget a disposizione dell'impresa e date di scadenza concordate con il cliente.

A livello industriale, esistono 5 dimensioni specifiche relazionate al profitto che vengono comunemente usate per valutare le prestazioni di un'attività di sviluppo del prodotto (Ulrich, et al., 2000). Queste dimensioni sono:

- La qualità del prodotto.
- Il costo del prodotto.
- Il tempo di sviluppo del progetto.
- Il costo di sviluppo del progetto.
- Le risorse necessarie per lo sviluppo del progetto.

Dal momento che la produzione di un manufatto o di un bene da parte di un'azienda è sempre relazionata al profitto, un'impresa competitiva cercherà di migliorare la qualità del proprio prodotto e di aumentare la propria produttività senza incidere notevolmente su costi di produzione e senza aumentare, al tempo stesso, tempi e costi di sviluppo del prodotto in considerazione (Ulrich, et al., 2000).

La complessità derivante dallo sviluppo di un valido progetto può essere ridotta e semplificata usando un approccio sistematico di organizzazione ed integrazione del flusso di informazioni che viaggiano lungo le varie fasi della progettazione, compresa quella che concerne l'analisi e lo sviluppo delle procedure di assemblaggio. Tuttavia, i modelli che sfruttano questo tipo di approccio, affinché producano un risultato concreto, hanno la necessità di venir costruiti su degli input che siano sempre i più recenti ed accurati possibili.

Grazie al continuo sviluppo tecnologico e grazie all'introduzione dei calcolatori in supporto alla progettazione e al data management, il personale di un'impresa possiede ora una notevole quantità di informazioni inerenti il progetto. Avendo a disposizione tali risorse, si può dunque pensare di sviluppare degli strumenti che forniscano un supporto nelle scelte del progettista.

E' ancora del tutto noto il fatto che il progettista, nonostante sia aiutato nella sua scelta da un calcolatore, debba possedere una vasta conoscenza delle problematiche riguardanti l'intero ciclo di vita del prodotto al fine di produrre dei risultati competitivi. Proprio per questo motivo è importante avere, già dai primi stadi della progettazione, un'idea di quale saranno i processi di produzione richiesti, siano essi delle lavorazioni meccaniche, dei processi di trasformazione o delle fasi di montaggio, e come si potrà intervenire su essi per apportare delle modifiche al prodotto.

Bisogna innanzitutto chiarire che ogni prodotto ha un suo ciclo di vita (Product Lifecycle) che può essere generalmente suddiviso in cinque macrogruppi (come mostrato in Figura 1.1) che sono: il concept e lo studio di fattibilità, la progettazione e lo sviluppo del prodotto ideato, la sua produzione, il suo uso e il riciclo finale una volta che il prodotto perde di utilità o diventa obsoleto.

I metodi e le tecniche sviluppate in questa tesi possono idealmente venir racchiuse all'interno di quella fase del ciclo di vita del prodotto che prende il nome di progettazione e sviluppo. Quello che può essere di supporto al progettista durante lo sviluppo e il progetto del prodotto è una metodologia sistematica che richiami a se tutti i metodi e le tecniche utili a fornire un prodotto innovativo che abbia successo sul mercato. Bisogna tenere presente che nel ramo della progettazione non ricade solo la creazione di un nuovo modello, bensì anche la riprogettazione di un modello già esistente; non è detto che quest'ultima attività richieda tempi inferiori rispetto alla generazione di un prodotto nuovo sul mercato.

Da quanto detto sopra, si capisce come illustrare l'intero processo di sviluppo mediante una serie di fasi sequenziali ed ordinate, sia soltanto un

modo per schematizzare una realtà molto più complessa. La presenza di feedback interni al ciclo rende il processo notevolmente complesso, oltre alla presenza di interazioni ed effetti a cascata sul ciclo (per esempio dalla produzione all'ingegneria, fino a risalire a decisioni strategiche). Inoltre, può capitare che una delle scelte prese durante le diverse fasi progettuali perda di coerenza, specialmente nel caso in cui si presentino delle criticità a cavallo di due diverse fasi della progettazione.

Si pensi ad un problema di produzione inatteso, che provoca la necessità di apportare modifiche ad una o più parti del prodotto in questione; questo può comportare il bisogno di dover modificare (o nei casi peggiori di riprogettare) la linea di produzione a valle della modifica.



Figura 1.1 Ciclo di vita del prodotto

Il processo di progettazione e di sviluppo di un prodotto è sempre stato un argomento di interesse, sul quale molti ricercatori hanno provato a creare delle proprie metodologie sistematiche, al fine di non trascurare attività fondamentali alla progettazione durante l'iter logico di progetto.

A sua volta il processo di progettazione e di sviluppo può essere sommariamente suddiviso in quattro grandi aree che sono: la concettualizzazione, l'integrazione, la fase di dettaglio e di analisi, l'acquisizione delle informazioni e delle rappresentazioni fondamentali (Pahl, et al., 2007), queste vengono molto spesso precedute da una importante fase di progettazione del prodotto dove informazioni riguardo il mercato attuale, lo stato dell'arte delle tecnologie applicabili alla produzione, le normative vigenti e la concorrenza vengono collezionate ed analizzate con lo scopo di indirizzare nella direzione più opportuna l'intero ciclo di sviluppo del prodotto. Questo primissimo stadio della progettazione prende il

nome di Studio Di Fattibilità di un prodotto. Tra le molteplici opportunità offerte dallo studio di fattibilità, in base al budget a disposizione, un portfolio di progetto viene scelto delineando tempi e costi di produzione, risorse necessarie e tecnologie applicabili per la creazione del “prodotto obiettivo”.

Nel recente 1995 il ricercatore John Sharpe (Sharpe, 1995) ha identificato le successive fasi di progettazione e sviluppo di un prodotto secondo i lavori di ricerca portati avanti da:

- **Pahl e Beitz** : Funzionamento delle strutture.
- **Hubka e Eder** : Teoria dei sistemi tecnici.
- **Tjalve** : Teoria dei domini e degli schemi ad albero funzionali.
- **Parmee e Sen** : L'applicazione di algoritmi genetici basandosi sulle tecniche di ottimizzazione e di decision making secondo molteplici criteri di selezione studiate da Sharpe.

Genericamente, secondo Pahl e Beitz, le difficoltà che il progettista incontra durante lo sviluppo di un progetto valido possono essere sommariamente riassunte in :

- Identificazione di un indesiderabile od insoddisfacente stato iniziale.
- Esistenza di un desiderabile od almeno più soddisfacente stato finale.
- Conoscenza degli ostacoli nella trasformazione dallo stato iniziale a quello finale ad un tempo fissato.

Queste difficoltà incontrate si può supporre in linea di massima che siano dovute ai seguenti fattori :

- Complessità del prodotto o della trasformazione
- Interazione tra diversi progetti di varie componenti in fase di progettazione.
- Incertezza: non tutti i requisiti sono noti, non tutti i criteri sono stabiliti; alcuni effetti non sono completamente noti ed emergono col tempo.

Il problema della progettazione meccanica è generalmente un problema di sintesi e, dunque, prevede lo sviluppo di capacità diverse da quelle analitiche, favorite da altre discipline di base per l'Ingegneria. Un modo per cercare di ridurre ulteriormente le difficoltà incontrate durante le attività di progettazione, potrebbe essere quello di basare le proprie scelte concettuali non più sull'esperienza acquisita, ma sull'applicazione di tecniche metodiche di decisione. La progettazione infatti è un'attività molto complessa e diversi sono gli elementi che vi fanno parte e le discipline che entrano al suo interno. La progettazione metodica, cioè sistemistica, cerca di mettere ordine a questa complessità di elementi. Il suo scopo è quello di fornire delle

procedure generali di ragionamento per affrontare dei problemi progettuali comunque complessi, senza tralasciare nessuno aspetto del ciclo di vita del prodotto. In questo lavoro, verrà preso come riferimento principale, ma non unico, la metodologia di progettazione proposta dai ricercatori Pahl e Beitz.

Secondo i loro studi, facendo riferimento alla Figura 1.2, gli stadi del processo di progettazione possono essere sommariamente raccolti in quattro fasi progettuali:

**1. Chiarificazione del compito**

Consiste nella raccolta e nella catalogazione di informazioni sulle caratteristiche e sui vincoli del progetto, al fine di definire le specifiche primarie come task e procedure, i requisiti, le specifiche e la ricerca dei principi risolutivi.

**2. Progetto concettuale**

Il Progetto concettuale riguarda il secondo passo del processo di progettazione. Le scelte fatte in questa fase hanno un'influenza significativa su parametri quali costi, prestazioni, affidabilità, sicurezza ed in generale sul successo commerciale del prodotto ideato. La maggior parte delle decisioni prese in tale contesto sono scelte strategiche decisive che solo con difficoltà, possono essere successivamente modificate. Sebbene una grande quantità di informazioni siano manipolate in un tempo relativamente ristretto, il progetto concettuale è difficilmente supportato da strumenti e metodi dedicati. Il progettista in linea di massima non segue un percorso logico o uno schema procedurale fissato, ma fonda le sue idee sull'intuito e l'esperienza

**3. Progetto di massima**

Durante questa fase, il progettista partendo da un concetto definisce le caratteristiche tecniche e di costruzione del prodotto sulla base delle scelte prese durante la progettazione concettuale e verificando che queste siano in linea con i limiti tecnici ed economici.

Infatti, il progetto di massima serve fundamentalmente a fornire al committente un'idea globale di quale sia l'opera che si sta per eseguire e a dargli una prima valutazione sul costo di costruzione, permettendo allo stesso di calibrare e di confrontare le proprie esigenze con la proposta del progettista.

**4. Progetto di dettaglio**

Il progetto di dettaglio consiste nell'ultima passaggio del processo di progettazione, in esso vengono decisi e ottimizzati parametri di particolare e finiture, vengono completati i disegni di dettaglio e definiti i principi di produzione. Vengono inoltre sistemate le dimensioni, le tolleranze ed i materiali di ogni singolo componente, con riferimento alla sua fase produttiva.

Nella seguente Figura 1.2 viene illustrato lo schema procedurale di massima proposto per il processo di progettazione; è facile intuire come i miglioramenti e le modifiche vengano apportate a monte e a valle di ogni passo della realizzazione. Oltre ai loro studi in questa disciplina ed oltre ad avere individuato queste fasi fondamentali, Pahl e Beitz hanno portato avanti diverse attività di verifica, di rappresentazione e di raccolta di informazioni dove diverse metodologie venivano impiegate, grazie all'ausilio di computer e database, al fine di ottimizzare e velocizzare il processo stesso. Lo sviluppo di assiemi meccanici complessi che costituiscono un prodotto per l'azienda, richiede un gran numero di persone, di materiali e di tecniche di lavorazione ed assemblaggio. Questi devono essere anticipatamente progettati e testati per poi poter essere integrati e , se possibile, ulteriormente testati: così facendo risulterà un processo lungo e molto complesso.

Varie tecniche informatiche sono state introdotte nel processo stesso al fine di ridurre i costi ed i tempi di sviluppo del progetto, tra le più comuni ci sono gli strumenti di simulazione e modellazione virtuale (CAD, CAM e CAE) e gli strumenti per la gestione delle informazioni del prodotto quali il Product Data Management (PDM). Al giorno d'oggi, ogni azienda di produzione manifatturiera si affida a questi strumenti con lo scopo di semplificare le operazioni necessarie presenti durante lo svolgimento del processo. La metodologia di Pahl e Beitz è stata così largamente accettata da molti ricercatori, anche grazie al fatto che questa attua un approccio funzionale al problema già dalle primissime fasi del processo di progettazione. Proprio per questo motivo, molti strumenti in ausilio alla progettazione sono stati creati sulla stessa linea guida della metodologia da loro proposta.



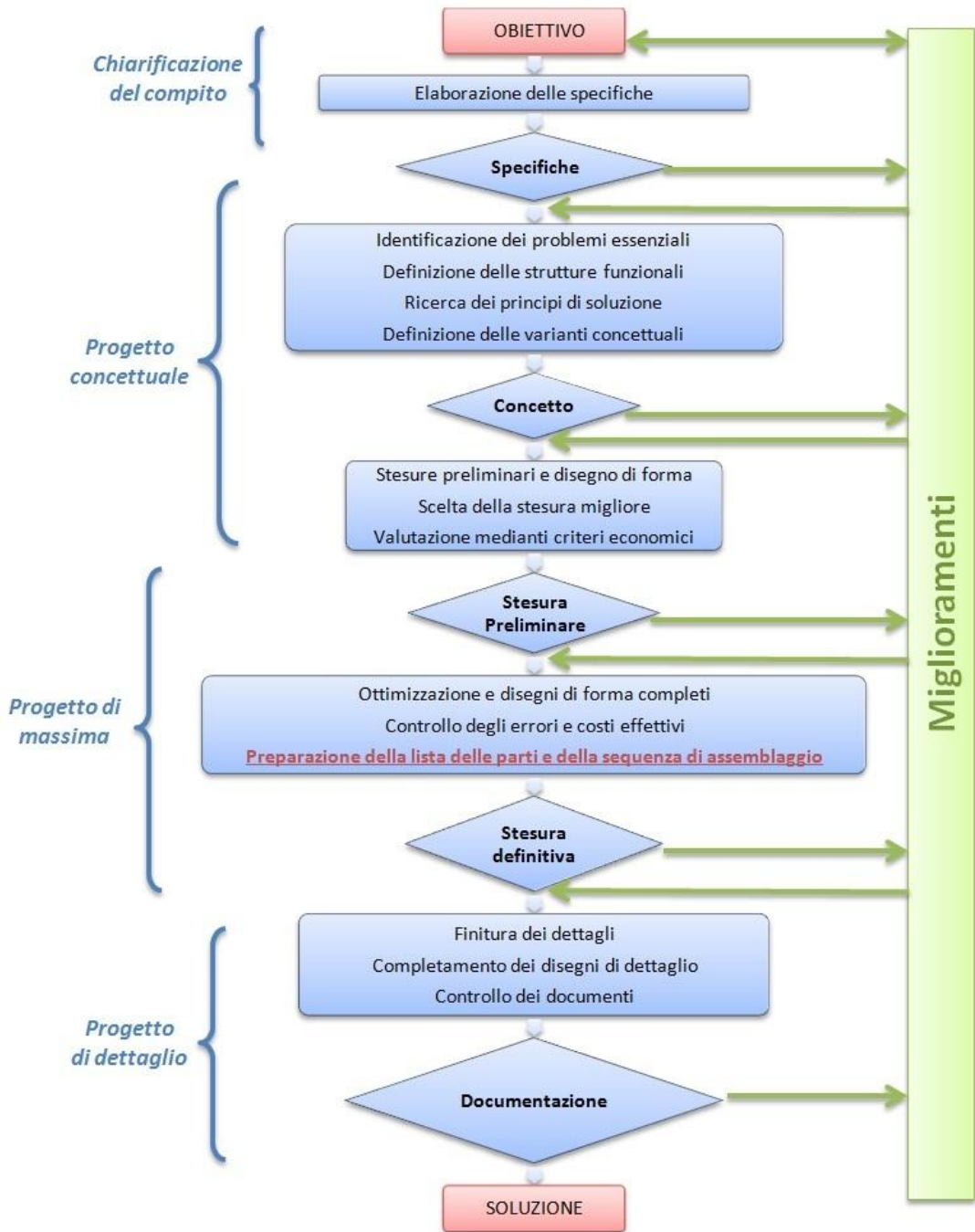


Figura 1.2 Schema del processo di progettazione definito da Pahl e Beitz.

In dipendenza del tipo di prodotto che deve essere sviluppato, alcune delle fasi mostrate in Figura 1.2 possono venire trascurate o inglobate in una sola istanza. Durante il passo concettuale del processo di sviluppo, il progettista può venir supportato nella generazione, nella decisione e nel paragone tra le diverse soluzioni costruttive che ha creato, attraverso degli strumenti dedicati. Nasce così la necessità di poter fornire strumenti di supporto alle scelte fatte durante tutto il processo di progettazione, in particolar modo durante la fase concettuale, dove la maggior parte delle idee e delle decisioni che influenzano la scelta del committente sono già state appurate.

Nonostante il processo di produzione viene definito in ogni sua parte soltanto nelle ultime fasi di dettaglio, dove tutti gli aspetti tecnici del prodotto sono già stati accordati, esistono dei layout globali del prodotto che contengono delle informazioni tecniche che sono state fissate durante le fasi antecedenti. Molte di queste informazioni sono solamente qualitative o topologiche e rimarranno invariate durante l'intero processo, nonostante tutti gli altri dati, quali dimensioni e tolleranze vengano modificati volta per volta. Questo fatto indica la possibilità di poter effettuare delle valutazioni concrete già nelle primissime fasi della progettazione.

In accordo con ciò, viene ipotizzato che: “partendo da un layout definito nelle primissime fasi del progetto, è possibile definire correttamente dei particolari che possono venire usati come informazioni preliminari al fine di ottenere promettenti principi risolutivi se non addirittura delle soluzioni concrete”.

Gli sviluppi del concept e le scelte prese nelle prime fasi del prospetto influenzano pesantemente il costo finale del prodotto, dal momento che, come già detto, nelle ultime fasi del piano di lavoro, una modifica sostanziale al concept o a delle caratteristiche funzionali richiede la necessità di dover rivedere tutte le scelte prese fino a quel punto, in termini di valutazioni sul prodotto e sul ciclo di produzione.

L'adempimento delle funzioni tecniche richieste dal prodotto, il raggiungimento di una fattibilità economica e l'osservanza delle norme di sicurezza in merito agli utenti e all'ambiente circostante possono essere considerati come gli obiettivi che un progettista si prefigge di raggiungere nel suo progetto. Il raggiungimento parziale di questi obiettivi, come ad esempio il raggiungimento delle richieste tecniche ma non della fattibilità economica, portano ad un insuccesso nel progetto finale; così, al fine di ottenere un progetto valido e meritevole, il progettista deve sempre cercare di non trascurare nessuno di questi tre aspetti. Oltre a queste tre caratteristiche fondamentali, in fase di progetto è importante tenere conto di altri aspetti, altrettanto importanti ma non sempre vincolati alla fattibilità o meno del prodotto, come la sicurezza, l'ergonomia, la produzione, i controlli

di qualità, l'assemblaggio, il trasporto, l'operabilità, la manutenzione, il dispendio energetico e il riciclaggio.

All'interno del ciclo di vita del prodotto, valutazioni e simulazioni sono relazionate all'analisi di parametri funzionali e di natura quantitativa del prodotto presenti solamente nelle fasi più mature della progettazione. Questi possono essere criteri quali dimensione, materiale impiegato, resistenza meccanica e via dicendo. L'impiego di modelli fisici o virtuali, per la valutazione del progetto in atto avviene solo nelle ultime fasi del processo di sviluppo.

In questo modo, se in fase di analisi viene deciso di effettuare qualche cambiamento sostanziale, il progettista è obbligato a rivedere e correggere tutto il lavoro precedente causando un grande spreco di tempo che in termini aziendali equivale ad una perdita di profitto. Proprio per questo motivo, ultimamente sono state condotte e portate avanti diverse ricerche per sviluppare e integrare prototipi virtuali nelle primissime fasi del processo di progettazione, a scapito di prototipi fisici molto più costosi e molto meno flessibili ai cambiamenti. Grazie all'ausilio di questi nuovi prototipi virtuali, che permettono una valutazione di un gran numero di parametri di progettazione, è possibile ridurre i tempi ed i costi di sviluppo del prodotto, aumentare la sua qualità e permettere al progettista di realizzarsi nel pieno delle sue capacità (Bordegoni, et al., 2006). Si sottolinea così, ancora una volta, la necessità di poter avere a disposizione dei parametri di stima già nelle primissime fasi della progettazione al fine di ottenere un progetto completo e di successo.

Nell'industria manifatturiera odierna, una buona parte delle attività produttive è dedicata all'assemblaggio delle parti che costituiscono il prodotto finito. Così, molte imprese, con la necessità di sviluppare un progetto competitivo rispettando gli stretti vincoli temporali imposti dal mercato, si sono spinte nella ricerca di tecniche di progettazione sempre più efficienti relazionate all'assemblaggio. In base a quanto appena affermato, lo scopo di questa tesi risulterà quello di inquadrare il miglioramento delle fasi di progettazione riferite all'assemblaggio. In particolare, ci si preme di valutare se un approccio di tipo metodico, in ausilio alle scelte del progettista, possa apportare dei contributi significativi nell'efficienza di pianificazione delle sequenze di assemblaggio.

## 1.2 Tecniche di progettazione metodica e Assembly Sequences Analysis

La fase di pianificazione delle sequenze di assemblaggio di un assieme meccanico composto da un numero discreto di parti, in linea di massima, è sempre svolta in maniera manuale sul modello finito, generalmente tramite l'ausilio di un software CAD che individua i contatti e le interferenze tra le parti agevolando il progettista nella scelta della sequenza ottimale.

Questa parte del processo di sviluppo del prodotto può essere molto lunga e costosa e non è detto che porti a dei risultati soddisfacenti durante la produzione del prodotto. Inoltre la definizione di una corretta sequenza di assemblaggio richiede delle informazioni che non sono reperibili nelle fasi concettuali quali la geometria, la posizione relativa della parte nell'insieme e le interazioni che sussistono tra le varie parti che lo compongono.

Negli ultimi 25 anni diversi progetti di ricerca sono stati condotti al fine di individuare un metodo univoco e veloce per caratterizzare delle possibili sequenze di assemblaggio di un assieme composto da un numero discreto di parti. Questa pressione sulla ricerca in tale ambito è dovuta sostanzialmente al fatto che, in accordo con quanto dice Ritchie (Ritchie, et al., 1999), la creazione di prodotti per assemblaggio è di fondamentale importanza per l'economia manifatturiera moderna. Infatti, si attribuisce il 40 % del costo del prodotto alla sola fase di montaggio.

Come già detto, la pianificazione dell'assemblaggio in passato è sempre stata svolta in maniera soggettiva e senza nessuno schema procedurale, sfociando così in lunghi processi di ideazione basati solamente sull'esperienza e sulla intuitività. Il progettista si ritrovava così a dover formare un cosiddetto "piano di assemblaggio" in coda a tutte le altre fasi di progettazione, spesso dopo che molte delle decisioni importanti e incisive erano già state completate. Dalla fine del 1980, molte aziende hanno decentrato la propria produzione, spostando in sedi specializzate la fase finale di assemblaggio; questo per velocizzare e rendere economicamente più vantaggioso un lavoro lento e dispendioso. Il risultato di questo cambiamento consistette in una diminuzione della propria capacità produttiva, in una riduzione del controllo sulla qualità del prodotto finito e nell'aggiunta di lunghe code e tempi di attesa nell'intero ciclo produttivo. Queste conseguenze portarono alcune aziende a fare un ulteriore sforzo, in termini di costi e di tempi, al fine di sviluppare una fase di progettazione dedicata esclusivamente all'assemblaggio: questa prese il nome dal suo acronimo DFA (Design For Assembly). Grazie agli studi portati avanti nel campo della progettazione per assemblaggio, ossia nel DFA, alcune grandi aziende manifatturiere sono riuscite a rendere più veloce ed economica la fase di montaggio delle parti

che costituiscono il prodotto; inoltre, grazie alle migliorie apportate da questi nuovi studi, è stato addirittura possibile, in alcuni casi, automatizzare o rendere metodico l'intero processo.

L'organizzazione delle operazioni di assemblaggio può generalmente venir suddivisa in due macroaree che riguardano principalmente il progetto orientato all'assemblaggio ed il flusso delle operazioni che lo compongono. Boothroyd classifica i sistemi di assemblaggio, o più generalmente le sue macchine, in base al modo temporale in cui esse compiono l'operazione di messa assieme, quindi suddivide le macchine che operano in maniera sincrona da quelle che operano in maniera asincrona (Boothroyd, et al. 1993). Un sistema di assemblaggio sincrono è un sistema che permette di montare le parti solo secondo un preciso ordine cronologico, viceversa, un sistema di assemblaggio asincrono non è caratterizzato da una discretizzazione temporale delle attività di montaggio. Per quanto riguarda il primo, la maggior parte delle volte esso è completamente automatizzato ed è in grado di assemblare una sola tipologia di prodotto, mentre il secondo è caratterizzato sicuramente da lead-time molto più lunghi, ma anche da una maggiore versatilità sulle diverse tipologie di prodotti da montare.

Lo sviluppo di metodologie atte all'identificazione o alla generazione di sequenze di assemblaggio di un prodotto è cresciuto in maniera del tutto indipendente dalla letteratura sugli stessi argomenti. Questo perché i metodi analitici forniti sono difficilmente applicabili su un assieme con un discreto numero di parti senza che questo comporti un dispendio di tempo in termini di calcoli computazionali e analisi dei risultati ottenuti. Proprio per questo motivo, finché gli strumenti a disposizione forniranno un elevato numero di soluzioni di cui non è certa la fattibilità, i lunghi tempi di selezione dei risultati concreti da parte di un esperto, renderanno gli stessi strumenti di scarsa utilità da un punto di vista pratico.

Con lo scopo di ottenere delle soluzioni efficaci, i precedenti autori affermano di comune accordo la necessità di rispettare i seguenti punti base:

- Assicurarsi che il prodotto sia assemblabile.
- Identificare ogni strumento di montaggio e di fissaggio delle parti.
- Ottenere dei tempi e dei costi per le relative attività di assemblaggio.
- Fornire un documento di riferimento comprensibile agli addetti nella linea di montaggio.
- Sviluppare dei gruppi di operazioni da effettuare in maniera che il corpo di ognuno venga schedato e ottimizzato singolarmente.

- Determinare dei punti convenienti in cui effettuare dei test di qualità e preparare le superfici.

All'inizio, lo studio dell'assemblaggio di un prodotto veniva svolto in coda a tutte le altre fasi del progetto indipendentemente e senza tenere conto del ciclo di vita esistente. Questa tendenza ha lentamente lasciato spazio ad una progettazione sempre più mirata ad inglobare tutti gli aspetti del ciclo di vita del prodotto. In questo modo sono nate delle branche nella progettazione che si dedicano ad un particolare aspetto produttivo quali il Design For Manufacture (DFM), il Design For Assembly (DFA), il Design For Life-cycle (DFLc), il Design For Quality(DFQ) e così via. Tutti questi metodi di progettazione hanno preso così il nominativo di Design For X, ossia della progettazione rivolta all'ottimizzazione di una determinata parte del ciclo produttivo del prodotto, che nel caso generico rimane intesa come acronimo X. Chiaramente il Design For Assembly (DFA) e il Design For Manufacturing (DFM) sono fortemente correlati tra loro, dal momento che molte operazioni di montaggio vengono effettuate direttamente sulla linea produttiva di manifattura.

Esiste un'ulteriore suddivisione basilare sui possibili sistemi di assemblaggio esistenti; questi sono i sistemi manuali (dove le operazioni di montaggio vengono effettuate manualmente), i sistemi con macchine automatizzate flessibili (semiautomatizzate) ed i sistemi con macchine di assemblaggio completamente automatiche (Boothroyd, et al., 1993).

Il Design For Assembly è principalmente orientato a progettare il prodotto al fine di ridurre i costi ed i tempi nelle operazioni di montaggio, ad esempio cercando di minimizzare il numero di parti o assicurando un facile montaggio tra queste. Alla stessa maniera, il DFA porta alla ricerca di una modularità del prodotto dal momento che creando delle famiglie di manufatti, si classifica in qualche modo la molteplicità di casistiche riguardanti il montaggio.

Whitney ha inoltre reso disponibile una buona revisione delle teorie correlate alla DFA dove alcune idee da lui trattate possono essere elencate come segue in ordine cronologico (Whitney, et al., 1999) :

- Assiemi determinati staticamente (1954)
- Modelli connettivi di un assieme (1972)
- Progettazione basata sulle feature(1986)
- Analisi delle tolleranze in un assieme (1988)
- Modellazione funzionale di un assieme (1991)
- Vincoli cinematici (1993)

- Caratteristiche chiave (1996)
- Applicazioni industriali (1992)
- Azioni correttive (1996)

Per feature si intende una “caratteristica” di un modello 3D ottenuta attraverso diverse elaborazioni geometriche che vengono effettuate tramite estrusioni, considerando anche smussi e raccordi; di conseguenza per modellazione Feature-Based si intenderà la modellazione basata sulla creazione e la modifica di queste caratteristiche “Feature”. Il lavoro di Whitney si affida alla teoria dell’analisi delle sequenze di assemblaggio (ASA) sviluppata da De Fazio (De Fazio, et al., 1987), la quale cerca di definire delle sequenze di assemblaggio fattibili per un prodotto con una specifica architettura, analizzando tutte le possibili sequenze combinatorie; questo risulterà molto utile per assiemi con un discreto numero di parti.

La difficoltà nel determinare tutte le possibili sequenze di montaggio si può supporre che prenda origine dai seguenti fattori:

- Un alto numero dei collegamenti rispetto al numero di sottoassiemi
- Tolleranze stringenti tra le parti
- Ristretti vincoli di progettazione
- Sottoassiemi labili o labilità di alcune parti all’interno di un assieme
- Presenza di un numero discreto di sottoassiemi che compongono a loro volta un altro assieme di livello superiore e che necessitano di essere completati in maniera definitiva prima di salire con il livello di assemblaggio.

Il problema dell’individuazione e dell’analisi delle possibili sequenze di assemblaggio risale al 1964 tramite Prenting e Battaglin, i quali studiarono le sequenze di assemblaggio ottimali al bilanciamento della linea di montaggio e, a seguire, da ricercatori come Bourjault (1984), De Fazio e Whitney (1987), Gustavson (1989), Baldwin (1989), Sukhan Lee (1989) e Wilson (1992) che hanno proposto il loro metodo attraverso domande sulle precedenze, paradigmi di disassemblaggio, traiettorie libere, interferenze e direzioni di fuga. Alcuni tra i precedenti approcci sono basati sull’analisi della teoria dei grafi tramite i diagrammi di collegamento e la loro semplificazione, l’analisi sistematica delle liste dei collegamenti da cui è possibile dedurre delle relazioni sulle precedenze. Gli strumenti forniti dal Design for Assembly (DFA) che vengono principalmente usati, includono il metodo di valutazione dell’assemblabilità (AEM) di Hitachi, il sistema di valutazione della produttività di Fujitsu ed il metodo basato sull’esperienza di Boothroyd-Dewhurst.

Questi metodi vengono serviti in ausilio al progettista per semplificare al massimo il processo di assemblaggio dell'assieme, per ridurre il numero di parti che compongono il prodotto, per ridurre il materiale usato e dove possibile per unire due parti in una sola semplificandone così l'architettura del prodotto e conseguentemente la sua assemblabilità. I sistemi di pianificazione dell'assemblaggio di un prodotto assistiti al calcolatore (dall'inglese Computer Aided Assembly Planning CAAP) sono stati oggetto di numerose ricerche nell'ultimo decennio, ma nessuna di queste ha raggiunto risultati significativi nell'industria manifatturiera. Nonostante le ricerche effettuate abbiano prodotto dei risultati utili, ai fini della comprensione delle problematiche legate all'assemblaggio, la loro applicazione nell'industria è limitata dal fatto che questi richiedano lunghissimi tempi di calcolo computazionale e non sempre forniscano dei risultati ragionevoli.

Dal momento che alcuni ricercatori hanno considerato intrattabili alcune delle casistiche riguardanti il problema dell'individuazione automatica delle fasi di assemblaggio (a causa dell'elevato numero di parti che possono comporre un prodotto moderno), per un lungo periodo la ricerca si è limitata a piccole aree semplici ben note, dove la vastissima gamma di possibilità non altera i risultati sperimentali; quanto appena detto varrà anche negli argomenti trattati in questa tesi. Il primo tentativo che è stato fatto al fine di pianificare in maniera automatica, ossia tramite una metodologia univoca o un'intelligenza artificiale, una sequenza di assemblaggio valida, riguarda casistiche molto sconnesse dalla realtà industriale. Si pensi, ad esempio, alla risoluzione di un puzzle o di un giocattolo per bambini, dove l'obiettivo è quello di comporre in maniera ordinata un numero discreto di parti opportunamente sagomate. Nonostante ciò, queste ricerche hanno contribuito a fondare le basi per l'analisi delle sequenze di assemblaggio riguardo a casistiche industriali (con acronimo ASA dall'inglese Assembly Sequence Analysis). Uno tra i molteplici scopi degli studi in questa disciplina è proprio quello di individuare metodologie e tecniche utili nella pianificazione delle fasi di montaggio di un assieme a livello industriale.

Oggi, questa area di ricerca in pieno sviluppo include l'analisi di molteplici aspetti relazionati alle sequenze di assemblaggio quali (Homem de Mello, et al., 1991):

- La descrizione geometrica (Feature-Based) delle parti estratte da, o compatibilmente con un programma CAD.
- I vincoli relativi alle interferenze tra parti e utilizzo di strumenti dedicati.
- I fissaggi e le montature richieste per l'assieme.
- Gli accoppiamenti, le loro caratteristiche e le altre relazioni tra le parti in contatto.



- Le attività delle parti richieste per il montaggio.
- Le manipolazioni e le prese richieste in fase di montaggio per le varie parti.
- L'identificazione automatica dei sottoassiemi.

Buona parte di questi sviluppi vengono oggi presi in considerazione durante la progettazione di moderni e complessi prodotti industriali dal momento che, grazie al loro contributo, è possibile ricavare informazioni importanti come le relazioni di precedenza dell'assieme e le sequenze di assemblaggio desiderate.

La successiva idea di aggiungere delle caratteristiche non geometriche al modello CAD, come le informazioni riguardanti la definizione delle connessioni tra le parti, ha reso possibile la maggior parte degli studi nel campo della pianificazione assistita al calcolatore delle sequenze di assemblaggio (Computer Aided Assembly Planning CAAP) (Wesley, et al., 1980). Per quanto riguarda lo sviluppo delle sequenze di assemblaggio, Borjault (Borjault, 1984) ha stabilito un primo contributo significativo proponendo un metodo utile nella generazione di tutte le sequenze di montaggio valide di un prodotto. Questo metodo iterativo, come si vedrà meglio in seguito, richiede di rispondere a degli interrogativi sulle precedenze nei collegamenti tra le varie parti. Ad ogni modo, è stato non prima del 1987 che De Fazio e Whitney (De Fazio Thomas L. and Whitney Daniel E., 1987) hanno scoperto che il lungo e laborioso processo manuale di imposizione delle precedenze mediante l'analisi del modello, potrebbe venir potenziato automatizzando l'interrogazione sistematica dell'assieme con un software CAD. Nonostante ciò, questo processo ha fatto nascere delle ulteriori difficoltà nell'ottenere delle risposte consistenti. Di conseguenza, l'approccio comune per lo sviluppo di sistemi CAAP si è sempre e solo basato sull'interrogazione del modello CAD su caratteristiche geometriche e non geometriche delle parti al fine di ottenere una lista di sequenze valide. Delchambre ha riconosciuto però in questa metodologia un problema di consistenza, chiarendo che il metodo iterativo delle domande e delle risposte sulle precedenze è soggetto ad errori che possono portare alla generazione di sequenze non valide o spurie sfociando in una maggiore difficoltà nell'individuare quelle valide (Delchambre A., 1990). Così, l'idea di identificare in maniera automatica le sequenze di assemblaggio basandosi sulla conoscenza dei collegamenti tra le parti componenti l'assieme è stata oggetto di studi migliorativi. Detto ciò, è comunque possibile affermare che una volta individuate le relazioni di precedenza basate sul modello CAD, è teoricamente possibile ricavare da queste tutte le sequenze di assemblaggio valide per comporre l'assieme in maniera completa.

Queste sequenze potranno venir rappresentate per mezzo di uno schema ad albero che mostra tutte le molteplici soluzioni ricavate o tramite un più compatto schema ad albero dove verranno mostrate le sole transizioni di stato tra i sottoassiemi (De Fazio, et al.,1987). Queste rappresentazioni però rischierebbero di perdere di agevolezza nella comprensione per assiemi con un numero di parti considerevoli, così, nel 1990 Homem de Mello usò dei grafici AND/OR per mostrare in maniera molto più compatta e comprensibile la molteplicità di soluzioni trovate. Questo metodo di rappresentazione molto più efficace era però di difficile gestione, soprattutto per quanto riguarda la sua elaborazione e la sua modifica.

### **1.2.1 Soluzioni teoriche**

Generalmente, nel Design For Assembly (DFA) le informazioni più rilevanti, utili nello sviluppo di un nuovo metodo applicativo, sono le relazioni che persistono tra le varie parti del prodotto; una tra queste è la topologia dell'assieme. Tramite questa informazione è possibile unicamente valutare la posizione relativa tra le parti che compongono il prodotto. Questa può essere usata per risalire a quale sia la sequenza di montaggio ottima di tale assieme (Lai, et al., 2004). Come preannunciato, nelle teorie del DFA, diversi approcci sono stati proposti dai ricercatori con lo scopo di identificare le molteplici o le uniche sequenze di assemblaggio fattibili per un generico prodotto; di seguito verranno illustrati brevemente quelli più significativi, al fine di evidenziarne pregi e difetti.

#### ***Boothroyd - Dewhurst (1983)***

Nonostante il seguente metodo non riguardi direttamente le tecniche di CAAP bensì la generica progettazione destinata all'assemblaggio (DFA), verrà comunque illustrato, dal momento che questo lavoro fonda diverse caratteristiche su di esso.

La loro metodologia prende inizio con la selezione del tipo di assemblaggio considerato, che può essere manuale, semiautomatico o automatico dedicato. Bisogna successivamente valutare sei diversi fattori quali: il volume di produzione per turno, il tipo di prodotto da produrre, il numero di parti che lo compongono, il numero di parti richieste per modificarlo, il numero dei maggiori cambiamenti subiti dal prodotto durante l'intero ciclo di vita e la politica di investimento dell'azienda produttrice. Recuperando la riga e la colonna corrispondente ai 6 fattori appena enunciati attraverso una tabella empirica, viene stabilito il relativo metodo di assemblaggio ottimale. Una volta che la metodologia è scelta, il progettista non farà altro che valutare la bontà dello specifico processo di montaggio. Considerando ,a titolo esemplificativo, una linea di assemblaggio manuale, la prima attività sarà

quella di ottenere il maggior numero di informazioni del prodotto basandosi sui disegni progettuali o sul prototipo esistente (reale o virtuale). Il passo successivo consisterà nello smontare il prodotto, o di supporre un metodo per farlo, ed imporre una sequenza numerica ogni volta che una parte di esso viene tolta. Il terzo ed ultimo passaggio consisterà nel riassemblaggio del prodotto come da progetto. Una volta che il prodotto sarà completamente rimontato, dati alla mano, si analizzerà il tempo ed il costo di ogni operazione di assemblaggio delle parti e si prenderanno decisioni progettuali che potranno comportare la rielaborazione del prodotto. Una volta completata la lista delle azioni di modifica del prodotto e dei passaggi di assemblaggio, i valori parziali vengono sommati per fornire il tempo ed il costo totale di assemblaggio manuale, oltre che al numero minimo di parti necessarie per comporlo correttamente (Oh, et al., 1995).

Questo approccio presenterà fondamentalmente tre svantaggi: il primo consiste nel dover fornire al metodo dei consigli, durante i passaggi, su come affinare il progetto. Il secondo è che il progettista deve compiere lunghi e tediosi passaggi ripetitivi al fine di aumentare l'efficienza del processo implementato. Per ultimo, il progettista potrà solamente partire da un metodo di assemblaggio in precedenza selezionato e calcolarne per ogni componente l'efficienza di pianificazione. Quindi, tramite questa metodologia non sarà possibile generare delle sequenze di assemblaggio, ma si potrà solamente valutarne la qualità.

### ***Bourjault (1984)***

E' il primo metodo implementato, tramite un algoritmo dettato da una serie di regole, in grado di generare delle sequenze di assemblaggio di un prodotto specifico. Le regole vengono imposte rispondendo ad un determinato numero di domande discriminanti, riguardanti gli accoppiamenti tra le varie parti e la molteplicità di quelle presenti. Inizialmente vengono impiegate le informazioni contenute nel disegno dell'assieme e nella lista delle parti, al fine di caratterizzarlo tramite una rete composta da nodi che rappresentano le parti dell'assieme e da linee tra essi, ossia i suoi lati, che rappresentano una qualche informazione di interazione tra le stesse; generalmente questa relazione è il loro contatto fisico. Successivamente, ogni passo di montaggio è caratterizzato dalla creazione di uno o più collegamenti tra le parti dell'assieme. Così, una volta finito, l'assieme potrà essere caratterizzato fin dal principio da una stringa di numeri, che in qualche sequenza rappresenteranno i contatti creati in quel dato passaggio. Le regole che permettono all'algoritmo la generazione di una sequenza valida, derivano dalla risposta a delle domande specifiche che riesaminano gli accoppiamenti creati. Ogni risposta è intesa come una semplice risposta booleana che può essere vera o falsa.

Le domande sono generalmente di due tipologie, in particolare saranno espresse nelle seguenti forme:

- E' vero che  $L_i$  non può essere eseguito dopo che  $L_j$  e  $L_k$  siano stati eseguiti?
- E' vero che  $L_i$  non può essere eseguito se  $L_j$  e  $L_k$  non sono stati ancora eseguiti?

Dove  $L_i$  è inteso come il generico collegamento i-esimo, mentre  $L_j$  e  $L_k$  sono denominati il corpo della domanda.

Le domande, in entrambi le forme, si presentano in due fasi del processo chiamate prima e seconda fase; la seconda fase è organizzata in diversi moduli. Ad ogni iterazione, il corpo della domanda viene ridotto da un'ulteriore interrogazione dell'utente o da una regola riguardante i contatti che non siano fattibili; la sequenza parziale è così prodotta ed il processo prosegue volta per volta attraverso i moduli di livello superiore.

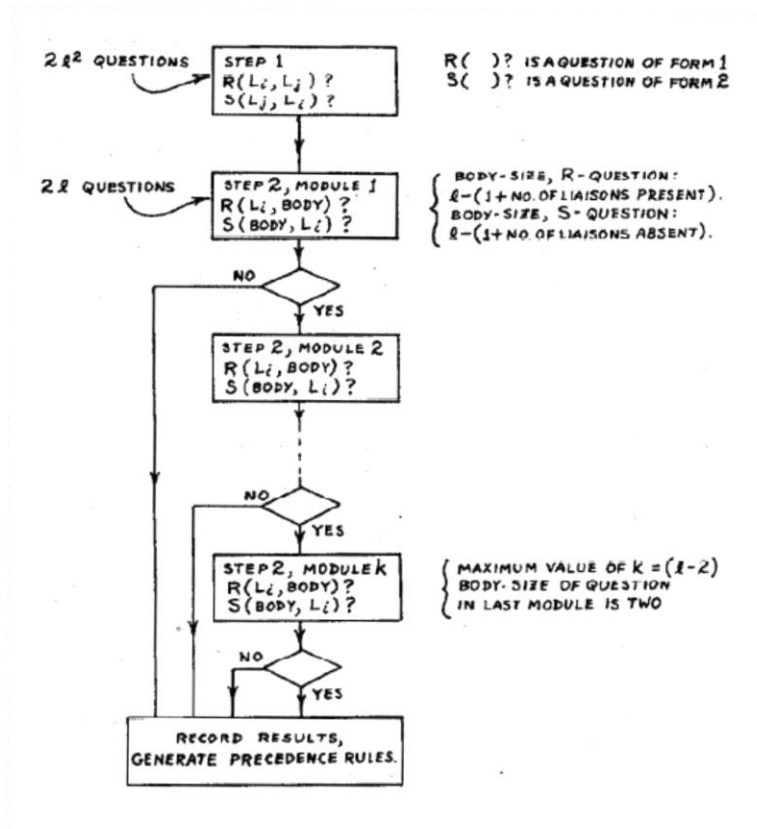


Figura 1.3 Diagramma di flusso del metodo di Bourjault (preso da De Fazio, et al.,1987).

In Figura 1.3 viene mostrato un possibile diagramma di flusso del metodo appena descritto. Le sequenze di assemblaggio derivanti, saranno così rappresentate da uno schema ad albero caratterizzato da diverse ramificazioni, dove ogni ramificazione costituirà una possibile sequenza di montaggio. Alla radice dello schema sarà presentato il sistema completamente disassemblato e salendo di livello saranno rappresentate man mano le possibili soluzioni ottenute. In questa maniera, il numero di sequenze di collegamento utili a completare l'assieme sarà pari al numero di volte che verrà rappresentato il sistema completo al suo ultimo livello. Il numero di domande  $Q$  associate all'ultimo modulo della seconda fase, dipenderà così dalla serie di risposte a tutte le domande precedenti  $l$ , ma nel contempo il numero totale di tutte le domande non potrà superare il valore massimo di  $l \cdot 2^l$ .

Ad esempio, per  $l \geq 3$ , si otterrà che il numero delle domande poste in un dato livello dovrà essere contenuto entro i seguenti limiti :

$$l \cdot 2^l > Q > 2 \cdot (l^2 + l)$$

Si può facilmente osservare che il numero di domande necessario a questo tipo di approccio è estremamente alto, in particolar modo per prodotti che presentano un numero di collegamenti pari o superiori alla decina di elementi.

### ***De Fazio & Whitney (1987)***

Con l'idea di ridurre il numero di domande necessarie da porre all'utente per generare le sequenze tramite un algoritmo, De Fazio e Whitney hanno ripreso il metodo di Bourjault e hanno modificato la tipologia di domande ed il metodo risolutivo. Entrambi i metodi sono comunque molto simili dal momento che comportano l'interrogazione dell'utente attraverso delle domande di precedenza sui collegamenti, la differenza sostanziale consiste nel fatto che, in questo metodo, l'addetto inizia con la rappresentazione dell'assieme tramite una rete di nodi e linee di collegamento tra essi. Ogni parte verrà rappresentata da un nodo ed ogni collegamento tra essi sarà enumerato e costituirà il contatto tra le parti che collega. L'utente avrà la possibilità di decidere, basandosi caso per caso, su quale siano i collegamenti validi e quali siano quelli errati. Una volta fatto ciò, lo stesso dovrà rispondere ad una coppia di domande per ogni collegamento.

Le coppie di domande a cui dovrà rispondere saranno del tipo:

1. Domanda 1: Quale collegamento deve essere fatto prima del collegamento  $i$  ?
2. Domanda 2: Quale collegamento deve essere fatto dopo aver fatto il collegamento  $i$  ?

Per  $i$  che va da 1 al numero di collegamenti  $l$ .

Le risposte dovranno venire espresse sotto forma di relazioni di precedenza tra i collegamenti oppure attraverso delle combinazioni logiche di collegamenti per via di operatori booleani. Ad ogni collegamento, come detto, competeranno esattamente due domande di precedenza. Una lista esplicita di quali collegamenti bisognerà portare a termine e quali bisognerà omettere costituirà lo stato finale dell'assieme. L'assemblaggio procederà da stato a stato, aggiungendo allo stato successivo il collegamento della parte o del sottoassieme precedentemente creato con una nuova parte non ancora considerata. La generazione delle sequenze di collegamento avverrà analizzando ciascun collegamento in relazione alle coppie di domande associate; questo viene effettuato considerando volta per volta il collegamento che non ha ancora precedenze sugli altri. In questa maniera verrà creato un diagramma di ranghi, utile alla definizione delle sequenze, caratterizzato da tanti ranghi quanti sono il numero di parti che compongono l'assieme. In Figura 1.4 viene rappresentato a titolo dimostrativo un diagramma dei collegamenti associato al relativo diagramma delle parti dove, i nodi rappresentano le parti componenti l'assieme e le corde tra i nodi rappresentano l'informazione di contatto.

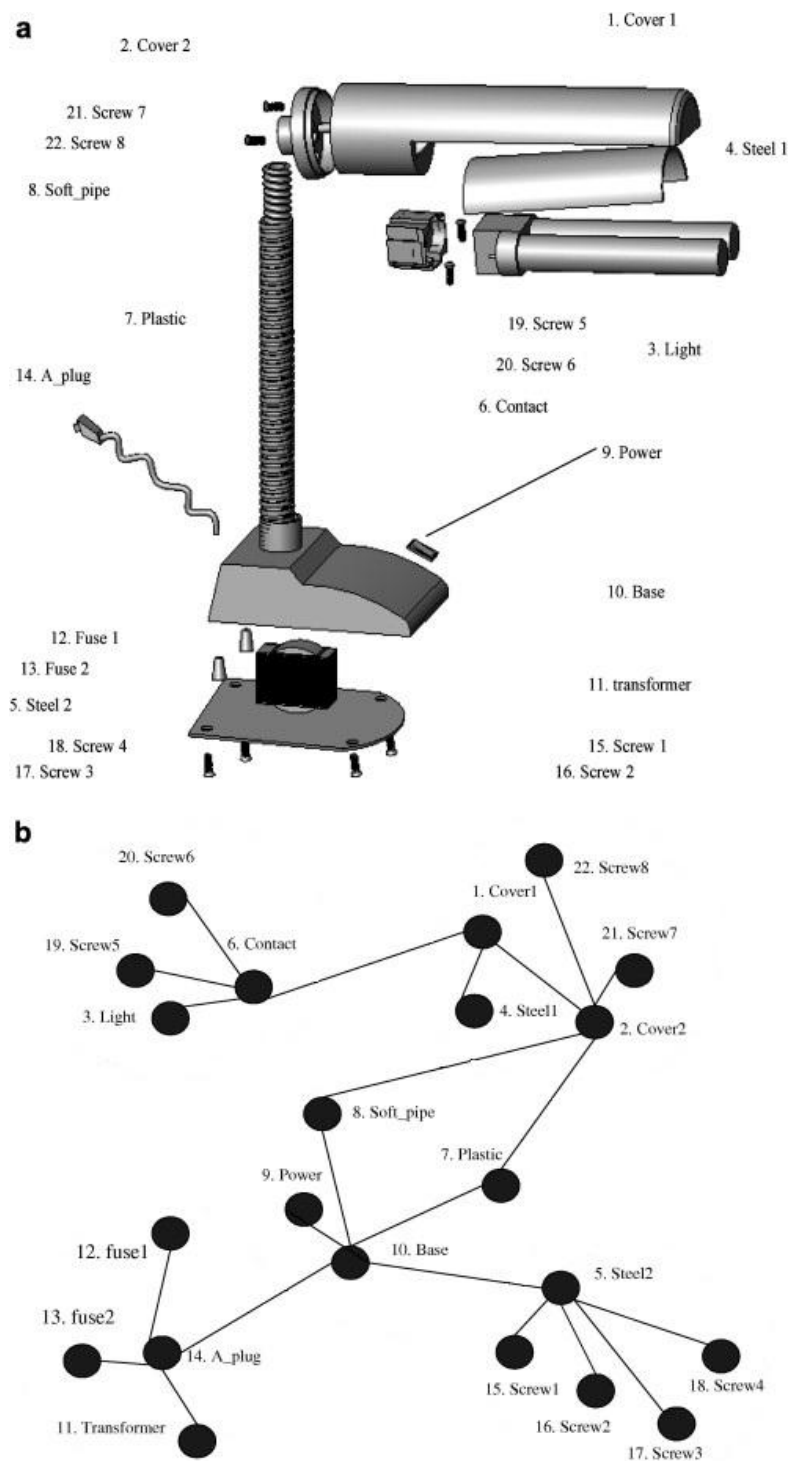


Figura 1.4 Diagramma delle parti (a) e relativo grafo non orientato di una lampada da tavolo (b) (Hwai, et al., 2008).

***Sukhan Lee (1989)***

Il presente metodo consiste nella decomposizione ricorsiva dell'assieme in dei sottoassiemi generati attraverso l'analisi degli elementi presenti nel grafo equivalente dei collegamenti. Il metodo consiste nell'unire quelle parti che, a causa di motivi di infattibilità pratica o di dipendenza dalla funzionalità, non possono essere mutuamente separate ad un determinato livello di montaggio. Quindi, viene assegnato un peso ai collegamenti del nuovo grafo, che è stato precedentemente ridotto, basandosi sulle caratteristiche di stabilità e di connettività strutturale degli stessi (come rappresentato nella Tabella 1, ricavata empiricamente) al fine di individuare le priorità di collegamento.

Tabella 1 Tabella delle connettività proposta da Sukhan Lee: stabilità relativa del collegamento basate alla tipologia di fissaggio.

<i>Tipologia di fissaggio</i>	<i>Stabilità relativa</i>
Accostamento	0.1
Incollaggio	0.3
Forzamento	0.4
Push & Twist	0.5
Vite	0.6
Connettore	0.7
Saldatura	1

La selezione del sottoassieme migliore viene effettuata attraverso una valutazione su tutti gli assieme disponibili che si basa su indici che pesano la mobilità dello stesso, le sue preferenze strutturali, la sua mobilità e il suo parallelismo all'interno della struttura. La generazione automatica delle sequenze si ottiene espandendo, dall'ultimo passo, tutti i sottoassiemi creati nei sottoassiemi o nelle parti che ne hanno dato origine.



### ***Sviluppi seguenti e considerazioni generali***

In seguito a questi, nel 1991 Baldwin propose un nuovo sistema CAAP, alternativo ai precedenti che, non usando come dati di partenza i collegamenti estratti dal modello CAD tra le parti, produce il set completo di sequenze di assemblaggio e aiuta successivamente l'utente nella scelta delle sequenze valide e di quelle ottimali, basandosi su alcuni criteri quali le difficoltà di unione tra due parti, la stabilità di ogni sottoassieme, gli orientamenti reciproci ed i sistemi di fissaggio (Baldwin et al.). Studi recenti hanno messo in luce promettenti approcci per la determinazione dell'assemblaggio basandosi sull'uso di regole euristiche per generare dei "vincoli leggeri" in aggiunta ai vincoli sull'assemblaggio classici (considerati molto più stringenti e riguardanti la geometria delle parti e la disposizione topologica). Questi cosiddetti "vincoli leggeri" sono stati classificati in vincoli di infattibilità, vincoli di orientamento, vincoli di difficoltà di assemblaggio, vincoli di instabilità del sottoassieme e vincoli di dissimilarità (Banerjee, et al.,1999). Si rimanda alla bibliografia per un approfondimento in merito.

In generale, una delle maggiori difficoltà legate all'analisi delle sequenze di assemblaggio è dovuta allo spazio delle soluzioni. Nella risoluzione di una struttura, composta da un numero discreto di parti, può sistematicamente presentarsi la possibilità di un esplosione del numero di soluzioni coerenti alle teorie ASA applicate. Diventa quindi molto difficile, se non impossibile, gestire facilmente il metodo in maniera sistematica tramite l'ausilio di un elaboratore elettronico. Proprio per questo motivo, molti dei programmi che sono stati sviluppati in questo ambito, comportano la costruzione di previsioni adatte ad eliminare alla nascita ramificazioni spurie od indesiderate dello schema ad albero cercando in questo modo di contenere lo spazio delle soluzioni entro un valore limitato. Di fatto esistono tutt'ora casistiche in cui lo smisurato numero di soluzioni trovate (dovute principalmente ad un gran numero di parti coinvolte), fa cadere in difetto il metodo di risoluzione ASA utilizzato. Questo a causa dei lunghissimi tempi di scelta di una soluzione valida tra tutte le altre.

Occorre sottolineare il fatto che il numero di conoscenze richieste per una determinazione intelligente delle soluzioni di montaggio di un assieme meccanico è estremamente grande e diversificato. Per quanto concerne l'applicazione a livello industriale delle tecniche di ASA e di CAAP, molte di quelle sopraesposte vengono raramente applicate nelle realtà aziendali, se non con molto scrupolo. Questo è in primo luogo dovuto al fatto che, oltre alle ragioni mostrate, con buona probabilità un'impresa avrà già stabilito un suo piano di produzione ed avrà già a disposizione un team valido di progettisti che svolgeranno tale compito. Oltre alla circostanza per cui

esistono comunemente delle linee di assemblaggio standard, per specifici campi di lavoro, che permettono una buona flessibilità di cambiamento della linea per diversi prodotti.

Si consideri che un metodo sistematico utile nell'individuazione delle soluzioni di montaggio può fornire, oltre alle soluzioni cercate, delle soluzioni che risulteranno non realizzabili a livello pratico nonostante siano in perfetta sintonia con le ipotesi e la teoria selettiva applicata. Queste soluzioni non realizzabili andranno scartate dalla vasta gamma di soluzioni ottenute attraverso dei criteri di selezione che si esaminerà in seguito.

Una volta note, senza specificare per il momento come, una serie di sequenze di assemblaggio fattibili, il compito del progettista sarà dunque quello di individuare tra queste quella migliore.

Tra i possibili parametri che verranno presi in considerazione durante la selezione delle sequenze di montaggio fattibili, i più discriminanti saranno:

- Il costo.
- Il tempo e la semplicità di assemblaggio.
- La presa del robot ( nel caso di montaggio automatizzato).
- Le richieste di fissaggio e posizionamento.
- Il ricambio parziale della sequenza di montaggio in caso di imprevisti.
- La disposizione della linea di montaggio.
- I possibili test di valutazione richiesti durante il processo di montaggio.
- Il tempo associato alle pianificazioni alternative.

Con lo scopo di riuscire a valutare se una tra le tante sequenze sia fattibile o meno, verranno definiti dei vincoli di assemblaggio. I vincoli di assemblaggio sono condizioni restrittive sulla possibilità di unire due parti o meno e si dividono in due categorie che sono: i vincoli operativi (riferiti a due parti o a due sottoassiemi che definiscono se una determinata operazione di assemblaggio è possibile o meno) e i vincoli strategici (riferiti alle intere sequenze che definiscono se una di queste sia conveniente o sconveniente). Un esempio di vincolo strategico può essere quello di imporre alla sequenza di passare durante le fasi di montaggio per un determinato sottoassieme, oppure può essere quello di raggruppare delle parti che devono essere necessariamente montate in sequenza. I vincoli di tipo operativo sono di tre tipologie: quelli geometrici, quelli di stabilità e quelli sui materiali (ossia sulla possibilità di usare uno strumento ausiliario di montaggio piuttosto che un altro e sulla compatibilità tra le due parti)(Homem de Mello, et al.,1991).

Nonostante i numerosi sforzi da parte dei ricercatori in questo settore, è importante ribadire che nell'odierna realtà industriale, il lavoro della pianificazione delle sequenze di assemblaggio viene ancora svolto in maniera manuale da personale esperto basandosi sull'esperienza e sulle caratteristiche salienti del progetto del prodotto e del suo ciclo di vita.

### **1.3 Obiettivo della seguente ricerca**

Ci si interroga, a questo punto, sull'opportunità di rendere la pianificazione delle sequenze di montaggio da parte del progettista la più metodica possibile, nella stessa maniera in cui si applica un approccio metodico nella progettazione generale. Per far ciò ci si appella alla creazione di strumenti di generazione automatica delle stesse che assistano il progettista durante l'intera fase di concepimento di un piano valido di montaggio. In particolare, l'idea di questa tesi risiede nel poter individuare, in maniera automatica, una gamma di diverse scelte di montaggio valide per lo stesso prodotto, in supporto alle scelte di selezione della sequenza di assemblaggio ottimale. In questo modo, uno strumento automatico di identificazione delle sequenze fattibili, che lavori di pari passo con il sistema CAD sul quale si sta progettando il prodotto, potrebbe aiutare il designer ad avere una visione più ampia del problema e, nel caso, a prendere delle decisioni utili che si riflettano in termini di risparmio di tempo e di costi sulle fasi seguenti alla progettazione. La possibilità di poter avere a disposizione un diverso numero di soluzioni d'assemblaggio nei primissimi passi dello sviluppo di un nuovo artefatto, può semplificare in maniera incisiva la sua futura fabbricazione. Questo perché una semplice modifica durante le prime fasi della progettazione, può avere un ampio impatto sulle fasi successive dell'intero ciclo di vita. Tuttavia, volendo sviluppare uno strumento in grado di ottenere autonomamente delle sequenze di assemblaggio, risulta necessario che gli stessi layout contenenti i principi risolutivi, siano modellati tridimensionalmente, in modo da poter valutare la disposizione spaziale reciproca tra i componenti. Oggigiorno, con la vasta disponibilità di calcolatori e di software dedicati alla progettazione (CAD), queste richieste sono praticamente sempre soddisfatte.

L'obiettivo che ci si prepone in questa sede, come già detto, è quello di aiutare il progettista nella scelta della migliore sequenza di assemblaggio attuabile dato un tipico assieme industriale. Lo scopo del seguente lavoro di ricerca è quindi quello di fornire al progettista una molteplicità di sequenze valide, oltre a quelle dedotte dal suo intuito e dalla sua esperienza, derivanti dall'applicazione di metodologie ASA applicate sistematicamente. Tuttora, infatti, non esiste uno strumento automatico implementato al calcolatore in grado di identificare autonomamente ed in maniera corretta rispetto alla

realtà costruttiva tutte le possibili sequenze di assemblaggio fattibili di un generico assieme. Diverse sono le difficoltà che si incontrano volendo ottenere un set di sequenze di montaggio valide, alcune di queste possono venir riassunte come segue (Homem De Mello, et al.,1990):

### **1. La rappresentazione dell'assieme**

La scelta di quali informazioni considerare dell'assieme è di fondamentale importanza ai fini di una corretta analisi del problema. Inoltre è necessario decidere come queste vengano rappresentate attraverso il computer.

La forma delle parti, le relazioni geometriche tra queste, le tolleranze e qualunque altro aspetto geometrico e di contatto dovrebbe venir preso in considerazione affinché non si perdano caratteristiche salienti che potrebbero influenzare le precedenze di montaggio.

### **2. La rappresentazione del " piano di assemblaggio"**

Oltre alle tecniche risolutive, è importante decidere come esprimere il piano di montaggio ottenuto: tramite grafi direzionali, relazioni di precedenza, condizioni di posizione reciproca, grafici con condizioni booleane. Una chiara comprensione di queste alternative è molto importante ai fini dello sviluppo di una buona procedura.

### **3. La correttezza e la completezza del processo di pianificazione**

La metodologia creata non deve necessariamente sviluppare tutte le casistiche possibili, quello che conta è che il metodo risolutivo abbia le potenzialità necessarie ad ottenere una buona parte del campo di soluzioni.

### **4. La generazione di sequenze di assemblaggio fattibili**

In quanto le sequenze che risulteranno non fattibili ai fini pratici non desteranno nessun successo ai fini industriali, dal momento che non forniranno nessun valore aggiunto nella semplificazione del lavoro del progettista.

### **5. La selezione dei piani di assemblaggio**

Trovare un metodo sistematico che individui quale sia la sequenza ottimale di montaggio è un annosa questione che merita altrettanto interesse. Esistono sostanzialmente due diversi approcci al problema preposto: il primo, più qualitativo si limita a verificare se ci siano delle operazioni di montaggio dettate dalla sequenza che richiedano uno sforzo maggiore mentre il secondo, non si limita a questo ma dà un peso quantitativo all'onerosità di ogni singola operazione ed al costo che ne potrebbe derivare in termini economici.

## **6. L'efficienza del processo di pianificazione**

La massima efficienza della metodologia applicata si ottiene quando nessuna delle sequenze valide venga trascurata. Questo sempre entro i limiti del costo computazionale del problema dal momento che il tempo di elaborazione gioca un ruolo fondamentale nelle tempistiche di consegna di un'azienda.

## **7. L'integrazione con programmi CAD**

L'utilità del metodo risiede proprio in questo, in quanto uno degli obiettivi è quello di estrapolare dal software informazioni utili al fine di creare in maniera semi automatica solo le corrette sequenze di assemblaggio.

## **8. L'integrazione con i progettisti delle fasi e delle tecniche di montaggio**

Il solo piano di assemblaggio può essere insufficiente se non accompagnato da delle descrizioni pratiche sui metodi di assemblaggio. Si pensi ad esempio alla generale asimmetria dei componenti, che determina molto spesso un'univocità nell'orientamento della parte in fase di assemblaggio.

Per quanto riguarda la descrizione geometrica delle parti estratte da un modello CAD, ma soprattutto la sua analisi al fine di cercare informazioni utili all'estrapolazione di possibili sequenze di assemblaggio dell'assieme, le Feature su cui si basa il modello disegnato non sono del tutto univoche dal momento che è possibile raggiungere il medesimo risultato attraverso caratteristiche geometriche differenti. Un esempio utile ad esplicitare meglio questo concetto può essere il fatto che è possibile ottenere il modello al CAD di un cilindro cavo attraverso diverse modellazioni Feature Based; ad esempio è possibile partire da un cilindro pieno e rimuovere per estrusione la parte interna che deve essere cava, ma allo stesso tempo è possibile partire da una corona circolare ed estruderla ottenendo il medesimo risultato, ma attraverso caratteristiche di modellazione differenti.

Una volta individuato un set di sequenze possibili, derivanti da una delle tecniche di analisi ASA e dalla corrispettiva metodologia di implementazione al computer CAAP, si vuole che quest'ultime siano tutte le sequenze di assemblaggio utili a montare completamente il prodotto. Lo scopo è quello di individuare tra tutte le soluzioni ottenute, quella che nella maniera più semplice e veloce possibile permetta il corretto montaggio dell'assieme.

In base a quanto detto , viene proposta l'implementazione di uno strumento di pianificazione automatico delle sequenze di assemblaggio, che fonda le sue scelte esclusivamente sulle informazioni topologiche dell'assieme senza l'intervento di un esperto (R. Vigano and G.O. Gomez,2011). Lo strumento di ausilio che si vuole creare, dovrà essere in grado di sviluppare delle sequenze di assemblaggio che non dipendano da parametri industriali e che potranno essere facilmente modificate e inserite in un contesto aziendale specifico.

# **CAPITOLO 2**

## **APPROCCIO ALLA DEFINIZIONE DELLE SEQUENZE D'ASSEMBLAGGIO**

### **2.1 Tecniche di generazione delle sequenze**

Si vuole, in primo luogo, sviluppare un processo automatico in grado di ottenere le informazioni topologiche di un'assieme tipicamente industriale necessarie nella definizione delle sequenze di assemblaggio partendo dal suo modello CAD tridimensionale. Questo approccio preliminare è nettamente differente dalle tecniche tradizionali di analisi delle sequenze di assemblaggio ASA fin'ora sviluppate, dal momento che non verranno poste delle domande sulle precedenze di connessione e non verrà richiesto l'intervento di uno o più esperti. Proprio per questo proposito è possibile che il metodo non identificherà le sole sequenze di montaggio fattibili, ma ingloberà tra le soluzioni generate un determinato numero di sequenze, coerenti alla teoria, non valide nella pratica. In un secondo momento, rifacendosi a domande discriminanti sulle precedenze (ossia alle tecniche trattate nella bibliografia) sarà possibile ridurre sistematicamente il numero, fino ad ottenerne almeno una valida tra tutte quelle ottenute. Le sequenze di assemblaggio così generate, avranno origine esclusivamente da informazioni riguardanti i collegamenti reciproci tra le parti, e la validità delle informazioni usate potrà essere valutata attraverso successivi parametri tecnologici legati al tipo di processo di produzione considerato o attraverso la valutazione di parametri legati all'utente, come l'ergonomia o le preferenze soggettive individuali.

Con lo scopo di motivare la validità del metodo automatico preso in considerazione, verranno successivamente rappresentati diversi approcci gerarchici di individuazione delle sequenze di assemblaggio; partendo dal massimo numero di combinazioni possibili che si possono avere tra un

numero discreto di parti, fino a mostrare la tecnica proposta in questa tesi per la generazione di un numero di sequenze possibili.

Le difficoltà tecniche derivanti dalla generazione autonoma delle sequenze di assemblaggio possono, come già detto precedentemente, essere riassunte nelle seguenti classi (R. Viganò and G. O. Gomez, 2011):

- la rappresentazione degli assiemi.
- La rappresentazione delle sequenze di assemblaggio.
- L'efficienza del processo di pianificazione.
- La selezione dei piani di assemblaggio.

Di seguito verrà data una motivazione alle problematiche riscontrate in questi quattro argomenti e le possibili modalità risolutive attuate per essi.

### 2.1.1 Strategia di implementazione

#### *La rappresentazione degli assiemi*

In generale, si può affermare che un assieme meccanico modellato tramite 3D-CAD presenta 2 possibili livelli di rappresentazione che possono venir riassunti come segue:

1. informazioni sui contatti tra le parti componenti l'assieme.
2. Informazioni sulla geometria completa di ogni parte.

In questa sede si suppone che solo il primo livello di rappresentazione, ossia le informazioni di contatto tra le parti, sia necessario e sufficiente al fine di comporre in maniera compiuta un piano di assemblaggio. In un modello 3D-CAD tutte le parti che compongono il sistema sono disposte nella propria posizione di equilibrio stabile. Da questo modello, il generico software di modellazione sarà in grado di identificare le relazioni di contatto tra le parti analizzando dove sussistono delle tolleranze globali pari a zero o delle interferenze e creando per ognuna di queste una stringa testuale del collegamento contenente il nome delle due parti che creano il contatto. Si ottiene come risultato una lista di tutti i contatti presenti all'interno dell'assieme in considerazione come mostrato in Figura 2.1e in Figura 2.2. Da questa lista di collegamenti è possibile successivamente creare un grafo non orientato, considerando ogni componente come se fosse un nodo e ogni collegamento tra gli stessi tramite un arco non direzionato tra due nodi. Una buona rappresentazione supplementare, è usare una matrice binaria e simmetrica di adiacenza dove viene posto un valore pari ad uno in corrispondenza della presenza di un contatto ed altrove un valore nullo.



Come asserito da De Fazio (De Fazio, et al., 1987), le informazioni di contatto tra le parti che includono forze di incastro, filettature, adesioni ed ogni altra caratteristica al di fuori della mera informazione di contatto non verranno considerate in questa trattazione. Questa caratteristica porta il notevole vantaggio che nessuna conoscenza tecnica dell'insieme viene richiesta a colui che sviluppa il modello virtuale.

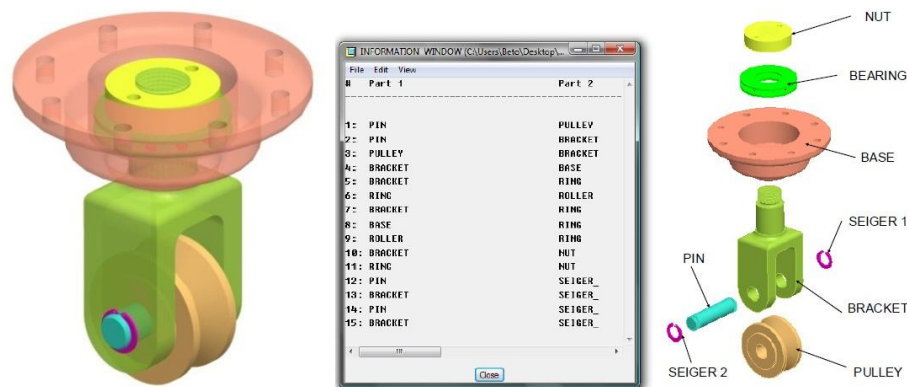


Figura 2.1 Modello 3D CAD dell'insieme e relativa lista dei contatti.

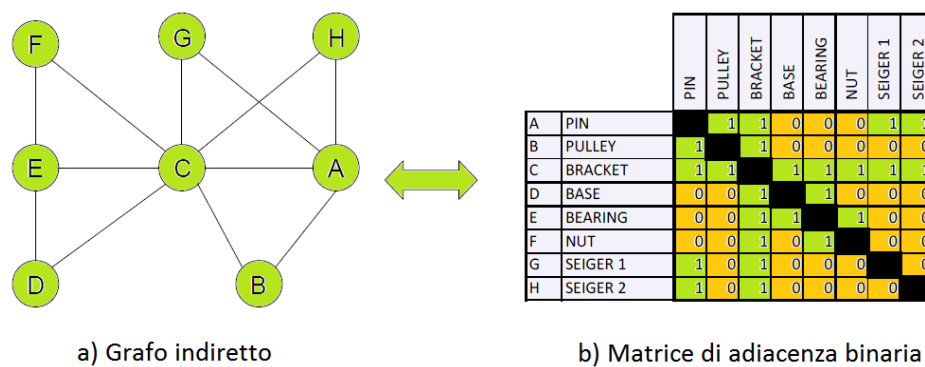


Figura 2.2 Matrice di adiacenza e grafo non orientato ricavato dal modello CAD.

Il software di modellazione preso come riferimento per la seguente ricerca è PTC Creo<sup>®</sup>, grazie ad esso è infatti possibile ricavare automaticamente le condizioni di contatto tra le parti, dal momento che al suo interno è integrata un'applicazione dedicata a questo tipo di analisi. Ad ogni modo è possibile ampliare l'utilizzo a qualunque altro software di modellazione.

Grazie alla semplicità intrinseca delle informazioni utilizzate, che non riguardano proprietà dimensionali o di materiali usati, l'approccio sviluppato risulta molto utile anche nella generazione di sequenze di montaggio nelle fasi concettuali o di massima partendo da un layout 3d di principi risolutivi identificati, dove nessuna caratteristica tecnica e costruttiva è stata ancora definita con esattezza.

### ***La rappresentazione delle sequenze di assemblaggio***

Le sequenze di assemblaggio ottenute dal software sviluppato in questa trattazione, non sono altro che una lista composta dal nome delle parti componenti l'assieme; queste sono disposte in maniera gerarchica dalla prima all'ultima e la loro lettura va intesa come delle istruzioni dove l'utente deve iniziare il montaggio dell'assieme partendo dalla prima parte della lista, collegando ordinatamente volta per volta tutte le successive, fino ad arrivare all'ultima presente. Dal momento che il metodo informatico applicato richiede la formazione di sottoassiemi per lo sviluppo della soluzione, si sottolinea la possibilità di poter esprimere le sequenze sia in maniera esplicita, ossia composte dalla totalità delle parti prese in considerazione, che in maniera implicita (molto più compatta), ossia composte dai nomi dei sottoassiemi creati volta per volta.

Ad ogni modo, una volta noto il set di sequenze ottenute, ogni qualunque altra modalità di rappresentazione può essere implementata nel software come, ad esempio, attraverso grafici direzionali, grafici AND/OR oppure attraverso qualunque altra rappresentazione simbolica che possa essere utile al paragone con i risultati ottenuti attraverso i diversi metodi.

### ***L'efficienza del processo di pianificazione***

Come più volte affermato, la generazione delle sequenze di assemblaggio presentata in questo metodo è molto differente dal caso teorico; infatti non viene richiesto a personale esperto di dare risposta ad un determinato numero di domande al fine di ottenere tutte le possibili sequenze fattibili, ma viene generata una serie di sequenze di assemblaggio, derivanti dal grafo equivalente dell'assieme, in maniera del tutto autonoma.

Nell'approccio sperimentato in questa sede, l'idea è dunque quella di riuscire a trovare delle sequenze di assemblaggio fattibili e non, senza l'ausilio dell'intervento umano, ma semplicemente basandosi sulle sole informazioni topologiche dell'assieme. Quello che conta non è ottenere tutte le eventuali possibilità risolutive dell'assieme, ma avere le potenzialità di generare un insieme di soluzioni che contenga come minimo una sequenza fattibile. In

questo modo, se non si ottenesse neanche una soluzione valida significherebbe che l'assieme considerato non abbia nessuna possibilità di venir montato correttamente. Una possibile causa può essere dovuta a degli errori presenti nella progettazione, in tal caso risulta necessario ricontrollare l'intero progetto al fine di individuare delle possibili instabilità tra le parti o dei vincoli di iperstaticità imposti.

### ***La selezione dei piani di assemblaggio***

Trovare un metodo sistematico per restringere il campo delle possibili alternative trovate è un punto cruciale nella pianificazione di un assemblaggio. In accordo con Homem de Mello ( Homem de Mello, et al., 1991), esistono due tipi di approccio presi in considerazione. Uno, più qualitativo, è lo sviluppo di regole che possono venir usate per eliminare le sequenze che contengono delle operazioni difficili o dei sottoassiemi "scomodi" da mettere in atto. Un altro approccio, molto più quantitativo, sarà quello di implementare una funzione di stima che valuti il merito del piano di assemblaggio in base, ad esempio, al costo delle risorse richieste per completare l'assieme, al tempo totale di montaggio oppure alla difficoltà di esecuzione di una delle operazioni. Lo sviluppo di un approccio automatico corrisponde alla prima classe descritta, anche se le regole non derivano dall'interrogazione del personale esperto come spesso accade, bensì dall'interrogazione sistematica (in maniera automatica) delle informazioni di contatto ricavate dal modello virtuale dell'assieme considerato. Si suppone, in questo contesto, che le informazioni di contatto ottenute dal modello virtuale e la relativa matrice di adiacenza binaria siano sufficienti per la generazione di almeno una sequenza fattibile. Questo può essere dedotto osservando le soluzioni ottenute attraverso l'applicazione di altri metodi convenzionali presenti in bibliografia, dove si denota una notevole dipendenza tra le adiacenze delle parti all'interno delle sequenze ottenute rispetto alle connessioni presenti.

## **2.2 Ipotesi e considerazioni preliminari**

Al fine di comprendere in maniera esaustiva il metodo sistematico applicato in questa sede, bisogna innanzitutto porre delle condizioni sulla sua validità, in particolare specificando quali siano le ipotesi preliminari che consentano di ottenere un risultato coerente e spiegando come valutare tale risultato con criticità; di seguito, a tal proposito, verranno enunciate alcune considerazioni basilari.

Innanzitutto, il metodo proposto tratta principalmente sia prodotti in fase di sviluppo che prodotti già ingegnerizzati, in generale prodotti di natura

meccanica composti da un numero discreto di parti; in linea teorica, sarebbe comunque possibile espandere l'implementazione ad altri campi di utilizzo, come ad esempio lo sviluppo di layout di impianti industriali. L'idea di partenza è quella di considerare ogni parte costituente l'assieme come un nodo e l'informazione di contatto tra esse come un arco tra due nodi (R. Viganò and G. O. Gomez, 2011). Nel caso più componenti siano in contatto reciproco, può accadere che gli archi creino un percorso chiuso che costituirà un ciclo.

Considerando un generico assieme meccanico, da un punto di vista topologico si suppone che la sua assemblabilità, oltre a dipendere dall'esatta forma e dimensione dei componenti, dipenderà anche dalla modalità con cui questi verranno montati assieme. In questo modo, la funzionalità del sistema meccanico potrà essere garantita attraverso due differenti tipologie di interazione: le interazioni interne (tra le parti) e le interazioni esterne (tra il sistema, l'ambiente circostante e l'utente); quest'ultime non verranno prese in considerazione nel grafo dei collegamenti e quindi non prenderanno parte a questa ricerca. In tal modo si può supporre che le soluzioni generate siano corrette da un punto di vista topologico, mentre la loro esattezza a livello quantitativo dovrà essere comunque valutata in separata sede, ad esempio attraverso la simulazione con prototipi reali e virtuali.

L'approccio applicato non considererà la tipologia di linea di assemblaggio usata, che sia essa automatica, semiautomatica o manuale. Chiaramente la valutazione delle sequenze valide nei primi due casi sarà più lunga e complessa rispetto al caso manuale, dove la possibilità di poter montare con mano due parti conferisce una maggiore flessibilità nella selezione dei risultati.

Al fine di semplificare la comprensione di quanto ottenuto e con lo scopo di ridurre il costo computazionale impiegato, elementi di fissaggio come viti, bulloni e rivetti o anche saldature ed incollaggi non verranno considerati nel modello 3D-CAD di partenza usato per la valutazione. Questa carenza potrà ad ogni modo essere colmata considerando alla stessa maniera gli elementi che compongono "l'assieme fissaggio" come un nuovo sottoassieme da elaborare in separata sede. Il metodo automatico impiegato fornisce come soluzione una serie di sequenze che contengono la sola informazione riguardante l'ordine di montaggio delle parti dell'assieme. Infatti le sequenze generate non conterranno informazioni riguardanti le tecniche di assemblaggio e neppure informazioni riguardanti la disposizione specifica tra le parti durante ogni singola attività di montaggio.

Le sequenze individuate saranno delle sequenze lineari, ogni attività di assemblaggio verrà intesa come il montaggio di una ed una sola parte ad un'altra, che sia essa una parte semplice o un sottoassieme ricavato durante il

procedimento. Inoltre, per la stessa ipotesi di linearità delle sequenze, ogni parte non potrà avere più di un collegamento con le altre: ad esempio, se in un ciclo composto da  $k$  collegamenti,  $(k-2)$  collegamenti saranno già stati eseguiti, la riduzione di un ulteriore collegamento comporterà la riduzione dell'intero ciclo ad un semplice collegamento. Si suppone inoltre che, considerando i collegamenti totalmente rigidi, non verranno prese in considerazione caratteristiche come vincoli cedevoli o deformabilità delle parti.

Il metodo utilizzato non è altro che una riduzione graduale di cicli e nodi pesati, quindi una volta che un nodo o un ciclo verrà semplificato, generando un nuovo sottoassieme, quel sottoassieme rimarrà tale e non potrà più essere scomposto nelle sue parti costituenti. Una volta che il sottoassieme sarà creato, tutte le sue parti avranno un ordine di composizione che non potrà più essere alterato. I componenti, ossia i nodi, con un unico contatto verranno inglobati al proprio nodo adiacente formando un nuovo sottoassieme. Questo nuovo sottogruppo o sottoassieme e sarà considerato come un nuovo nodo e sarà necessario ricalcolare la matrice di adiacenza al fine di identificare il prossimo passaggio di riduzione. Il procedimento verrà ripetuto finché l'intero sistema non verrà ridotto ad un unico elemento.

### 2.2.1 Tutte le soluzioni possibili

Al fine di poter considerare tutte le possibilità combinatorie di un dato assieme meccanico, si consideri inizialmente come informazione il solo numero di parti che compongono il generico sistema meccanico; qualsiasi altra relazione verrà trascurata. Così, l'assieme meccanico potrà venir considerato come un semplice set di elementi che possono essere disposti secondo diverse sequenze lineari, costruite usando uno alla volta tutti gli elementi a disposizione. Questa caratteristica corrisponde all'esatto opposto di avere una sola sequenza ordinata composta da tutti gli elementi presenti, ed in particolare, corrisponde ad avere il numero massimo di permutazioni che è possibile fare partendo dalla generica sequenza ordinata.

Nel calcolo combinatorio, per definizione, il numero massimo di permutazioni che è possibile fare su una sequenza ordinata lineare, composta da  $n$  elementi, è pari a  $n!$  fattoriale.

Ad esempio, un sistema meccanico composto da 5 elementi può venir ordinato in  $5!=120$  modalità differenti; questo trascurando qualsiasi altra informazione al di fuori dello stesso numero di componenti. Una analogia molto stretta per comprendere meglio il significato, è quella di considerare dei mattoncini giocattolo tutti uguali di tipo brick-type (come mostrato in Figura 2.3), con la stessa forma e geometria, che possono venir montati uno

con l'altro in qualunque ordine senza nessun vincolo di precedenza. Generalmente è molto raro trovare una condizione simile in sistemi meccanici tipicamente industriali. Se si vuole rappresentare un sistema meccanico sotto forma di grafo, bisogna considerare che questo presenterà dei contatti tra le parti, quindi dovranno esistere dei collegamenti tra i nodi che riportino questa informazione (Diestel, 2005).

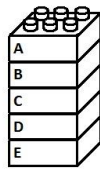


Figura 2.3 Assieme brick-type.

Quindi per un assieme meccanico composto da  $n$  parti, il numero massimo di sequenze, uguale a  $n!$ , corrisponde ad un grafo completo dove ogni parte sarà collegata a tutte le altre che costituiscono il sistema, come rappresentato in Figura 2.4

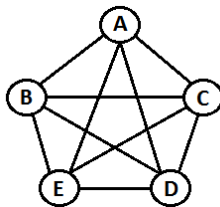


Figura 2.4 Grafo completo composto da 5 elementi  $K^5$ .

D'altro canto, il numero minimo di sequenze che è possibile ricavare sullo stesso assieme, sarà indipendente dal suo numero di parti e sarà pari a 2; questa situazione corrisponde ad un grafo lineare, come rappresentato in Figura 2.5, dove esisteranno solo due possibili sequenze che si otterranno partendo da un nodo all'estremità fino ad arrivare al nodo sull'altra estremità e viceversa.

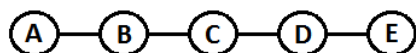


Figura 2.5 Grafo lineare composto da 5 elementi.

Nella Figura 2.6 viene rappresentato il numero di sequenze possibili, sia massimo che minimo, che un sistema meccanico può presentare al variare del numero di parti che lo compongono.

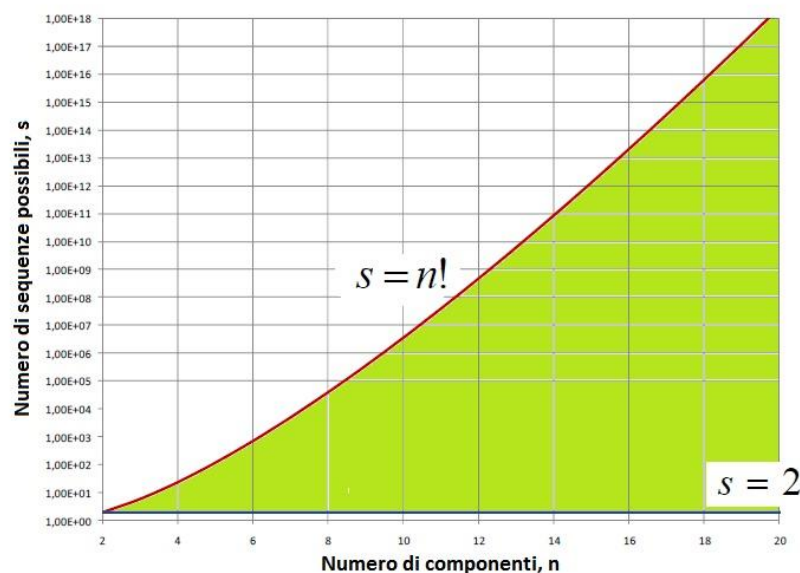


Figura 2.6 Numero di sequenze possibili al variare del numero dei componenti.

L'area inclusa tra queste due curve rappresenterà l'intero campo delle possibili soluzioni di assemblaggio. Questo approccio nonostante da un punto di vista meccanico non fornirà nessun supporto all'analisi e alla creazione delle reali sequenze di assemblaggio, sottolinea la vasta molteplicità di soluzioni combinatorie che si possono ottenere. Così, si mette in evidenza la necessità di trovare un criterio utile a ridurre il campo di soluzioni della ricerca effettuata.

### 2.2.2 Riduzione dei collegamenti

Una volta nota la totalità delle soluzioni che è possibile ottenere considerando il solo numero di componenti che fanno parte dell'insieme, la prima riduzione del campo di soluzioni risulterà necessariamente quello di considerare il numero di collegamenti in gioco presenti nel sistema considerato, ossia il numero dei collegamenti del grafo equivalente. Così, dalla totalità di sequenze ottenute precedentemente, vengono innanzitutto

scartate quelle in cui sono presenti dei collegamenti che non esistono nel assieme reale, e quindi nel grafo equivalente.

Questa riduzione è corretta da un punto di vista meccanico solo se i componenti o i sottoassiemi valutati siano considerati rigidi durante l'assemblaggio (come già osservato nelle considerazioni preliminari). Ad ogni modo, il numero di sequenze ricavate con questo approccio dipenderà esclusivamente dal numero di collegamenti presenti e non dalla loro disposizione nel grafo; inoltre, il numero di soluzioni sarà ancora una volta troppo elevato per gli obiettivi preposti. Quindi la definizione di una regola che si basa sul solo numero di componenti e di collegamenti presenti non porterà a nessun risultato concreto, se non a circoscrivere ulteriormente il campo risolutivo. E' comunque molto utile, al fine di comprendere l'argomento, capire quale possa essere la relazione che lega il numero minimo e massimo di collegamenti presenti in un grafo in funzione del numero di parti che lo compongono. Come rappresentato in Figura 2.7, dato un numero  $n$  di parti che compongono l'assieme, il massimo numero di collegamenti corrisponderà ad un grafo completo come visto precedentemente, mentre il numero minimo corrisponderà al numero di collegamenti necessari per unire tutte le  $n$  parti una volta sola creando un grafo lineare, ossia con  $(n-1)$  collegamenti.

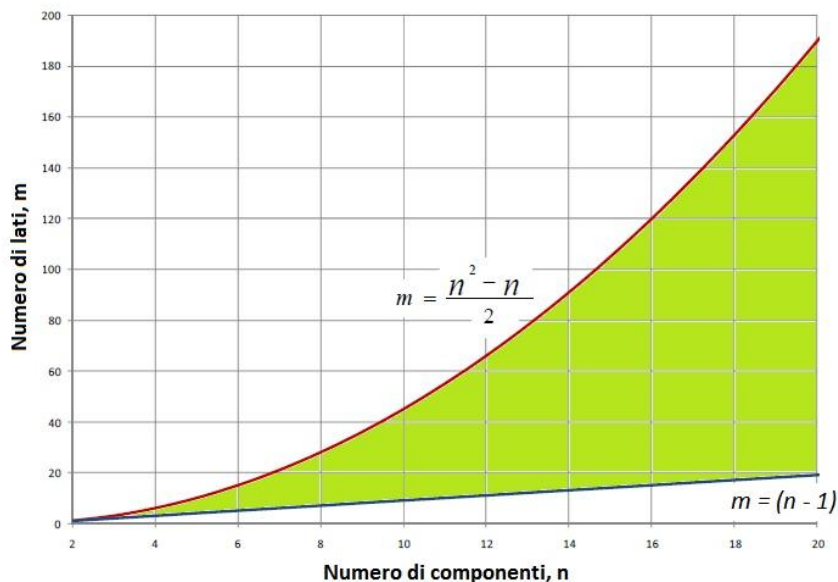


Figura 2.7 Numero di lati minimi e massimi in funzione del numero di componenti.



Una riduzione delle possibili soluzioni analoga a questa è illustrata da Lai ed altri ricercatori (Lai, et al., 2004), dove partendo da un grafo completo, e recuperando tutte le informazioni di progettazione del prodotto, si trasformano i collegamenti in relazioni di precedenza di assemblaggio basandosi sulle forme geometriche delle parti e sulla disposizione topologica dell'assieme. Alcuni metodi, come questo d'altronde, impiegano differenti indici numerici, che richiedono la valutazione da parte di personale esperto al fine di dare un corretto valore agli stessi.

### 2.2.3 Considerazioni sui grafi

Dal momento che differenti teorie ASA richiamano l'applicazione dei grafi con lo scopo di ottenere informazioni quali relazioni di precedenza, si presume sia utile trattare, almeno in parte, la teoria dei grafi e degli algoritmi applicati ad essa.

Le relazioni tra le parti che compongono il layout di un prodotto possono venir rappresentate attraverso un grafo dove ogni componente costituisce un nodo (o vertice) enumerato ed ogni interazione di contatto è rappresentata da un lato del grafo che congiunge i vertici. Per un sistema in cui le precedenze di montaggio siano già state identificate, lo stesso grafo può venire integrato da delle direzioni dei lati espresse da dei vettori. Questo grafo completo delle direzioni preferenziali sui lati prenderà il nome di grafo orientato, mentre lo stesso grafo privo di queste informazioni prenderà il nome di grafo non orientato; la Figura 2.8 mostra un esempio di entrambi.

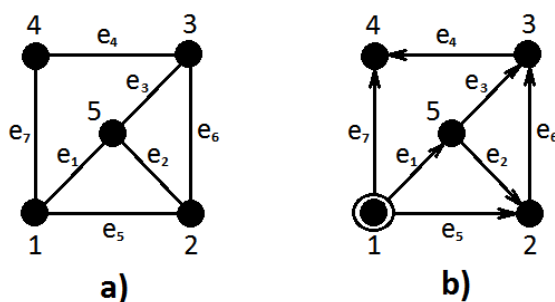


Figura 2.8 (a) Grafo non orientato, (b) Grafo orientato

L'assieme meccanico può venir rappresentato da un grafo  $G$ , che generalmente si esprime nella forma  $G = (V, E)$  dove  $E \subseteq [V]^2$ . Gli elementi  $V$  prenderanno il nome di vertici (o nodi) del grafo  $G$ , mentre gli elementi  $E$

prenderanno il nome di lati (o archi) del grafo  $G$ . Con il termine in maiuscolo  $E$  o  $V$  si vuole indicare l'insieme totale dei vertici e dei lati presenti nel grafo, mentre con il termine minuscolo  $e$  o  $v$  si vuole indicare il generico elemento considerato. Il grado  $d(v)$  di un vertice  $v$  sarà pari al numero di elementi vicini allo stesso, la serie di vicini ad un vertice  $v$  verrà denotato dalla dicitura  $N(v)$ . All'interno di un grafo è possibile delineare un ciclo percorrendo con un cammino chiuso i lati tra i vertici; la lunghezza del ciclo così indicato, sarà data dal numero di vertici (che in questo caso equivale al numero di lati) percorsi. Il ciclo di lunghezza  $k$  prenderà il nome di ciclo  $k$ -esimo e verrà generalmente indicato con l'espressione  $C^k$ . Oltre alla semplice informazione di contatto, i lati del grafo potranno essere accompagnati da qualsiasi altra informazione che riguarda delle relazioni esistenti tra le parti. Tramite l'uso di pesi o di numeri caratteristici è possibile rappresentare diverse tipologie di parametri quali precedenze di contatto, connettività e manipolazioni reciproche.

Ad ogni modo, qualsiasi dato che esula dalla semplice informazione di contatto verrà trascurata ai fini della sola generazione delle sequenze di assemblaggio (R. Viganò and G. O. Gomez, 2011). Questo perché si è supposto che l'approccio applicato richieda di partire dal semplice grafo non orientato, che è molto più generico da un punto di vista strettamente progettuale. In un tipico grafo non orientato rappresentante un sistema meccanico, tutti i componenti presenteranno come minimo una connessione. All'interno di un grafo non orientato sarà possibile definire dei percorsi che passino per i nodi percorrendo i lati presenti; diverse tipologie di percorsi caratteristici potranno così venir definiti, ma sostanzialmente tutte queste ricadranno in due categorie principali: i percorsi Euleriani ed i percorsi Hamiltoniani. Un tragitto di tipo Euleriano è un tragitto sul grafo in cui tutti i lati considerati verranno percorsi al massimo una volta sola, di contro un tragitto di tipo Hamiltoniano è un tragitto in cui ogni vertice viene visitato una volta sola (Diestel, 2005)(Gross, et al., 2006)(MacKenzie, et al., 1993). La Figura 2.9 mostra entrambe le tipologie di cammino che è possibile inscrivere su di uno stesso grafo non orientato. La maggior parte dei problemi di tipo Euleriano, derivanti dagli omonimi percorsi, ammettono una soluzione che è ricavabile sistematicamente attraverso l'uso di algoritmi polinomiali basati sul peso dato ai lati. Questi algoritmi derivano da una semplice caratterizzazione dei grafi euleriani in termini di grado del vertice. In netta contrapposizione, non esiste una tecnica risolutiva riconosciuta che fornisca una soluzione univoca ad un grafo caratterizzato da un percorso Hamiltoniano, quindi i metodi applicati per la loro risoluzione richiederanno diversi sforzi e non sempre porteranno ad una soluzione valida.

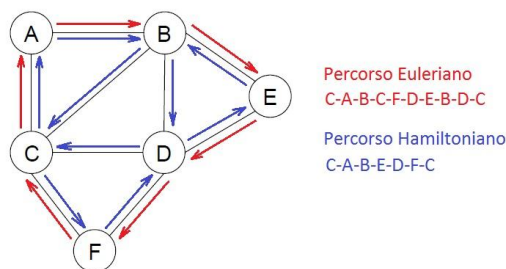


Figura 2.9 Percorso Euleriano e percorso Hamiltoniano

Ad ogni modo, esistono degli algoritmi sviluppati dai ricercatori che sono molto utili nella ricerca di questi cicli o percorsi. Ne sono un esempio il BFS (dall'inglese Breadth First Search) ed il DFS (dall'inglese Depth First Search). In accordo con le considerazioni appena fatte riguardo la rappresentazione attraverso un grafo di un assieme tipicamente meccanico, si suppone che sia sufficiente visitare ogni componente una volta sola, garantendo così la completezza del processo di assemblaggio; in questo modo tutti i collegamenti identificati nel grafo saranno rispettati anche se non saranno direttamente chiamati attraverso un grafo trasversale o una sequenza lineare. Per questo motivo il metodo che verrà implementato e analizzato farà riferimento solamente a percorsi di tipo Hamiltoniano e non di tipo Euleriano. Come si può facilmente notare osservando un generico grafo ricavato da un'assieme meccanico succede molto spesso che ci sia la presenza di nodi di grado unitario, ossia collegati solamente ad un altro elemento. La presenza di questi nodi è una condizione restrittiva nella ricerca dei percorsi Hamiltoniani in quanto imporrà la necessità che il percorso passi anche per quel dato nodo. Di contro, la riduzione ad un unico elemento di questi nodi al loro vicino, permetterà l'esistenza e quindi l'individuazione di un maggior numero di percorsi di tipo Hamiltoniano. Così, il metodo sviluppato per la generazione delle sequenze di assemblaggio presenterà come primo passaggio di iterazione, la riduzione di tutti i nodi di grado unitario presenti nel grafo. In questa maniera, ogni volta che si presenterà un nodo di grado semplice, questo verrà ridotto e sul nuovo grafo corrispondente sarà possibile applicare la ricerca dei percorsi Hamiltoniani con maggior semplicità. Sfortunatamente, dalla validazione del metodo attraverso alcuni casi test, si vedrà che questo metodo non risulta molto idoneo nella generazione delle sequenze, dal momento che molte di queste risulteranno infattibili ai fini pratici. Questo fatto rinforza l'idea che gli strumenti di ricerca delle sequenze che si applicano sui grafi non considerino alcuna informazione riguardo l'elemento che costituisce il nodo, ma si basino esclusivamente sulle caratteristiche di connettività tra le parti presenti all'interno dell'assieme.

## 2.3 Metodologia automatica proposta

Come menzionato precedentemente, a livello industriale capita spesso che gli addetti alla produzione e lo staff di progettazione collaborino al fine di definire quale sia la migliore sequenza di assemblaggio di un determinato prodotto; per far ciò, essi si basano sulla propria esperienza in tale ambito, oltre a fare uso di software di modellazione CAD dedicati e algoritmi automatizzati che impongono dei vincoli sulle geometrie delle parti. Una volta scelta la sequenza ottimale, gli stessi dovranno anche ottimizzarla attraverso strumenti di pianificazione virtuale. Questa fase viene svolta solo quando tutte le informazioni di dettaglio del progetto del prodotto sono già disponibili; una possibile modifica potrebbe comportare un grande sforzo di rielaborazione del progetto in termini di tempo e di costo. Così uno strumento automatico per la generazione di sequenze di assemblaggio fattibili, che necessiti solamente delle informazioni che siano disponibili nei primissimi stadi della progettazione, come ad esempio nella fase concettuale, verrà sviluppato in questo contesto.

La metodologia di pianificazione delle sequenze di assemblaggio proposta in questo lavoro richiederà esclusivamente come dati di partenza le sole informazioni di contatto reciproco tra le parti, che sono facilmente ricavabili dal modello 3DCAD dell'assieme in considerazione (R. Viganò and G. O. Gomez, 2011). Questo metodo offrirà il notevole vantaggio di fornire un certo numero ridotto di sequenze di assemblaggio fattibili al progettista ed al generico utente, come punto di partenza nella valutazione, nella generazione e/o nell'ottimizzazione dei processi di assemblaggio.

La fattibilità delle soluzioni trovate dipenderà da ulteriori parametri quantitativi quali la forma, le dimensioni, le geometrie o i materiali impiegati dal momento che in questo metodo verranno considerati solamente parametri topologici. Così facendo, si garantisce che tra tutte le sequenze ricavate in questo modo, ne esista almeno una fattibile da un punto di vista funzionale. Il metodo verrà sviluppato con lo scopo di identificare dei possibili sottoassiemi all'interno dell'intero sistema, i quali verranno poi arrangiati secondo una precisa gerarchia stabilita fino a ricomporre l'assieme completo di tutte le sue parti. Per ogni sottoassieme, un elemento base o piattaforma verrà identificato con lo scopo di generare le relative sottosequenze dello stesso. Dal momento che dei sottoassiemi verranno individuati e successivamente montati, essi costituiranno un nuovo unico elemento all'interno dell'assieme. Il metodo verrà ripetuto fino a che non rimarrà un unico elemento costituente l'intero assieme. Per raggiungere lo scopo prefissato, verranno dunque creati degli indici analitici in grado di classificare tutti i nodi e tutti i cicli presenti nel grafo ad ogni passaggio di riduzione. Gli indici di classificazione dei nodi così creati permetteranno di

identificare in maniera sistematica quei componenti che meglio si presteranno ad essere la base del sottoinsieme individuato. Allo stesso modo, gli indici di classificazione dei cicli permetteranno di identificare quei sottoinsiemi che si presteranno al meglio a venire montati separatamente o per primi. Il calcolo di questi indici si baserà esclusivamente sulle informazioni dedotte dal grafo e dalle tecniche applicative della teoria dei grafi (Diestel, 2005). Proprio a tal proposito, per comprendere al meglio la tecnica applicata, di seguito viene fornita una breve descrizione delle principali definizioni usate nei calcoli.

### 2.3.1 Definizioni

Come già precedentemente illustrato, un'insieme meccanico può essere rappresentato da un grafo  $G$ , che generalmente viene espresso nella forma  $G = (V, E)$  dove  $E \subseteq [V]^2$ . Di seguito verranno brevemente descritti i principali parametri che sono coinvolti nella metodologia proposta, rimandando ogni ulteriore approfondimento alla teoria dei grafi.

#### Coefficiente di Clustering

Il coefficiente di clustering di un vertice in un grafo  $G = (V, E)$  viene impiegato per fornire un peso su quanto un vertice ed i suoi corrispondenti vicini siano prossimi a creare una cricca oppure un grafo completo (Watts, et al., 1998) (Schank, et al., 2005). Una cricca di un grafo  $G$  non è altro che un sottografo di  $G$  isomorfo ad un grafo completo.

Per definizione, il coefficiente di clustering  $CCF(v_i)$  di un vertice  $v_i$  è dato dal rapporto tra il numero di collegamenti esistenti tra i nodi vicini al vertice stesso  $N(v_i)$  ed il numero di collegamenti che potrebbero teoricamente esistere tra i medesimi. Un grafo non orientato ha la proprietà che il collegamento  $e_{ij}$  ed il suo inverso  $e_{ji}$  sono considerati equivalenti. Ad ogni modo, se un vertice  $v_i$  ha  $k_i$  nodi vicini, il numero di lati che potrebbero esistere tra di loro è pari a  $\frac{k_i(k_i-1)}{2}$ .

Così, il coefficiente di clustering del generico vertice  $v_i$  di un grafo non orientato può essere definito come:

$$CCF(v_i) = \frac{2 \cdot |\{e_{jk}\}|}{k_i(k_i - 1)}; v_j, v_k \in N_i; e_{jk} \in E$$

Questo coefficiente assumerà valore unitario se ogni vicino connesso a  $v_i$ , sarà anche connesso a tutti gli altri vicini dello stesso vertice, mentre assumerà valore nullo se non persisteranno collegamenti tra i suoi vicini.

In questo modo è possibile ottenere un riferimento diretto sulla connettività tra le coppie di vertici adiacenti ad un determinato elemento. Questa connettività va intesa come un valore di modularità degli stessi. Dal momento che questo indice, considerato così com'è, non definisce un vero e proprio elemento discriminante tra i vari nodi costituenti il grafo, si definisce il coefficiente di clustering pesato  $WCC(v_i)$  dato dal prodotto del grado del vertice  $v_i$  considerato,  $d(v_i)$ , per il corrispondente coefficiente di clustering  $CCF(v_i)$ .

$$WCC(v_i) = d(v_i) \cdot CCF(v_i)$$

### Cicli Indotti

Nella teoria dei grafi, un ciclo indotto all'interno di un grafo  $G$ , è un ciclo che non presenta corde o collegamenti incrociati. Una corda è definita come un lato che collega due vertici di un ciclo, ma che a sua volta non è parte costituente del ciclo. Per comprendere meglio il concetto, nella Figura 2.10 sono presentati 3 cicli, di cui i 2 indicati con  $C^3$  sono indotti, mentre quello indicato con  $C^4$  non lo è, in quanto presenta una corda tra il vertice 3 e il vertice 5. L'importanza di questi cicli in questo approccio automatico è dovuta al fatto che essi rappresentano il sottoinsieme con il minor numero di elementi che si può prendere in considerazione. Sorge la difficoltà che, nella pratica comune, non esiste un algoritmo utile all'identificazione automatica di questi cicli indotti all'interno di un grafo generico.

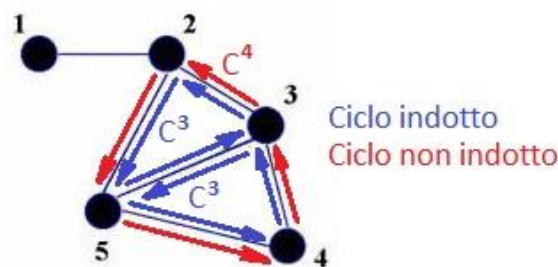


Figura 2.10 Cicli indotti  $C^3$  e ciclo non indotto  $C^4$  che presenta la corda 3-5.

La teoria che più si avvicina a questo determinato problema riguarda la ricerca dei cicli fondamentali in un grafo precedentemente menzionata. In letteratura, i cicli fondamentali di un grafo vengono definiti come i cicli più semplici che è possibile ottenere dal relativo schema ad albero, ma lo stesso

autore osserva che, tra quelli individuati in questa modalità, potrebbe esserci la presenza di cicli contenenti collegamenti incrociati (Gotlieb, et al., 1967)(paton, 1969)(Deo, et al., 1982)(Gibbs,1969).

Supponendo di considerare un generico grafo connesso, esisterà sicuramente il relativo schema ad albero composto da tutti gli “ $n$ ” vertici e gli “ $n-1$ ” lati del grafo. Aggiungendo anche un solo lato nello schema ad albero, questo creerà un ciclo, chiamato ciclo fondamentale. Esiste quindi una corrispondenza univoca tra i cicli fondamentali ed il lati non presenti nello schema ad albero, ossia, un grafo con “ $m$ ” lati avrà “ $m-n+1$ ” cicli fondamentali (Eppstein, 1996). Un esempio su quanto appena detto, è illustrato in Figura 2.11, dove la serie di cicli fondamentali, come da definizione della letteratura, è:  $FC = \{c_1, \dots, c_5\}$ , dove  $c_1 = (A-B-F-A)$ ,  $c_2 = (A-F-E-A)$ ,  $c_3 = (C-F-D-C)$ ,  $c_4 = (A-B-C-F-A)$  e  $c_5 = (A-F-D-E-A)$ . Come è stato osservato precedentemente, i cicli  $c_4$  e  $c_5$  contengono collegamenti come B-F e E-F che sono collegamenti incrociati.

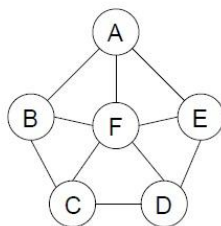


Figura 2.11 Grafo generico con 6 nodi e 10 lati.

Il metodo applicato inoltre non identifica i cicli  $c_6 = (B-C-F-B)$ ,  $c_7 = (D-E-F-D)$  e  $c_8 = (A-B-C-D-E-A)$  che esistono e non presentano collegamenti incrociati al loro interno. Quindi, questo metodo applicativo sistematico non risulta utile ai fini di questo lavoro. Esiste però, un'ulteriore ricerca svolta da Kang (Kang, 2006) che non tratta problemi di natura meccanica, ma riguarda la capacità di sopravvivenza delle reti di comunicazione, dove lo stesso ricercatore attribuisce all'individuazione dei cicli indotti in un grafo, una possibile soluzione per quanto riguarda la protezione di queste reti di comunicazione. L'algoritmo da lui proposto, nonostante sembra essere di scarsa utilità ai fini di questa ricerca, se opportunamente modificato può condurre ad un ottimo punto di partenza al fine di raggiungere i risultati voluti.

L'algoritmo in argomento si basa sugli stessi strumenti utilizzati per la ricerca dei cicli fondamentali, ossia sui metodi Breadth First Search (BFS) e Depth First Search (DFS). Entrambi i metodi generano tutti i cicli che si possono generare basandosi sullo schema ad albero del corrispondente grafo.

Il metodo BFS controlla ogni possibile tragitto ,partendo da ogni nodo, per determinare se esso forma un ciclo chiuso; il termine Breadth indica che il metodo dà la priorità, durante l'esplorazione, ai cicli più corti. Analogamente, ma all'opposto, il metodo DFS esplora ogni possibile tragitto partendo da tutti i nodi; nel caso il tragitto in questione non contenga un nodo di destinazione, un altro tragitto verrà esplorato rintracciando quello percorso precedentemente fino a che un ciclo chiuso non venga individuato. Secondo Greenlaw (Greenlaw, 1993), combinando opportunamente i due algoritmi è possibile dare una soluzione al problema dell'individuazione dei cicli indotti partendo dall'individuazione dei cicli fondamentali. Così, questo nuovo algoritmo ibrido esplorerà prima tutti possibili cicli fondamentali che passano per un determinato nodo, dopodiché il nodo in questione verrà rimosso dal grafo e il processo sarà ripetuto fino a che tutti i nodi verranno esaminati. Infine, tramite la numerazione dei cicli individuati, sarà possibile eliminare quelli che si presentano più volte. In aggiunta, ogni volta che l'algoritmo identificherà un ciclo tramite la chiusura di un percorso, lo stesso controllerà che i nodi appartenenti a quel ciclo non creino delle corde, ed in tal caso eliminerà il ciclo in questione.

### Classifica di centralità di un nodo

La centralità di un nodo deriva da una serie di parametri, o indici, utili a identificare caratteristiche salienti del nodo all'interno di un assieme, fornendo così un'informazione importante sull'ordine di assemblabilità (R. Viganò and G. O. Gomez, 2011). I fattori che vengono identificati come i più discriminanti sono: il grado del nodo, la sua relativa partecipazione a costituire cicli nel grafo, la media dei gradi dei nodi vicini al nodo considerato ed infine il valore di clustering. Sommando questi parametri in maniera pesata, è possibile ottenere un indice che rappresenterà la priorità di montaggio che un determinato nodo (e quindi componente o sottoassieme) ha su tutti quelli presenti. Ad ogni passaggio di iterazione, verrà quindi valutato l'indice di centralità dei nodi presenti, in modo da poterli classificare e successivamente poter scegliere quello più adatto da usare come "nodo piattaforma". E' importante notare che gli indici classificheranno i nodi in funzione di tutti quelli presenti nell'assieme in analisi. Di seguito vengono brevemente introdotti gli indicatori che verranno usati per definire l'indice di centralità del nodo.

### **Indice del grado del nodo, $I_{ND}$**

Il grado di un nodo corrisponde al numero di nodi vicini a quello preso in considerazione, questo valore non è altro che il numero di collegamenti pertinenti ad un determinato nodo; per nodo vicino si intende quel nodo che ha



in comune almeno un collegamento con quello preso in considerazione. In un sistema meccanico, il grado di un componente indicherà la sua importanza all'interno dell'assieme, fornendo a tal proposito il numero di parti che mostrano un contatto con esso.

L'indice del grado del nodo  $I_{ND}$ , corrisponde al rapporto tra il grado del nodo e la somma dei gradi di tutti i nodi presenti nel grafo e viene definito come segue:

$$I_{ND_i} = \frac{d(v_i)}{\sum_{v_i \in V} d(v_i)} ; \quad 0 \leq I_{ND_i} \leq 1$$

### **Indice della media dei gradi dei nodi adiacenti, $I_{DAN}$**

Questa unità rappresenterà l'importanza di un componente in relazione ai suoi elementi vicini, che siano essi dei nodi o dei sottoassiemi generati all'interno del sistema meccanico. Dall'analisi delle prove effettuate, è stato osservato che un basso valore di questo indice determina una scelta vantaggiosa di quel determinato nodo come "piattaforma"; proprio per questo motivo, l'indice della media dei nodi adiacenti,  $I_{DAN}$ , verrà pesato negativamente. La media pesata dei nodi adiacenti ad un determinato elemento ( $v_i$ ), detta  $\overline{DAN}(v_i)$ , sarà data dalla somma dei gradi dei vicini diviso il numero degli stessi.

Ossia:

$$\overline{DAN}(v_i) = \frac{\sum_{v_i \in N(v_i)} d(v_i)}{d(v_i)}$$

Mentre l'indice  $I_{DAN}$ , sarà definito dalla relazione tra il valore  $\overline{DAN}(v_i)$  del nodo ( $v_i$ ) e la somma degli stessi valori calcolati per tutti gli altri elementi presenti nel grafo  $V$ , come:

$$I_{DAN_i} = \frac{\overline{DAN}(v_i)}{\sum_{v_i \in V} \overline{DAN}(v_i)} ; \quad 0 \leq I_{DAN_i} \leq 1$$

### **Indice di pertinenza ai cicli, $I_{PC}$**

Tutti i nodi ( $v_i$ ) di un grafo non orientato e connesso con grado maggiore o uguale a due appartengono come minimo ad un ciclo indotto  $c(v_i)$  del set totale di cicli  $C$ . Questo fatto è vero se nessun vertice con grado uguale ad uno esiste nel sistema. In un sistema meccanico, questo indice rappresenta il valore di intercambiabilità di un nodo riguardo i differenti cicli presenti, ossia i potenziali sottoassiemi del sistema. Questo numero è sensibilmente influenzato dal grado del nodo, dal momento che un nodo con un grado alto, normalmente indica una grande partecipazione dello stesso all'interno del

ciclo considerato. Proprio per questo motivo, questo indice sarà un ottimo parametro discriminante per la classificazione dei nodi con lo stesso grado presenti nel medesimo ciclo.

Il rapporto di pertinenza  $PC(v_i)$  di un nodo ( $v_i$ ) viene definito dal rapporto tra il numero di cicli pertinenti a quel nodo  $c(v_i)$  ed il numero di cicli dell'intera serie dei cicli  $C$ ; equivale a dire:

$$PC(v_i) = \frac{|c(v_i)|}{|C|}$$

Per definizione,  $c(v_i)$  indica i cicli indotti del set  $C$  che contengono il nodo ( $v_i$ ) come uno dei suoi elementi. L'indice di pertinenza ai cicli  $I_{PC}$ , corrisponderà, come in tutti gli altri casi, al rapporto di pertinenza di un determinato vertice diviso la somma dei rapporti di pertinenza di tutti i vertici presenti nel grafo, ossia:

$$I_{PC_i} = \frac{PC(v_i)}{\sum_{v_i \in V} PC(v_i)} ; \quad 0 \leq I_{PC_i} \leq 1$$

### **Indice di clustering, $I_{CCF}$**

Questo indice rappresenterà il grado di connessione tra i vicini del nodo considerato ( $v_i$ ), come mostrato nella definizione del coefficiente di clustering di un nodo  $CCF(v_i)$ . Per un sistema meccanico, questa misura indicherà se un determinato elemento sarà connesso ad un possibile modulo presente. Come per l'indice della media dei gradi dei nodi adiacenti,  $I_{DAN}$ , dall'analisi di alcuni casi test, si è verificato che un basso valore dell'indice di clustering fornirà una scelta vantaggiosa nell'imporre lo stesso nodo come piattaforma per il sottoassieme da ridurre; quindi per il calcolo dell'indice di centralità il presente parametro verrà pesato negativamente.

L'indice di clustering di un nodo  $I_{CCF}$  sarà definito dal rapporto tra il coefficiente di clustering pesato per quel dato nodo  $WCCF(v_i)$ , diviso la somma degli stessi coefficienti di tutti i nodi componenti il sistema  $V$ ; ossia:

$$I_{CCF_i} = \frac{WCCF(v_i)}{\sum_{v_i \in V} WCCF(v_i)} ; \quad 0 \leq I_{CCF_i} \leq 1$$

### **Indice totale di centralità, $I_{CT}$**

Infine, tutti gli indici menzionati, per un determinato nodo, verranno pesati e fusi insieme in un unico indice che rappresenterà la totalità di giudizio dello stesso. Il valore dei pesi per i differenti indici verrà deciso volta per volta,

euristicamente, basandosi sul tipo di assieme (e quindi grafo) considerato e sui risultati dei casi test precedentemente analizzati.

Il valore di questo indice viene calcolato sommando tutti gli indicatori precedentemente calcolati, moltiplicati per il relativo peso; quindi si avrà che per un generico vertice ( $v_i$ ), l'indice totale di centralità sarà dato da:

$$I_{CT_i} = \frac{w_{ND} \cdot I_{ND_i} + w_{PC} \cdot I_{PC_i} - w_{DAN} \cdot I_{DAN_i} - w_{CCF} \cdot I_{CCF_i}}{(w_{ND} + w_{PC} - w_{DAN} - w_{CCF})}; \quad 0 \leq I_{CT_i} \leq 1$$

Dove:

- $w_{ND}$  indica il peso dell'indice del grado del nodo
- $w_{PC}$  indica il peso dell'indice di pertinenza ai cicli
- $w_{DAN}$  indica il peso dell'indice della media dei gradi dei nodi adiacenti
- $w_{CCF}$  indica il peso dell'indice di clustering

Come è stato già affermato, il valore del peso degli indici di media dei gradi dei nodi adiacenti e di clustering corrisponde ad un valore negativo, dal momento che un valore alto svantaggerà la scelta del nodo come piattaforma del sottoassieme. Questo indice fornirà così un criterio numerico di scelta del miglior nodo piattaforma ad ogni livello di riduzione dell'assieme, utile per l'implementazione automatica del metodo proposto.

### Classifica di importanza di un ciclo

Come per i nodi, ogni volta che un sottoassieme dovrà essere selezionato tra tutti possibili cicli indotti presenti nel grafo, verranno definiti alcuni parametri che forniranno un criterio numerico di scelta che potrà essere successivamente automatizzato. Quindi, nella stessa modalità impiegata per i nodi, l'importanza di un ciclo verrà dedotta tramite l'influenza di alcuni indici che valutino caratteristiche quali il grado del ciclo, la sua connettività e la sua centralità all'interno dell'assieme completo. Ancora una volta, tutti gli indicatori calcolati classificheranno i cicli in relazione a tutti quelli presenti nel grafo ad un determinato passo di riduzione. Con lo scopo di poter classificare i cicli presenti, al fine di individuare quello che costituirà il miglior sottoassieme da ridurre al successivo passo di riduzione, verranno di seguito definiti tre diversi indici.

### **Indice del grado del ciclo, $I_{CD}$**

Il grado di un ciclo  $c_i$ , come per i nodi d'altronde, corrisponderà al numero di nodi vicini connessi con tutti i componenti del ciclo stesso; questo valore non sarà necessariamente pari al numero di lati che comporranno il ciclo.

Bisogna prestare attenzione al fatto che, se due nodi appartenenti al ciclo siano collegati al medesimo elemento esterno ad esso, allora l'elemento esterno verrà considerato una volta sola. Riepilogando, il grado di un ciclo si riferisce al numero dei nodi vicini a quel ciclo senza contarne le molteplicità di collegamento dello stesso.

Il suo valore verrà definito attraverso il rapporto tra il grado del ciclo  $c_i$  in questione e la somma dei gradi di tutti i cicli presenti nel grafo, ad un dato livello di riduzione dell'assieme, ossia:

$$I_{CD_i} = \frac{d(c_i^k)}{\sum_{c_i^k \in C} d(c_i^k)} ; \quad 0 \leq I_{CD_i} \leq 1$$

Come precedentemente affermato, il valore dell'indice appena menzionato non dipenderà dal numero di lati del ciclo in esame.

### **Indice di connettività del ciclo, $I_{CC}$**

La connettività di un ciclo  $CC(c_i^k)$ , per un ciclo indotto con  $k$  componenti, viene definita dal rapporto tra il numero di vicini ed il numero di nodi che non appartengono a quel ciclo, come:

$$CC(c_i^k) = \frac{d(c_i^k)}{|V| - k}$$

Questo valore fornirà una stima di importanza del ciclo, ossia del sottoassieme considerato, all'interno dell'intero sistema.

L'indice di connettività  $I_{CC}$ , come per tutti gli altri indici, verrà calcolato relazionando il parametro sopra citato alla somma degli stessi per tutti gli altri cicli presenti, ovvero:

$$I_{CC_i} = \frac{CC(c_i^k)}{\sum_{c_i^k \in C} CC(c_i^k)} ; \quad 0 \leq I_{CC_i} \leq 1$$

### **Indice di centralità dei nodi del ciclo, $I_{NC}$**

A questo punto, lo sviluppo del ciclo  $i$ -esimo rispetto a tutti i suoi elementi esterni verrà valutato; la via migliore per analizzare lo sviluppo di un ciclo all'interno di un assieme completo, è quella di considerare la somma degli indici di centralità dei nodi dello stesso.

Al fine di poter effettuare una comparazione tra tutti i cicli presenti ad un dato livello, indipendentemente dalla loro lunghezza  $k$ , viene definita la centralità dei nodi del ciclo  $\overline{NC}(c_i^k)$ , come:

$$\overline{NC}(c_i^k) = \frac{\sum_{v_j \in c_i^k} I_{CT}(v_j)}{k}$$

Come per gli altri casi, il valore calcolato verrà relazionato allo stesso valore che è possibile calcolare su tutti i cicli restanti  $C$ , presenti nel grafo, come:

$$I_{NC_i} = \frac{\overline{NC}(c_i^k)}{\sum_{c_i^k \in C} \overline{NC}(c_i^k)}; \quad 0 \leq I_{NC_i} \leq 1$$

### Indice di importanza del ciclo, $I_{CI}$

I tre indici appena calcolati verranno successivamente pesati e fusi insieme in un unico indice che permetterà di realizzare un criterio numerico di diversificazione al fine di individuare il “ciclo migliore”. Il ciclo con l’indice di importanza maggiore, identificherà il migliore sottoassieme che si presterà ad essere montato in determinato passo di riduzione. Come nel caso della scelta del miglior nodo, ogni indice verrà pesato attraverso dei coefficienti che, sulla base degli studi empirici su prove sperimentali, si suppone abbiano lo stesso livello di collaborazione, ossia assumano lo stesso valore tra loro. Per semplicità di analisi, nelle prove sperimentali illustrate di seguito, questi indici verranno associati ad un valore unitario.

In questa maniera, sarà possibile definire l’indice di importanza di un ciclo  $I_{CI_i}$ , come segue:

$$I_{CI_i} = \frac{w_{CD} \cdot I_{CD_i} + w_{CC} \cdot I_{CC_i} + w_{NC} \cdot I_{NC_i}}{(w_{CD} + w_{CC} + w_{NC})}; \quad 0 \leq I_{CI_i} \leq 1$$

Dove:

- $w_{CD}$  indica il peso dell’indice del grado del ciclo
- $w_{CC}$  indica il peso dell’indice di connettività del ciclo
- $w_{NC}$  indica il peso dell’indice della centralità dei nodi del ciclo

Così facendo, si otterrà ad ogni passo iterativo del processo di generazione, un metodo numerico di classificazione (ranking) dei cicli utile per la scelta del miglior sottoassieme da ridurre ad un dato livello. Una volta resi noti gli indici di importanza dei cicli  $I_{CI_i}$  e di centralità totale dei nodi  $I_{CT_i}$ , non resta che definire una procedura univoca di riduzione dell’assieme, al fine di identificare un numero discreto di possibili sequenze di assemblaggio, fattibili e non fattibili, coerenti con la teoria proposta.

### 2.3.2 Procedura

Viene a questo punto illustrata la procedura sistematica impiegata con lo scopo di individuare un certo numero di differenti sequenze di assemblaggio di un determinato prodotto, basandosi su dei criteri di scelta numerica, come ad esempio tramite la classificazione attraverso gli indici appena esposti e sulla presenza di nodi con grado 1. Il diagramma di flusso di Figura 2.12 illustra in sintesi l'intero procedimento di generazione delle sequenze sviluppato da Osorio Gomez e Viganò (R. Viganò and G. O. Gomez, 2011) e implementato in questo contesto al calcolatore. Come mostra il diagramma di flusso, il primo passaggio consisterà nell'acquisizione delle informazioni necessarie dell'assieme e nella generazione del grafo e della struttura dati necessaria all'algoritmo. Dopodiché, verrà valutata la presenza di nodi con grado unitario e verrà indirizzata la semplificazione dell'assieme o alla riduzione iterativa dei nodi semplicemente connessi, o alla riduzione del migliore ciclo. Nel secondo caso, una sequenza di procedure, dedicate all'individuazione del sottoassieme e alla modalità di semplificazione dello stesso, viene applicata. E' importante notare che ad ogni passaggio di riduzione di un sottoassieme, tutti gli indici sui cicli e sui nodi verranno ricalcolati, mentre nel caso della riduzione dei nodi di grado unitario la classificazione tramite gli indici non sarà necessaria. La procedura si ripeterà fino a quando non rimarrà un unico elemento, o nodo, sul grafo che indicherà implicitamente il raggiungimento dell'ultimo livello di riduzione dell'assieme. Per meglio comprendere la procedura applicata, la spiegazione verrà suddivisa in diversi passi che costituiranno i blocchi delle operazioni fondamentali sul diagramma di flusso.

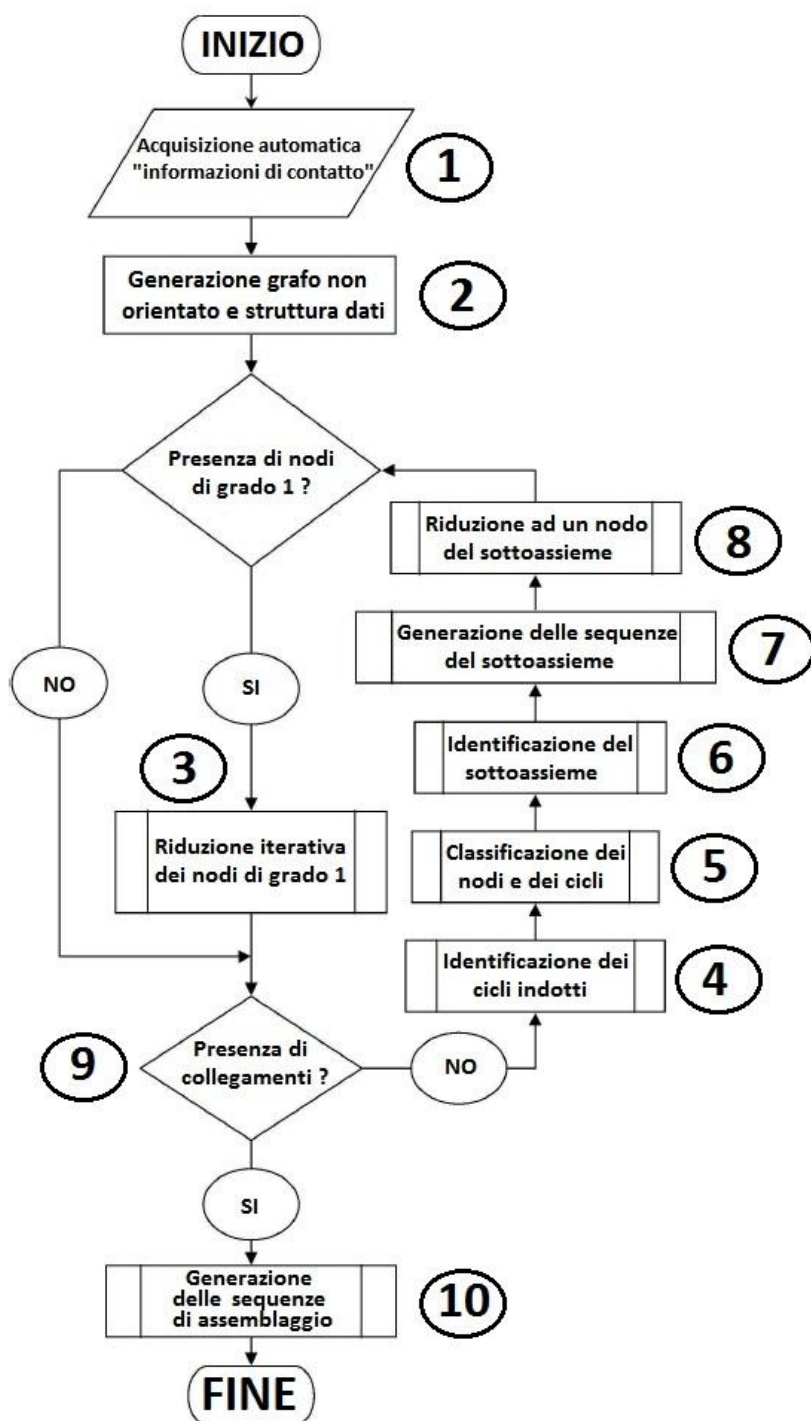


Figura 2.12 Diagramma di flusso per la generazione automatica delle sequenze.

Le attività fondamentali, ossia quelle che verranno descritte nel dettaglio, sono affiancate da un numero di riferimento nella Figura 2.12.

### 1. Identificazione dei contatti tra i componenti

Le informazioni utili per la pianificazione dei possibili assemblaggi si otterranno dal modello 3D-CAD dell'assieme completo. Partendo dalle informazioni di posizione delle parti, rispetto ad un riferimento assoluto, nell'assieme, si cercheranno tutte le interazioni di contatto tra di esse. Qualsiasi altra informazione, come ad esempio le relazioni di accostamento tra i componenti (come accoppiamenti, coassialità e angoli di orientamento) e le loro regole di assemblaggio, non verranno considerate ai fini di questo lavoro, dal momento che non fornirebbe nessuna collaborazione con il metodo proposto. Inoltre, si suppone che le parti che compongono l'assieme nel modello CAD, siano montate nella loro posizione di equilibrio stabile e non presentino giochi o interferenze indesiderate. Partendo dal modello 3D-CAD dell'assieme meccanico, il software CAD verrà interrogato sulle relazioni di contatto esistenti tra le parti, ossia dove si presenti un contatto fisico o un vincolo di fissaggio, e verrà creata una relativa tabella dei risultati dove ogni riga rappresenterà un collegamento tra due parti (come mostrato in Figura 2.13).

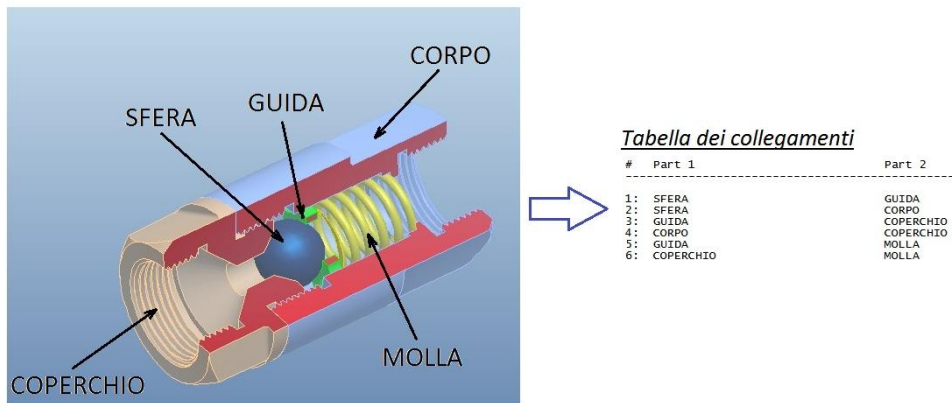


Figura 2.13 Informazioni di contatto ottenute dal modello 3D-CAD dell'assieme.

Con lo scopo di rendere il più semplice possibile l'elaborazione e l'analisi dei risultati ottenuti, si è deciso di seguire le ipotesi fatte da De Fazio e Whitney per cui si sceglie di omettere quei componenti di



fissaggio quali viti, dadi, rondelle, anelli di tenuta e perni che si presentino ripetutamente all'interno del modello (R. Viganò and G. O. Gomez, 2011). Questo non significa che il metodo sia applicabile solamente attraverso questa semplificazione, al contrario, nel caso in cui la loro presenza sia in numero ridotto o sia necessaria per una descrizione esauriente del modello, quest'ultimi verranno presi in considerazione.

Allo stesso modo, alcuni componenti montati sul modello potranno venir sintetizzati in un'unica parte, riducendo in maniera significativa la mole di calcoli. Una corretta modellazione del prodotto e una coerente disposizione delle parti che lo compongono, influenzerà in maniera positiva i risultati ottenuti, evitando inoltre la generazione di sequenze di montaggio errate. Quanto appena detto può essere facilmente appreso dall'esempio mostrato in Figura 2.14. Nell'esempio viene mostrato come una errata modellazione dell'assieme comporti a sua volta un'errata generazione della lista dei collegamenti, ossia del grafo equivalente.

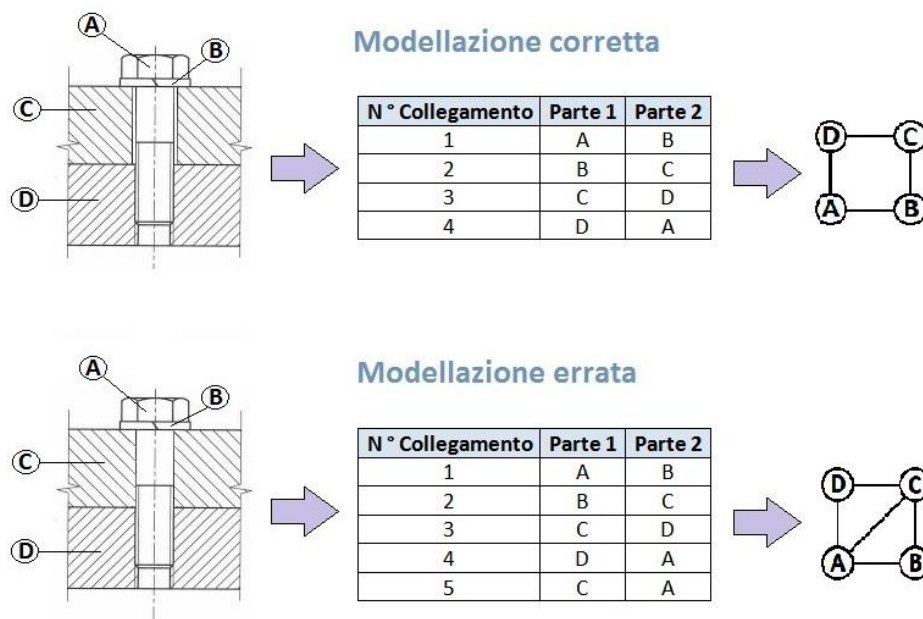


Figura 2.14 Relazione tra la modellazione dell'assieme e la generazione dei dati.

## 2. Generazione del grafo non orientato e della struttura dati

Dalla lista dei componenti e dalla lista delle informazioni di contatto, l'algoritmo genererà una struttura dati, chiamata *ASPIPData*,

allocando ad essa uno spazio di memoria. Tramite questa sarà possibile generare il grafo rappresentativo della topologia dell'assieme. Il grafo non orientato così rappresentato, non verrà impiegato nella metodologia iterativa di generazione delle sequenze, ma fornirà un ottimo strumento di rappresentazione in ausilio all'utente che dovrà verificare la correttezza delle informazioni riportate. Infatti, la valutazione da parte dell'utente delle informazioni acquisite dal software, sarà molto più semplice grazie alla visualizzazione di un grafo piuttosto che attraverso una lista di collegamenti. Quindi, ogni grafo che verrà rappresentato d'ora in avanti, non avrà più un'utilità diretta all'interno del procedimento, ma fornirà un ottimo strumento di comparazione dei risultati ottenuti. La Figura 2.15 mostra un esempio di assieme ed il relativo grafo dei contatti creato.

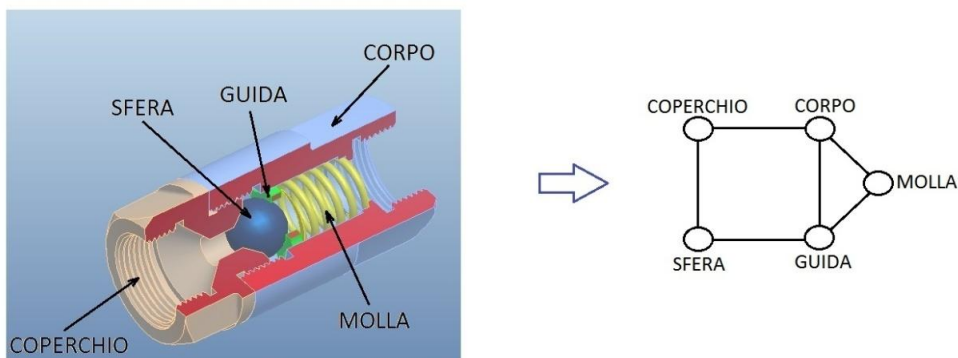


Figura 2.15 Rappresentazione assieme meccanico attraverso grafo non orientato.

### 3. *Riduzione iterativa dei nodi di grado unitario*

Una riduzione iterativa dei collegamenti dell'assieme verrà applicata con lo scopo di unire, riducendo ad un unico elemento, i nodi di grado unitario con il nodo adiacente. Il montaggio di parti di grado unitario con la loro parte vicina corrisponderà alla prima attività da svolgere in fase di assemblaggio. Infatti, se un nodo ha grado unitario, significa che la rispettiva parte sarà a contatto unicamente con un'altra parte dell'assieme; da questo si desume che la possibilità di montaggio è univoca, ossia la parte di grado uno dovrà essere montata sulla parte adiacente o viceversa. Questa riduzione verrà fatta secondo il valore del grado del nodo adiacente, questo perché se si presentasse un sottografo lineare all'interno del grafo completo che

presenti una sequenza di nodi di grado 2 e degli altri nodi adiacenti ad essa di grado uno, ad ogni passo di riduzione solo il nodo adiacente con il grado più basso permetterà un corretto montaggio delle due parti.

Si consideri a tal proposito l'esempio del modello di una penna a sfera generica (De Fazio, et al.,1987) e del suo relativo grafo dei contatti come mostrato in Figura 2.16 e Figura 2.17.

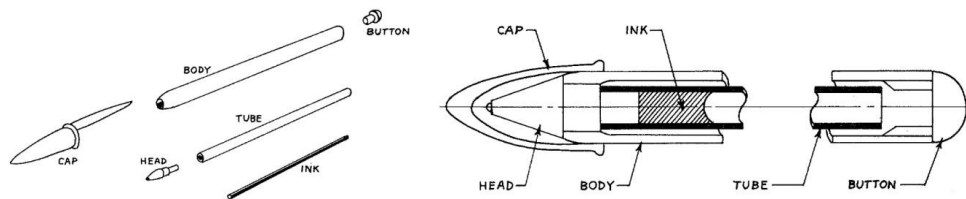


Figura 2.16 Modello di una generica penna sfera (De Fazio e Whitney).

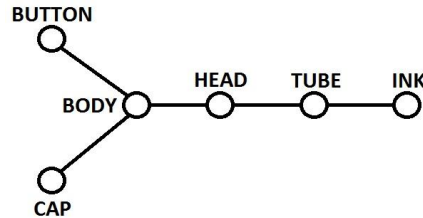


Figura 2.17 Grafo non orientato del modello penna a sfera (De Fazio e Whitney).

In questo caso, nonostante al primo passo di riduzione tre nodi di grado uno vengano identificati, solo il nodo *INK* verrà ridotto con il nodo *TUBE* dal momento che quest'ultimo sarà il nodo con il minore grado presente. La stessa situazione si verificherà nel secondo passo di riduzione dove il nuovo sottoassieme *INK-TUBE* verrà unito al nodo *HEAD*.

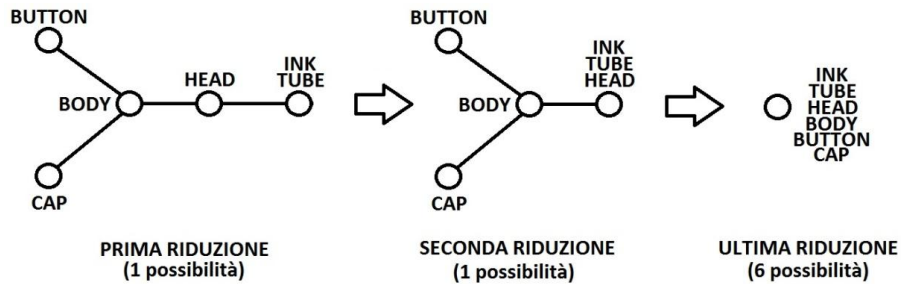


Figura 2.18 Riduzione dell'esempio penna a sfera (De Fazio e Whitney).

Nell'ultimo passaggio di riduzione saranno presenti tre nodi di grado unitario con in comune lo stesso nodo adiacente *BODY* di grado pari a 3; in questo caso, non essendoci nodi adiacenti discriminanti, tutte e sei le possibili sequenze verranno generate supponendo di poter comporre alternativamente l'assieme totale iniziando volta per volta da un nodo di grado unitario diverso e percorrendo tutte le possibili strade di riduzione.

Per l'esempio penna a sfera, la totalità delle possibili sequenze di assemblaggio ottenute viene riassunta nella seguente Tabella 2.1. Per l'esempio in questione, il numero totale delle possibili sequenze di assemblaggio generate corrisponderà alla moltiplicazione tra il numero delle sequenze ottenute ad ogni passo di riduzione, ossia verranno generate  $1 \times 1 \times 6 = 6$  possibili sequenze. Non è detto che le eventuali sequenze generate siano realizzabili, nonostante queste siano coerenti con la teoria applicata. A tal scopo, come verrà illustrato meglio successivamente, sarà possibile ridurne il numero rispondendo ad una serie di interrogativi riguardanti la fattibilità di ogni attività di assemblaggio contemplata dalla generica sequenza. A questo punto, si otterrà un assieme che non conterrà solamente nodi con grado maggiore o uguale a due. Il grafo non orientato che verrà generato al livello successivo, sarà un grafo connesso e composto da una serie di cicli indotti.

Tabella 2.1 Rappresentazione delle sequenze per l'esempio penna a sfera.

LIVELLO	COMPONENTI SOTTOASSIEME	NOME SOTTOASSIEME	POSSIBILE SEQUENZA DI ASSEMBLAGGIO
1	INK-TUBE	SA1	TUBE<INK
2	HEAD-SA1	SA2	HEAD<SA1
			BODY<SA2<BUTTON<CAP
			BODY<SA2<CAP<BUTTON
3	BUTTON-BODY-CAP-SA2	SA3	BODY<BUTTON <SA2<CAP
			BODY<BUTTON <CAP<SA2
			BODY<CAP<SA2<BUTTON
			BODY<CAP<BUTTON<SA2

#### 4. Identificazione dei cicli indotti

Una volta ridotti tutti i nodi di grado unitario, verranno creati i nuovi grafi non orientati dei collegamenti e la nuova struttura dati. Dunque si procederà con la riduzione ad un unico elemento del miglior ciclo indotto individuato. Per fare ciò, sarà necessario determinare tutti i cicli indotti presenti in quel dato livello. L'algoritmo di individuazione dei cicli indotti verrà applicato sulla nuova struttura dati. L'uso di un algoritmo metodico nella ricerca dei cicli indotti semplifica notevolmente il procedimento che altrimenti verrebbe svolto manualmente. Inoltre, la ricerca manuale dei cicli indotti che sono presenti in un grafo può essere complessa. Questa difficoltà è dovuta dal fatto che un grafo può venire espresso in diverse conformazioni, mettendo in luce solo alcuni cicli in ognuna di queste, a scapito di altri. Questa proprietà può essere compresa meglio facendo riferimento alla Figura 2.19, dove lo stesso grafo presenta due differenti disposizioni dei nodi che lo compongono.

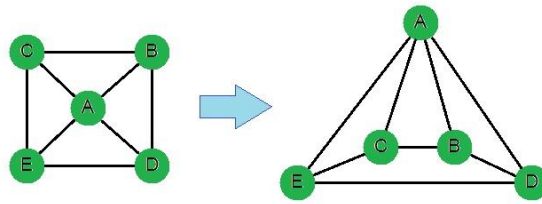


Figura 2.19 Differente disposizione dei nodi dello stesso grafo.

In questo particolare esempio, si nota immediatamente che ridisponendo i nodi del grafo secondo un ordine differente si esplicita la presenza del ciclo indotto  $E-C-B-D$ , che nella prima disposizione non era visibile. Viceversa, il ciclo  $E-A-D$  che nella prima configurazione era facilmente visibile, nella seconda disposizione non lo è più.

#### 5. *Classificazione dei nodi e dei cicli*

Una volta individuati tutti i cicli indotti presenti nel grafo, l'indice di centralità dei nodi,  $I_{CT_i}$ , e l'indice di importanza dei cicli,  $I_{CI_i}$ , verranno calcolati per tutti i nodi e tutti cicli presenti. Dopodiché sarà possibile classificare i nodi ed i cicli al fine di scegliere quale sia il miglior sottoassieme da ridurre e su che nodo piattaforma ridurlo.

#### 6. *Identificazione del sottoassieme da ridurre*

Il ciclo indotto con il maggiore indice di importanza  $I_{CI_i}$  verrà selezionato quale miglior sottoassieme da ridurre. Inoltre, tra tutti i nodi del ciclo considerato, quello con il maggiore indice di centralità  $I_{CT_i}$  diventerà il suo elemento piattaforma su cui si esegue la riduzione. Se, come può accadere, più cicli indotti presentassero il massimo valore dell'indice di importanza  $I_{CI_i}$ , entrambe le possibilità di riduzione sarebbero valide e verrebbero portate avanti separatamente, generando così due diversi sottolivelli di riduzione. Sarà così necessario discernere le due diverse strade di riduzione fino all'ultimo livello, e memorizzare tutte le possibili soluzioni ottenute in entrambi i casi.

### 7. Generazione delle sequenze di assemblaggio del sottoassieme

Una volta determinato il migliore sottoassieme, tutte le possibili sottosequenze che derivano da esso partiranno con il nodo base e, rispetto alle regole esposte nel paragrafo 2.2.2 riguardanti la riduzione dei collegamenti, saranno pari a:

$$2^{(k-2)} \text{ possibili sequenze generate}$$

Dove  $k$  indicherà la lunghezza del ciclo preso in considerazione.

Se invece più di un nodo appartenente al ciclo in esame presentasse lo stesso indice di centralità, allora altrettante sequenze verrebbero generate a partire da quel nodo, come è possibile intuire dalla forma generica della precedente relazione:

$$N \times 2^{(k-2)} \text{ possibili sequenze generate}$$

Dove  $k$  indicherà ancora una volta la lunghezza del ciclo preso in considerazione, mentre  $N$  indicherà il numero di nodi che hanno lo stesso valore dell'indice di centralità. Un esempio di quanto appena detto può essere descritto considerando un semplice grafo completamente connesso composto da tre elementi, come mostrato in Figura 2.20.

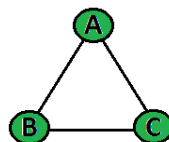


Figura 2.20 Grafo a tre elementi completamente connesso.

In questo particolare caso, un solo ciclo indotto A-B-C è identificabile. Dal momento che tutti e tre i nodi assumono lo stesso valore dell'indice di centralità  $I_{CT_i}$ , si avrà la generazione di:

$$3 \times 2^{(3-2)} = 6 \text{ possibili sequenze generate}$$

La serie di sequenze che si otterranno tramite la riduzione del sottoassieme composto dai tre elementi dell'esempio è illustrata in Tabella 2.2, dove la seconda colonna indica il relativo nodo base sul quale si genererà la sottosequenza.

Tabella 2.2 Sequenze generate dal grafo dell'esempio in Fig. 25.

Numero	Nodo Base	Sequenza generata
1	A	A<B<C
2		A<C<B
3	B	B<A<C
4		B<C<A
5	C	C<A<B
6		C<B<A

**8. Riduzione del sottoassieme nel grafo**

Una volta che il sottoassieme da montare per primo verrà identificato, l'algoritmo genererà e memorizzerà tutte le sue possibili sottosequenze di assemblaggio, assegnando un nome al sottoassieme, e riducendo lo stesso ad un unico nodo secondo le priorità imposte dal nodo base. Durante la semplificazione del ciclo in un unico nuovo nodo, che prenderà il nome dal relativo sottoassieme, verrà evitata la formazione di possibili lati doppi o "loop", come mostrato in Figura 2.21.

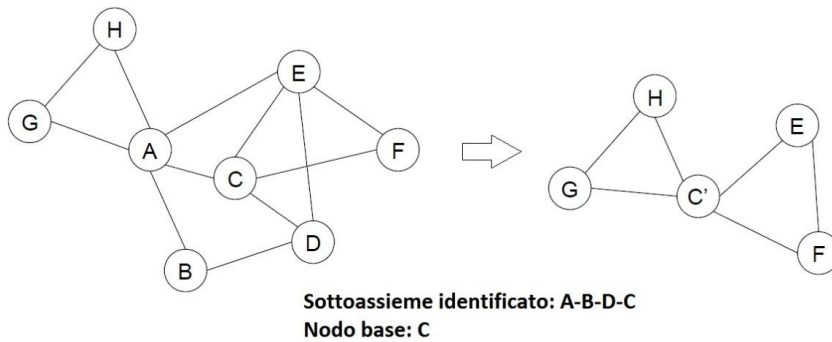


Figura 2.21 Riduzione di un ciclo all'interno di un grafo.



### **9. Iterazione**

La procedura riprenderà dal punto 3, ossia dalla ricerca e dalla semplificazione iterativa dei nodi di grado uno. Il metodo si ripeterà fino a che non si ottenga la completa riduzione di tutto il sistema ad un unico sottoassieme. Inoltre, se rimarrà un solo ciclo, lo stesso verrà ridotto ad un unico elemento fittizio e le relative sottosequenze di assemblaggio verranno generate e dipenderanno da  $k$  secondo la regola esposta al punto 7.

### **10. Generazione delle sequenze di assemblaggio totali**

L'ultima azione compiuta dall'algoritmo consisterà nella creazione di tutte le sequenze di assemblaggio. Queste sequenze, come già affermato precedentemente, potranno venire rappresentate sia in forma esplicita che in forma implicita. La forma implicita è la forma più compatta con cui è possibile rappresentare una singola sequenza e consiste nel rappresentarla con il minor numero di sottoassiemi disponibili, ossia con i sottoassiemi dell'ultimo livello. Viceversa, la forma esplicita consiste nel rappresentare la sequenza attraverso tutte le parti che compongono l'assieme; quest'ultima consisterà in una lista ordinata delle parti da assemblare volta per volta, partendo dall'assemblaggio della prima parte con la seconda fino a montare l'ultima parte presente nella sequenza. Al fine di poter successivamente valutare quale sia la validità delle sequenze generate, si è deciso di generarle in forma esplicita. L'algoritmo, che alla sua ultima riduzione creerà uno o più sottoassiemi, procederà esplicitando tutti i sottoassiemi generati, partendo da quelli all'ultimo livello e sostituendo al posto del nome di ogni sottoassieme tutte le sottosequenze generate che gli competono.

Così facendo, l'approccio fornirà tutte le possibili sequenze di assemblaggio corrispondenti alla moltiplicazione del numero dei piani di assemblaggio relativi a tutti i sottoassiemi generati ad ogni passaggio. Avendo così a disposizione tutta la serie di sequenze generate, coerenti con la teoria applicata, l'utente avrà la possibilità di poter filtrare le migliori tra queste (o semplicemente solo quelle fattibili) attraverso parametri qualitativi o tecnologici relativi alla linea di assemblaggio.

## 2.4 Implementazione dell'algoritmo iterativo

Il metodo di generazione delle sequenze di assemblaggio appena esposto, fino ad ora è stato presentato come se fosse un metodo applicativo da svolgere manualmente, passo per passo. Così facendo, nonostante lo stesso criterio porti sempre ad almeno una soluzione valida (R. Viganò and G. O. Gomez, 2011), la sua esecuzione manuale e la conseguente elaborazione dei risultati risulterà molto lunga e tediosa, soprattutto al crescere del numero di parti costituenti l'assieme. Nonostante questo, il metodo presenta tutte le caratteristiche necessarie per una successiva implementazione automatica al calcolatore; questo è il motivo per cui vengono impiegati dei criteri di scelta numerici basati sul calcolo di indici caratteristici. Una volta esposto in maniera esaustiva il funzionamento del metodo iterativo considerato, si vuole sviluppare per esso una routine in grado di svolgere autonomamente tutti i passaggi richiesti. A tal scopo, si è scelto di implementare un codice attraverso il linguaggio di programmazione ad oggetti. La Figura 2.22 e la Figura 2.23 mostrano due possibili schermate ottenute durante l'esecuzione del software sviluppato. Di seguito verrà brevemente descritta la struttura dati utilizzata e solamente la struttura del relativo codice sorgente sviluppato; rimandando ogni ulteriore chiarimento al codice stesso, che è stato a tal proposito commentato al suo interno.

### 2.4.1 Struttura dati

Come più volte affermato, il metodo applicato richiede come input iniziale la totalità dei collegamenti presenti nell'assieme in esame; questi vengono forniti all'algoritmo sotto forma di un file di testo generato tramite l'interrogazione del software CAD su cui si è sviluppato il modello. La prima attività svolta dal codice sorgente sarà proprio quella di esaminare, sfogliando riga per riga, il file di testo contenente i collegamenti e compilare una prima classe di oggetti (che assumerà il primo livello), che prenderà il nome di *DataCollection*, contenente tutte le parti presenti (*Components*) e tutti i collegamenti tra di esse (*Connections*). Nella stessa classe di oggetti verranno allocati in seguito i cicli che vengono individuati e verranno pertanto effettuati i calcoli di classificazione dei cicli e dei nodi. In Figura 2.24 viene riassunta, tramite l'uso di un organigramma, l'intera struttura dati utilizzata dall'algoritmo. Dal momento che il metodo prevede la riduzione e la successiva rielaborazione della struttura dati creata, si è deciso di disporre la stessa su diversi livelli (*DataLevel*) e sottolivelli (*DataSublevel*), caratterizzando ognuna di esse univocamente, in modo da poter richiamare e visualizzare volta per volta i risultati ottenuti ad ogni passo di riduzione.

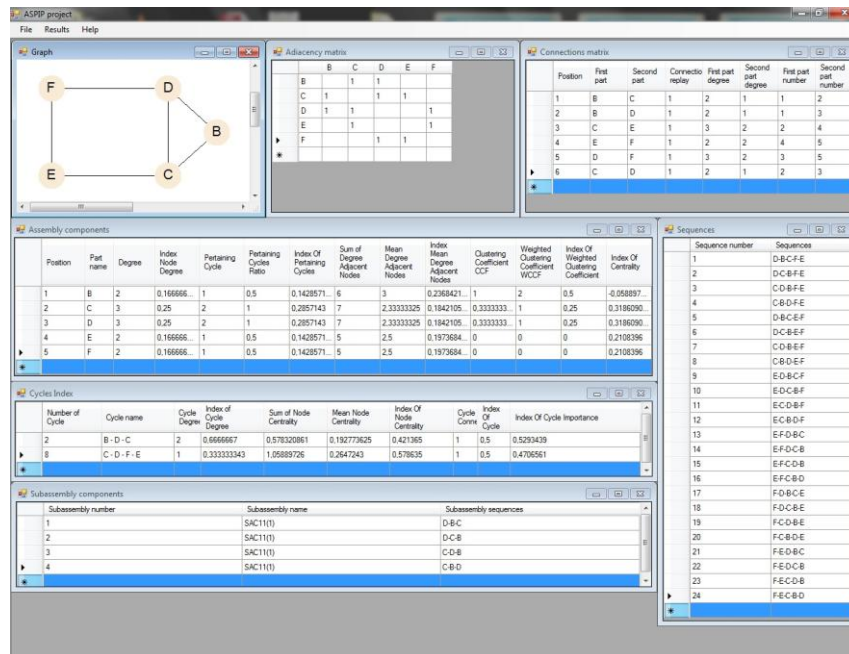


Figura 2.22 Schermata ottenuta durante l'esecuzione del software sull'esempio *Valvola Europa* (Capitolo Validazione dell'approccio automatico proposto).

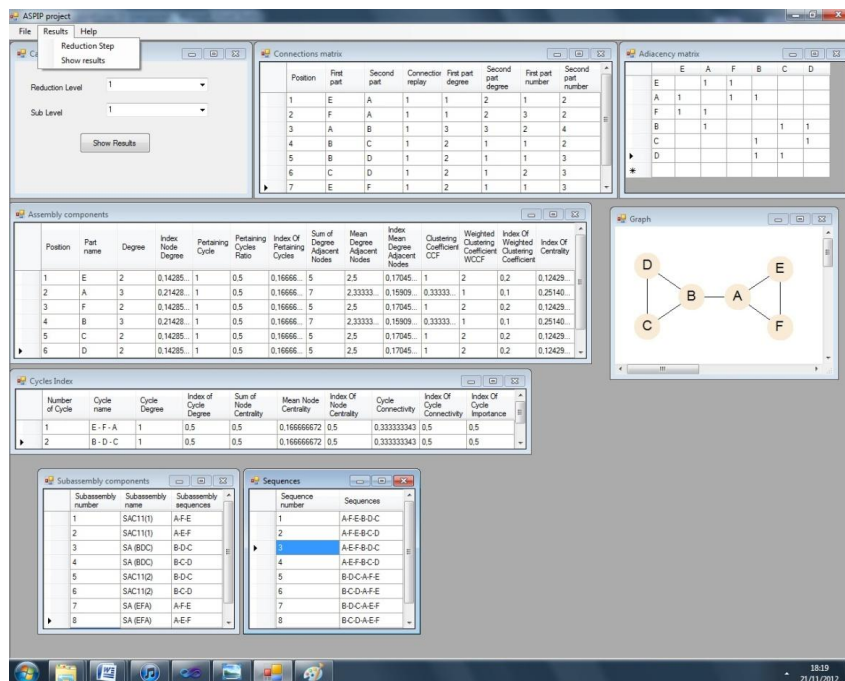


Figura 2.23 Schermata ottenuta durante l'esecuzione del software sull'esempio *Casistica N* (Capitolo Validazione dell'approccio automatico proposto).

Così facendo, ogni qualvolta un nuovo livello di riduzione verrà creato, su di esso verranno richiamate le routine di identificazione dei cicli e di calcolo degli indici dei cicli e dei nodi. Ogni livello ed ogni sottolivello indicheranno un passo di riduzione dove, il primo livello indicherà la struttura dell'assieme di partenza mentre l'ultimo livello (o gli ultimi livelli) indicheranno le riduzioni finali che sono state effettuate. Considerato che, come annunciato in precedenza, esistono dei casi in cui possono esistere due possibili riduzioni allo stesso livello, verrà usato in ausilio al livello del sistema un indice di sottolivello all'interno dello stesso livello. Un discorso differente può essere fatto per quanto riguarda la classe di dati temporanea, identificata dal nome *Temp* (dall'inglese Temporary), questa verrà creata per poter riapplicare la routine di riduzione in maniera ricorsiva solamente sui sottoassiemi individuati, senza chiamare in causa i livelli effettivi di riduzione, al fine di ottenere tutte le sottosequenze di montaggio dell'assieme. Per quanto riguarda le sequenze e le sottosequenze degli assiemi ridotti, queste vengono memorizzate al di fuori dei vari livelli di *DataCollection*, dal momento che una volta individuati, rimarranno univoci e non verranno ulteriormente elaborati.

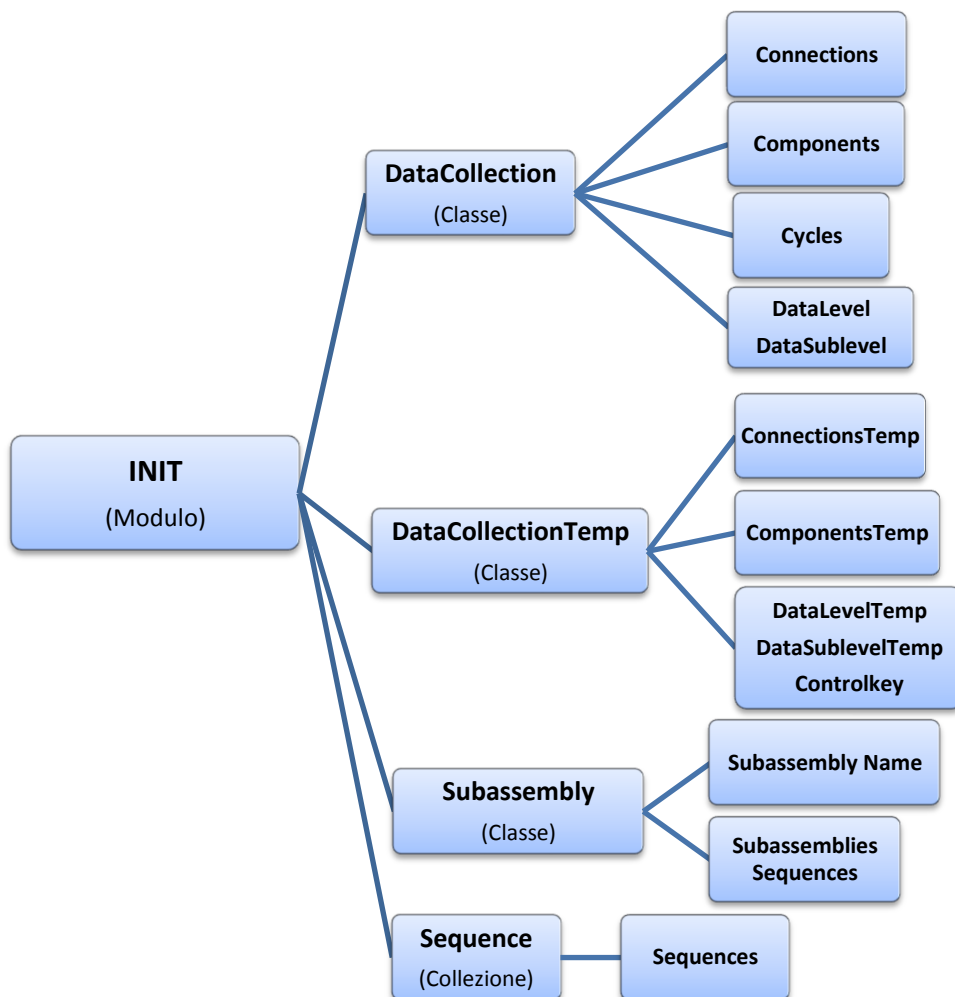


Figura 2.24 Organigramma delle strutture dati impiegate nell'algoritmo.

Nonostante questo, sia le sequenze che i sottoassiemi e le relative sottosequenze verranno anch'esse disposte sotto forma di classi dal momento che anche la loro compilazione risulterà ricorsiva. All'interno della classe sottoassiemi (*Subassembly*) verrà predisposto uno spazio di memoria al nome del sottoassieme (*Subassembly Name*) dal momento che questo risulterà utile nella compilazione delle sequenze finali.

Allo stesso modo, le parti, i collegamenti ed i cicli individuati saranno memorizzati secondo una ulteriore classe che conterrà tutte le informazioni necessarie nei calcoli e nelle scelte di classificazione numerica. In Figura 2.25 viene mostrata sinteticamente la classe di dati utilizzata per ognuna di esse.

COMPONENTS	CONNECTIONS	CYCLES
<ul style="list-style-type: none"><li>•Number</li><li>•Component</li><li>•Node Degree</li><li>•<math>I_{ND}</math></li><li>•Pertaining cycles</li><li>•<math>I_{PC}</math></li><li>•<math>DAN</math></li><li>•<math>I_{DAN}</math></li><li>•<math>WCCF</math></li><li>•<math>I_{CCF}</math></li><li>•<math>I_{NC}</math></li></ul>	<ul style="list-style-type: none"><li>•Number</li><li>•Part1</li><li>•Part2</li><li>•Replay</li><li>•Node Degree1</li><li>•Node Degree2</li></ul>	<ul style="list-style-type: none"><li>•Number</li><li>•Nodes components</li><li>•Name</li><li>•Cycle degree</li><li>•<math>I_{CD}</math></li><li>•<math>I_{CC}</math></li><li>•<math>I_{NC}</math></li><li>•<math>I_{CI}</math></li></ul>

Figura 2.25 Descrizione delle classi di Components, Connections e Cycles.

Ogni volta che verrà allocato uno spazio di memoria per uno di questi elementi, automaticamente verrà impegnato uno spazio ad ognuno dei relativi parametri sopradescritti, in modo da poter richiamare all'interno del codice gli stessi con estrema facilità.

### 2.4.2 Struttura del codice sorgente

Avendo implementato il metodo attraverso il linguaggio di programmazione Visual<sup>®</sup>, la routine principale del codice sorgente sarà riferita ad un *form* cardinale che permetterà l'interazione tra l'utente e il codice stesso.

I *form* sono l'unità base di un'applicazione e di conseguenza è fondamentale fare molta attenzione alla loro funzione e progettazione. Un *form* è una sorta di foglio o di lavagna pulita, in cui uno sviluppatore inserisce controlli, al fine di creare un'interfaccia utente e un codice per la modifica dei dati. A tale scopo, in Visual Studio<sup>®</sup> è presente un ambiente di sviluppo integrato (IDE, Integrated Development Environment) per facilitare la creazione del codice. Completando la funzionalità di tali controlli con il codice creato, è possibile sviluppare velocemente e senza difficoltà le soluzioni desiderate. In questo caso il *form* a cui si farà riferimento per strutturare il programma sarà il *form* principale *Main*. Al suo interno verranno richiamate tutte le istruzioni necessarie per svolgere le attività richieste dall'utente (chiamate eventi) attraverso l'interazione di quest'ultimo con i vari oggetti presenti nella scena. Facendo riferimento all'obiettivo preposti, ossia quello della generazione delle sequenze secondo la teoria di riduzione sopra esposta, il

*form* principale *Main* istituito, sarà lo strumento utile per il caricamento e la gestione dei file usati e dei *form* secondari di visualizzazione dei risultati. La schematizzazione di Figura 2.26 vuole, a tal proposito, rappresentare una descrizione riepilogativa del codice contenuto all'interno del *form Main*. Così facendo, ogni qualvolta che l'utente interagirà con uno degli oggetti (in questo caso le finestre *Open File*, *Close*, *Exit* ecc.) il codice avvierà una serie di attività relative a quel determinato oggetto. I *form* secondari invece, verranno creati all'interno del *Main* stesso e serviranno a visualizzare i risultati ottenuti ad ogni passaggio di riduzione ed in particolare rappresenteranno:

- La matrice di adiacenza binaria.
- Il grafo non orientato rappresentativo dell'assieme.
- La lista delle connessioni e dei parametri legati ad esse.
- La lista delle parti e i relativi indici caratteristici.
- La lista dei cicli e i relativi indici caratteristici.
- La lista dei sottoassiemi generati.
- La lista delle sequenze totali generate.

Ogni volta che verranno richiamati i risultati ad un dato livello, i *form* appena enunciati verranno generati all'interno del *form* principale. Oltre ai *form* sopracitati, esistono inoltre degli ulteriori *form* "di servizio" usati per interagire con l'utente quali il *form* di errore, che compare in caso di un disservizio del software, o quali il *form* di interrogazione del livello e del sottolivello dei risultati da richiamare.

Come si può facilmente comprendere dallo schema di Figura 2.26, il codice viene organizzato secondo una gerarchia di routine secondarie, o subroutine, necessarie per lo sviluppo completo del procedimento di riduzione applicato.

In questo contesto verranno esposte sinteticamente solo le subroutine più rilevanti chiamate dal codice sorgente, ed in particolare:

1. La subroutine di individuazione dei cicli indotti e dei calcoli di ranking.
2. La subroutine di riduzione iterativa dell'assieme.
3. La subroutine di generazione delle sequenze totali di assemblaggio.

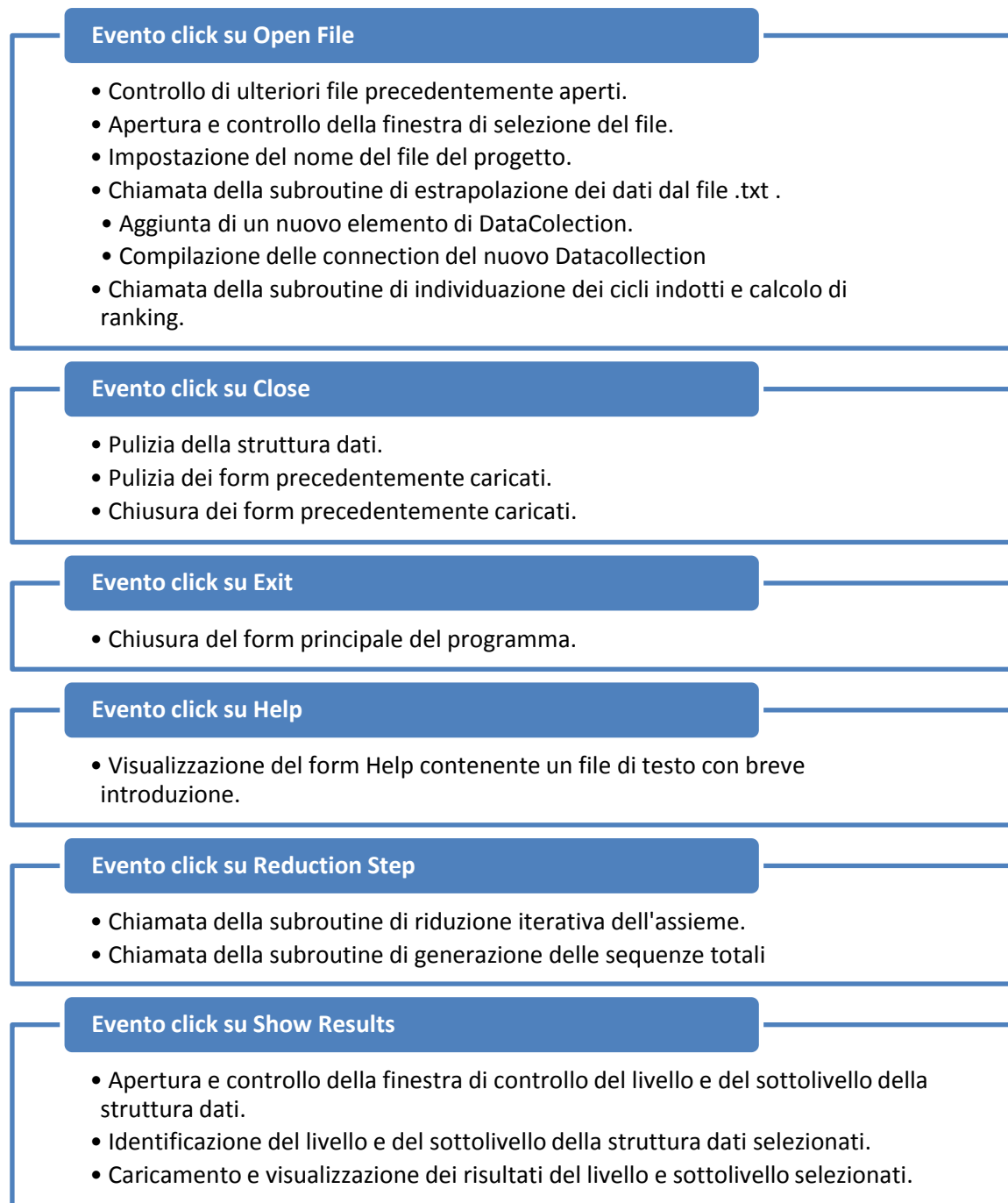


Figura 2.26 Descrizione sintetica della classe principale Main del codice sorgente.



### ***Individuazione dei cicli indotti e calcolo di ranking***

La subroutine che individua i cicli indotti ed esegue il calcolo di ranking svolge un ruolo fondamentale per l'algoritmo, in quanto fornisce i criteri di scelta per l'applicazione del metodo proposto. Questa porzione di codice verrà richiamata ogni qualvolta sarà necessario individuare dei cicli indotti o sarà necessario classificare l'insieme dei nodi e dei cicli. La subroutine richiede in entrata la sola classe dati di tipo `DataCollection` compilata dei soli collegamenti presenti ad un dato livello. Proprio per questo motivo, ogni volta che questa routine verrà impiegata nel codice principale, la forma della sua chiamata sarà del tipo:

`Calculus (ASPIPData)`

Dove *ASPIPData*, il parametro richiesto in ingresso dalla funzione *Calculus*, altro non è che il nome assegnato all'istanza creata per la classe di oggetti di tipo *DataCollection*. Quindi la struttura della subroutine in considerazione non sarà altro che una funzione pubblica, all'interno di un modulo, del tipo:

```
Public Function Calculus ( ByRef ASPIPData As DataCollections) As Boolean
```

Senza dover rendere esplicito l'intero codice della routine *Calculus*, basterà asserire che le principali attività svolte al suo interno consistano sommariamente in:

- 1) definizione della lista delle parti presenti.
- 2) Generazione del grafo.
- 3) ***Individuazione dei cicli indotti.***
  - a) Creazione dello schema ad albero corrispondente.
  - b) Applicazione del metodo di Greenlaw per l'identificazione dei cicli indotti mediante l'analisi dello schema ad albero.
  - c) Eliminazione dei cicli multipli (dal momento che il metodo può individuare più volte lo stesso ciclo).
- 4) ***Calcolo degli indici di ranking dei nodi.***
  - a) Calcolo degli indici del grado del nodo,  $I_{ND}$ .
  - b) Calcolo degli indici della media dei gradi dei nodi adiacenti,  $I_{DAN}$ .
  - c) Calcolo degli indici di pertinenza ai cicli,  $I_{PC}$ .
  - d) Calcolo degli indici di Clustering,  $I_{CCF}$ .
  - e) Calcolo degli indici totali di centralità dei nodi,  $I_{CT}$ .

5) **Calcolo degli indici di ranking dei cicli.**

- a) Calcolo degli indici del grado del ciclo,  $I_{CD}$ .
- b) Calcolo degli indici di connettività del ciclo,  $I_{CC}$ .
- c) Calcolo degli indici di centralità dei nodi del ciclo,  $I_{NC}$ .
- d) Calcolo degli indici di importanza del ciclo,  $I_{CI}$ .

E' rilevante notare che tutti i calcoli effettuati per gli indici si basano sui soli dati presenti nella struttura *ASPIPData*, nella forma *DataCollection*, contenente tutte le informazioni necessarie quali i collegamenti presenti, i nodi presenti ed i cicli individuati ad ogni livello.

**Riduzione iterativa dell'assieme**

La riduzione iterativa dell'assieme costituisce il cuore del metodo applicato, ossia la parte decisionale e iterativa su cui si generano le sequenze. Come per la parte di individuazione dei cicli e di calcolo degli indici, questa sarà costituita da una subroutine, all'interno di un modulo, che richiederà come dati in ingresso il livello ed il sottolivello della struttura dati *ASPIPData* precedentemente ricavata. Quindi ogni chiamata della stessa subroutine all'interno del codice sarà nella seguente forma:

```
Call Reduction ( level, sublevel )
```

Allo stesso modo, l'intestazione della medesima subroutine si presenterà nella forma:

```
Public Sub Reduction( ByVal levelkey As Int32, ByVal sublevelkey As Int32)
```

In Figura 2.27 viene mostrato il corrispettivo diagramma di flusso che riporta le principali attività svolte. Così, alla prima iterazione, la subroutine riceverà in ingresso il primo livello ed il primo sottolivello della struttura dati presente, ossia quella che costituisce il sistema di partenza. Dopodiché verrà verificata la possibilità di fare una riduzione, ed in particolare se questa riguarderà un nodo od un ciclo. Ogni volta che una nuova riduzione viene effettuata, e quindi una nuova struttura dati viene creata, la subroutine verifica la possibilità di fare ulteriori riduzioni o meno. Questo procedimento iterativo proseguirà fino a che tutte le strutture dati non raggiungano il loro ultimo livello di riduzione.

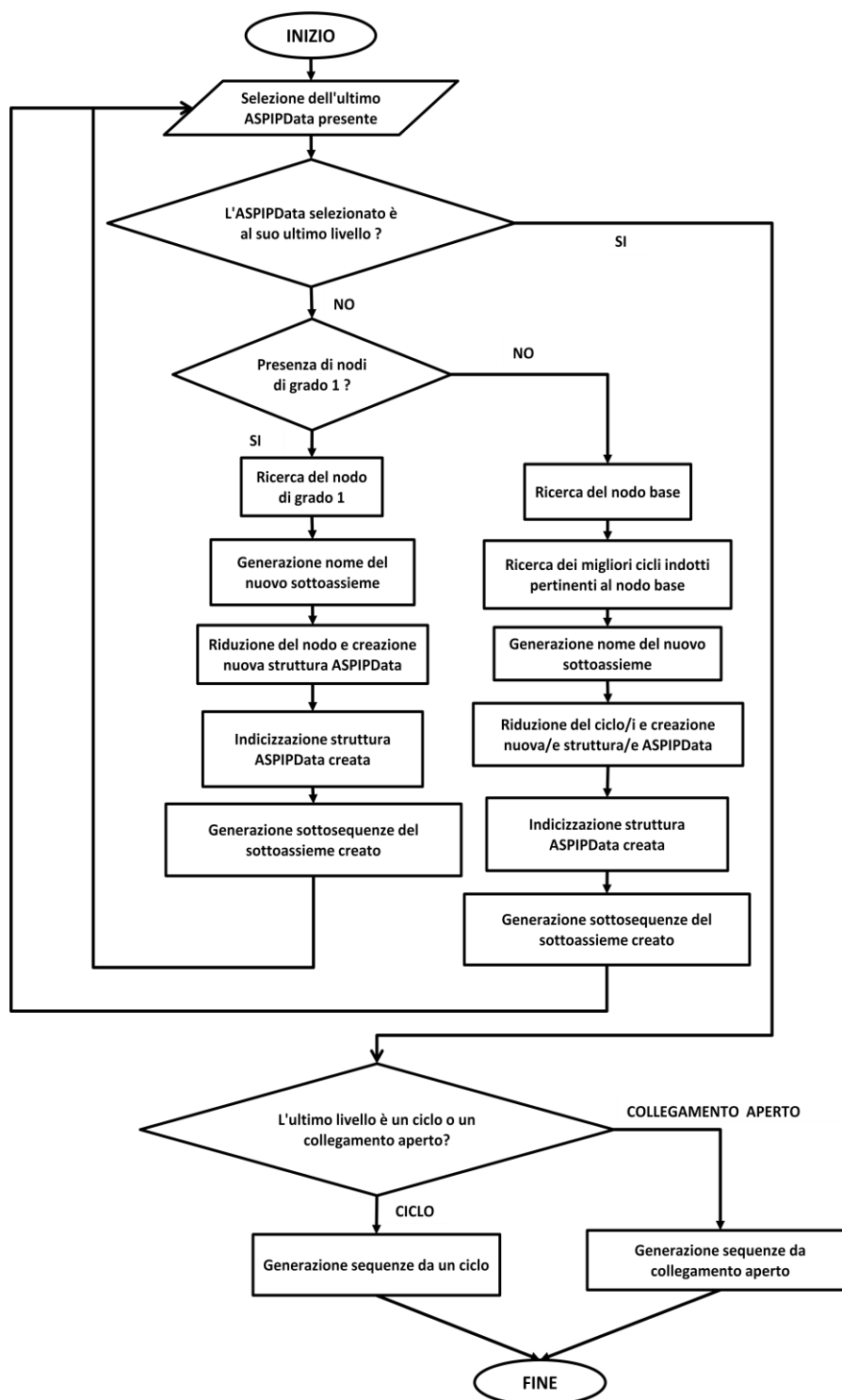


Figura 2.27 Diagramma di flusso rappresentativo della subroutine *Reduction*.

Bisogna specificare che per ultimo livello di riduzione si intende il soddisfacimento di una delle due seguenti condizioni sul sistema:

- Presenza di un solo collegamento restante.
- Presenza di un solo ciclo indotto restante.

### *Generazione della totalità di sequenze d'assemblaggio*

Una volta concluso il processo iterativo di riduzione, sarà infine necessario esplicitare all'interno delle sequenze generate, tutti i sottoassiemi generati volta per volta durante le relative riduzioni. Infatti, una volta noto il sottoassieme al suo ultimo livello, si potrebbe dire che le sequenze relative siano già state individuate, anche se in maniera implicita. Nonostante questo, con lo scopo di poter successivamente valutare in maniera chiara e comprensibile le soluzioni ottenute, si è pensato di esplicitare tutte le sequenze risultanti attraverso tutte le parti presenti nell'assieme di partenza. Per farlo è bastato considerare ogni sequenza sostituendo al nome del sottoassieme presenti in essa le relative sottosequenze precedentemente generate.

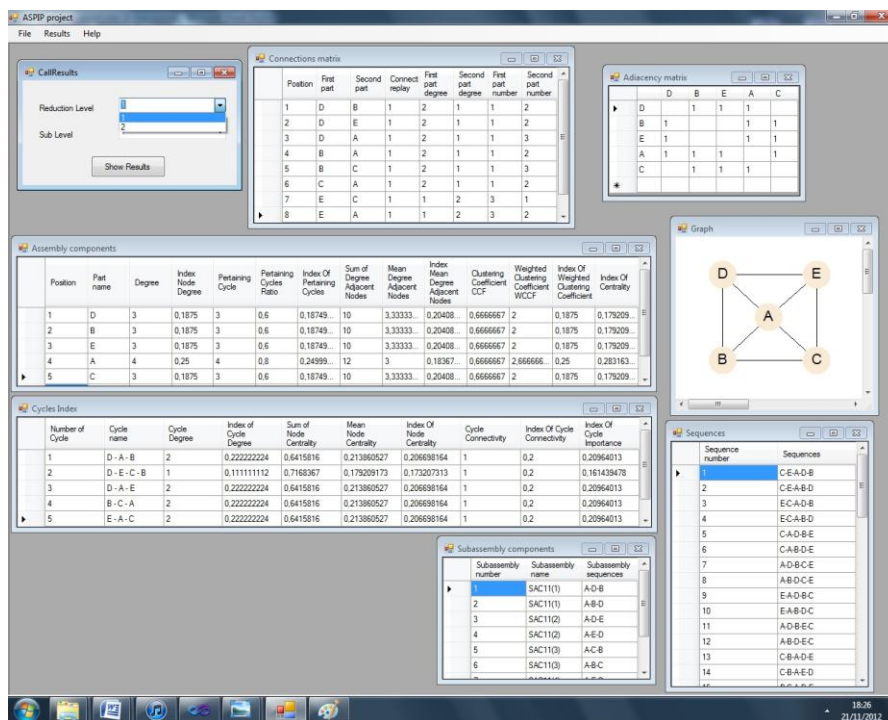


Figura 2.28 Schermata ottenuta durante l'esecuzione del software sull'esempio Casistica P (Vedi Capitolo Validazione dell'approccio automatico proposto).


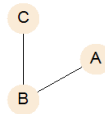
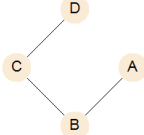
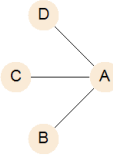
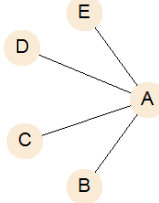
## **CAPITOLO 3**

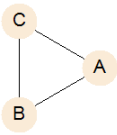
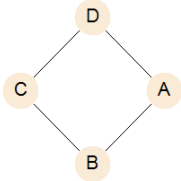
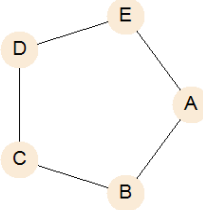
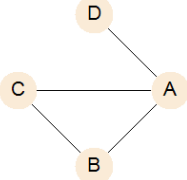
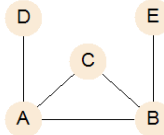
# **VALIDAZIONE DELL'APPROCCIO AUTOMATICO PROPOSTO**

### **3.1 Validazione attraverso casistiche base ricorsive**

Una prima validazione del metodo automatico qui proposto è stata effettuata facendo riferimento a delle casistiche molto semplici, ma che contraddistinguono al meglio le principali situazioni basilari in cui si può incorrere durante i vari passaggi di riduzione di sistemi più complessi. Proprio per tale motivo, quest'ultime sono state utilizzate non solo nella validazione finale del metodo, ma anche nelle validazioni intermedie man mano che l'algoritmo è stato sviluppato, verificando che lo stesso producesse risultati coerenti. La Tabella 3.1 elenca tutte le ipotetiche casistiche considerate; per completezza viene riportato insieme ad esse il numero di sottoassiemi che è necessario generare al fine di ottenere la completezza delle soluzioni con questo metodo ed il numero di sequenze che vengono generate per ogni caso. Nonostante vengano considerate molteplici casistiche, in linea di massima si può affermare che tutte queste possibilità possano venire raggruppate in tre categorie: i grafi semplici lineari ad  $n$  elementi, i grafi ramificati senza cicli indotti ad  $n$  elementi ed i grafi complessi costituiti da uno o più cicli indotti chiusi su se stessi.

Tabella 3.1 Elenco delle casistiche basi ricorsive.

INDICE	GRAFO EQUIVALENTE	ELEMENTI (N) COLLEGAMENTI (L) CICLI (C)	NUMERO DI SOTTOASSIEMI GENERATI	NUMERO DI SEQUENZE OTTENUTE
<b>Nessun ciclo, grado dei nodi presenti <math>\leq 2</math></b>				
<b>A</b>		$N = 2$ $L = 1$ $C = 0$	<b>0</b>	<b>2</b>
<b>B</b>		$N = 3$ $L = 2$ $C = 0$	<b>2</b>	<b>4</b>
<b>C</b>		$N = 4$ $L = 3$ $C = 0$	<b>6</b>	<b>6</b>
<b>Nessun ciclo, grado dei nodi presenti <math>\geq 1</math></b>				
<b>D</b>		$N = 4$ $L = 3$ $C = 0$	<b>9</b>	<b>12</b>
<b>E</b>		$N = 5$ $L = 4$ $C = 0$	<b>76</b>	<b>48</b>

<b>1 ciclo, grado dei nodi presenti = 2</b>				
<b>F</b>		$N = 3$ $L = 3$ $C = 1$	<b>0</b>	<b>6</b>
<b>G</b>		$N = 4$ $L = 4$ $C = 1$	<b>0</b>	<b>16</b>
<b>H</b>		$N = 5$ $L = 5$ $C = 1$	<b>0</b>	<b>40</b>
<b>1 ciclo, grado dei nodi presenti ≥ 1</b>				
<b>I</b>		$N = 4$ $L = 4$ $C = 1$	<b>1</b>	<b>6</b>
<b>J</b>		$N = 5$ $L = 5$ $C = 1$	<b>4</b>	<b>6</b>

2 cicli, grado dei nodi presenti $\geq 1$				
<b>K</b>		$N = 5$ $L = 6$ $C = 2$	<b>4</b>	<b>24</b>
<b>L</b>		$N = 4$ $L = 5$ $C = 2$	<b>8</b>	<b>16</b>
<b>M</b>		$N = 5$ $L = 6$ $C = 2$	<b>4</b>	<b>24</b>
<b>N</b>		$N = 6$ $L = 7$ $C = 2$	<b>6</b>	<b>24</b>
Casi complementari				
<b>O</b>		$N = 6$ $L = 7$ $C = 2$	<b>5</b>	<b>24</b>
<b>P</b>		$N = 5$ $L = 8$ $C = 5$	<b>8</b>	<b>48</b>



Gli esempi appena presentati sono stati in seguito impiegati nella metodologia automatica creata ed i risultati sono stati esaminati verificando la veridicità delle soluzioni ottenute, da un punto di vista puramente teorico.

Bisogna considerare che per queste casistiche, a parte la casistica  $K$ , non si può parlare di un riscontro pratico e quindi non si può valutare la fattibilità delle sequenze di assemblaggio; questo perché i grafi sono puramente inventati dall'autore e non si riferiscono ad esempi reali. L'esempio  $K$  invece, può essere considerato un primo esempio pratico dal momento che la disposizione dei suoi nodi nel grafo è del tutto analoga a quella del grafo ricavato per l'assieme della valvola Europa che presenta una simmetria assiale; questa simmetria si riflette anche sul grafo dei collegamenti.

Nel caso degli esempi  $F$ ,  $G$  e  $H$ , ossia quelli rappresentanti un unico ciclo indotto composto da più elementi, lecitamente nessun sottoassieme viene generato dal momento che l'assieme in questione risulta già al suo ultimo livello di riduzione. Quindi le sottosequenze generate per questi tre grafi, saranno direttamente le sequenze possibili che il metodo genererà. In particolare, dal momento che ogni nodo costituente è un nodo piattaforma, secondo quanto detto precedentemente la totalità della soluzione sarà in numero pari a:

$$k \times 2^{(k-2)} \text{ possibili sequenze generate}$$

Dove ancora una volta,  $k$  rappresenta la lunghezza del ciclo in questione e non ha nulla in comune con la lettera  $K$  che indica il nominativo dell'esempio. Dal momento che, in questi casi, il numero di nodi considerati come piattaforma sarà equivalente al numero di nodi componenti l'assieme, si otterrà che:

- per il grafo  $F$ , costituito da 3 elementi, 6 possibili sequenze verranno generate.
- Per il grafo  $G$ , costituito da 4 elementi, 16 possibili sequenze verranno generate.
- Per il grafo  $H$ , costituito da 5 elementi, 40 possibili sequenze verranno generate.

Inoltre, in altre casistiche qui considerate si presenta la situazione particolare in cui più cicli indotti con lo stesso indice di importanza sono pertinenti allo stesso nodo base. In questo specifico caso, ogni possibile strada di riduzione dovrà venire considerata come valida e quindi verrà portata avanti separatamente, generando a sua volta delle ulteriori possibili soluzioni. L'esempio che più caratterizza questa situazione è rappresentato dalla casistica  $P$  dove il nodo che assume il ruolo di piattaforma è il nodo  $A$  ed i cicli individuati a lui pertinenti sono 4 ed in particolare: il ciclo  $D-A-E$ , il ciclo  $E-A-C$ , il ciclo  $D-A-B$  ed il ciclo  $B-A-C$ .

In questo particolare esempio, quattro diverse strade di riduzione verranno prese in considerazione e sviluppate singolarmente, anche se, come si vedrà in seguito, molte delle soluzioni dei diversi percorsi di riduzione seguiti saranno le medesime e verranno eliminate. Una ulteriore nota in merito viene dedicata ad i casi in cui  $n$  nodi di grado uno sono collegati tutti ad un unico nodo centrale; ne sono un esempio caratteristico i casi *B,D* ed *E*. Questi esempi sottolineano il fatto che nonostante non sia presente nessun ciclo, e quindi nessun nodo base e nessuna possibilità di avere più cicli pertinenti, esistono più strade di riduzione in base a quale nodo di grado uno ridurre per prima. I casi in questione non sono altro che una generalizzazione di un grafo lineare composto da 3 elementi, dove in questo caso l'elemento centrale è collegato a più di due nodi di grado uno. Questi verranno trattati in maniera del tutto analoga ai casi di più cicli indotti con lo stesso indice di importanza, portando avanti tutte le soluzioni di riduzione possibili ed eliminando in seguito le soluzioni con una molteplicità.

#### 3.1.1 Esempio Valvola Europa

Riprendendo in considerazione la casistica denominata *K*, rappresentante il grafo dei collegamenti non orientato dell'assieme meccanico valvola Europa rappresentata in Figura 3.1; si desidera valutare passo per passo l'elaborazione compiuta dal calcolatore grazie al metodo automatico proposto.

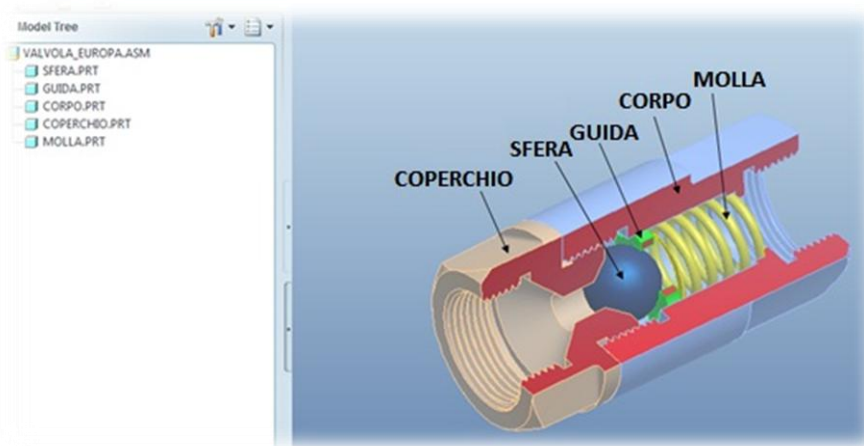


Figura 3.1 Esempio Valvola Europa

1. Nel programma di modellazione CAD, il prodotto viene creato vincolando ogni parte alle altre garantendo la condizione di equilibrio stabile per ognuna di esse (vedi Figura 3.1).
2. Dal programma di modellazione CAD, le informazioni di contatto ed interferenza vengono estrapolate in maniera automatica dal programma stesso al fine di poter definire l'elenco delle connessioni presenti e il relativo grafo non orientato mostrato in Figura 3.2.

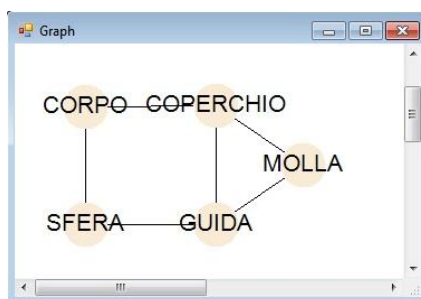


Figura 3.2 Grafo non orientato della Valvola Europa

3. La prima attività da svolgere sulle informazioni di contatto è la riduzione dei nodi con grado pari a uno e la composizione del relativo sottoinsieme utile all'identificazione delle sequenze. In questo caso particolare, non sono presenti nodi con grado pari a uno.
4. Successivamente, i cicli indotti vengono identificati dall'algoritmo e per ogni nodo e ciclo presente vengono calcolati gli indici di importanza e di centralità con lo scopo di poter classificare gli stessi.

In questo caso, come mostrato nelle Figura 3.3 e nella Figura 3.4, il software identifica due cicli indotti ed esegue il calcolo di ranking per tutti i cicli e tutti i nodi individuati.

	Position	Part name	Degree	Index Node Degree	Pertaini Cycle	Pertaini Cycles Ratio	Index Of Pertaini Cycles	Sum of Degree Adjacer Nodes	Mean Degree Adjacer Nodes	Index Mean Degree Adjacer Nodes	Clusteri Coeffici CCF	Weighti Clusteri Coeffici WCCF	Index Of Weighti Clusteri Coeffici	Index Of Centralit
1		SFERA	2	0,16...	1	0,5	0,14...	5	2,5	0,197...	0	0	0	0,210...
2		GUIDA	3	0,25	2	1	0,28...	7	2,33...	0,184...	0,333...	1	0,25	0,318...
3		CORPO	2	0,16...	1	0,5	0,14...	5	2,5	0,197...	0	0	0	0,210...
4		COPERCHIO	3	0,25	2	1	0,28...	7	2,33...	0,184...	0,333...	1	0,25	0,318...
5		MOLLA	2	0,16...	1	0,5	0,14...	6	3	0,236...	1	2	0,5	-0,05...

Figura 3.3 Calcolo degli indici di centralità per i nodi della Valvola Europa.

Number of Cycle	Cycle name	Cycle Degree	Index of Cycle Degree	Sum of Node Centrality	Mean Node Centrality	Index Of Node Centrality	Cycle Connectivity	Index Of Cycle Connectivity	Index Of Cycle Importance
1	SFERA - CORPO - COPERCHIO - GUIDA	1	0,333333...	1,05889726	0,2647243	0,578635...	1	0,5	0,4706561...
8	GUIDA - MOLLA - COPERCHIO	2	0,6666667	0,5783208	0,1927736	0,421365	1	0,5	0,5293439

Figura 3.4 Individuazione e calcolo degli indici di importanza nella Valvola Europa.

Si ottiene che il ciclo *GUIDA-MOLLA-COPERCHIO* è quello con il maggiore indice di importanza e sarà quindi il primo sottoassieme a venir generato, mentre i suoi nodi base (ossia i nodi pertinenti allo stesso ciclo con il maggiore indice di centralità dei nodi) saranno i nodi *GUIDA* e *COPERCHIO*.

Per questo sottoassieme dunque, le sequenze partiranno dai nodi base e saranno in numero pari a  $2 * 2^{(k-2)} = 2 * 2^{(3-2)} = 4$ , dove k indica la lunghezza del ciclo. La totalità delle soluzioni è elencata in Tabella 3.2.

Tabella 3.2 Sottosequenze del sottoassieme GUIDA-MOLLA-COPERCHIO.

NODO BASE	SOTTOSEQUENZE
GUIDA	GUIDA-COPERCHIO-MOLLA
	GUIDA-MOLLA-COPERCHIO
COPERCHIO	COPERCHIO-MOLLA-GUIDA
	COPERCHIO-GUIDA-MOLLA

- Il ciclo 2 viene quindi ridotto ad un unico elemento e secondo le regole di riduzione precedentemente espote, il grafo viene semplificato come rappresentato in Figura 3.5.

La riduzione dei cicli (sottoassiemi) e la definizione delle relative sottosequenze continua iterativamente fino a che non viene raggiunto l'ultimo livello di riduzione, che in questo caso corrisponde al ciclo rappresentato in Figura 3.5.

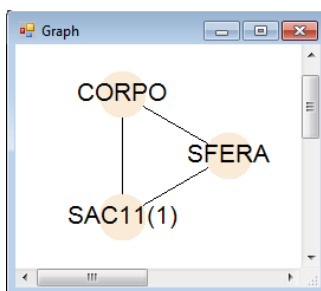


Figura 3.5 Grafo ridotto dell'esempio Valvola Europa

A questo punto, ogni nodo del ciclo viene considerato come un possibile nodo base (dal momento che i relativi indici di centralità assumono tutti lo stesso valore) e da ognuno di essi vengono generate le relative sotto sequenze che in questo caso saranno le sequenze finali di assemblaggio dell'assieme. In questo caso, ci sono 3 nodi base e quindi  $3 \cdot 2^{(k-2)} = 3 \cdot 2^{(3-2)} = 6$  sequenze vengono generate. Espandendo nuovamente le sequenze generate tramite la sostituzione del nome del sottoassieme con le relative sequenze si ottiene la totalità delle soluzioni teoriche ricavate per l'esempio in questione, ed in particolare avendo generato un solo sottoassieme a cui competono 4 sottosequenze ed avendo generato 6 sequenze totali, si avranno  $6 \cdot 4 = 24$  possibili sequenze che rispettano coerentemente la teoria di riduzione. La totalità delle sequenze ricavate viene rappresentata in Tabella 3.4.

Tabella 3.3 Sequenze generate implicite per l'esempio Valvola Europa

NODO BASE	SEQUENZE
CORPO	CORPO-SFERA-SAC11(1)
	CORPO- SAC11(1)-SFERA
SFERA	SFERA-CORPO- SAC11(1)
	SFERA- SAC11(1)-CORPO
SAC11(1)	SAC11(1)-CORPO-SFERA
	SAC11(1)-SFERA-CORPO

Tabella 3.4 Sequenze generate per l'assieme Valvola Europa.

NODO BASE	SEQUENZA IMPLICITA	SEQUENZA ESPLICITA	
CORPO	CORPO – SFERA – SAC11(1)	1	CORPO – SFERA – GUIDA – COPERCHIO – MOLLA
		2	CORPO – SFERA – GUIDA – MOLLA – COPERCHIO
		3	CORPO – SFERA – COPERCHIO – MOLLA – GUIDA
		4	CORPO – SFERA – COPERCHIO – GUIDA – MOLLA
	CORPO – SAC11(1)– SFERA	5	CORPO – GUIDA – COPERCHIO – MOLLA – SFERA
		6	CORPO – GUIDA – MOLLA – COPERCHIO – SFERA
		7	CORPO – COPERCHIO – MOLLA – GUIDA – SFERA
		8	CORPO – COPERCHIO – GUIDA – MOLLA – SFERA
SFERA	SFERA – CORPO – SAC11(1)	9	SFERA – CORPO – GUIDA – COPERCHIO – MOLLA
		10	SFERA – CORPO – GUIDA – MOLLA – COPERCHIO
		11	SFERA – CORPO – COPERCHIO – MOLLA – GUIDA
		12	SFERA – CORPO – COPERCHIO – GUIDA – MOLLA
	SFERA – SAC11(1)– CORPO	13	SFERA – GUIDA – COPERCHIO – MOLLA – CORPO
		14	SFERA – GUIDA – MOLLA – COPERCHIO – CORPO
		15	SFERA – COPERCHIO – MOLLA – GUIDA – CORPO
		16	SFERA – COPERCHIO – GUIDA – MOLLA – CORPO
SAC11(1)	SAC11(1)– SFERA – CORPO	17	GUIDA – COPERCHIO – MOLLA – SFERA – CORPO
		18	GUIDA – MOLLA – COPERCHIO – SFERA – CORPO
		19	COPERCHIO – MOLLA – GUIDA – SFERA – CORPO
		20	COPERCHIO – GUIDA – MOLLA – SFERA – CORPO
	SAC11(1)– CORPO – SFERA	21	GUIDA – COPERCHIO – MOLLA – CORPO – SFERA
		22	GUIDA – MOLLA – COPERCHIO – CORPO – SFERA
		23	COPERCHIO – MOLLA – GUIDA – CORPO – SFERA
		24	COPERCHIO – GUIDA – MOLLA – CORPO – SFERA

### 3.2 Validazione attraverso il caso di De Fazio & Whitney

In questo paragrafo, l'esempio trattato da De Fazio e Whitney (De Fazio, et al., 1987) (De Fazio, et al., 1989) (De Fazio, et al., 1999) riguardante l'assemblaggio della trasmissione di un automezzo pesante, che prende il nome di AFI (dall'acronimo inglese Assembly For Industry), viene analizzato con i 5 diversi approcci precedentemente visti. Gli stessi autori osservano che questo esempio rappresenta l'essenza dei problemi di assemblaggio in tutti i loro dettagli e nessuna caratteristica dello stesso è frutto della loro fantasia.

L'assieme presenta una geometria perfettamente rappresentabile attraverso una simmetria circolare intorno allo stesso asse di funzionamento rappresentato in Figura 3.6.

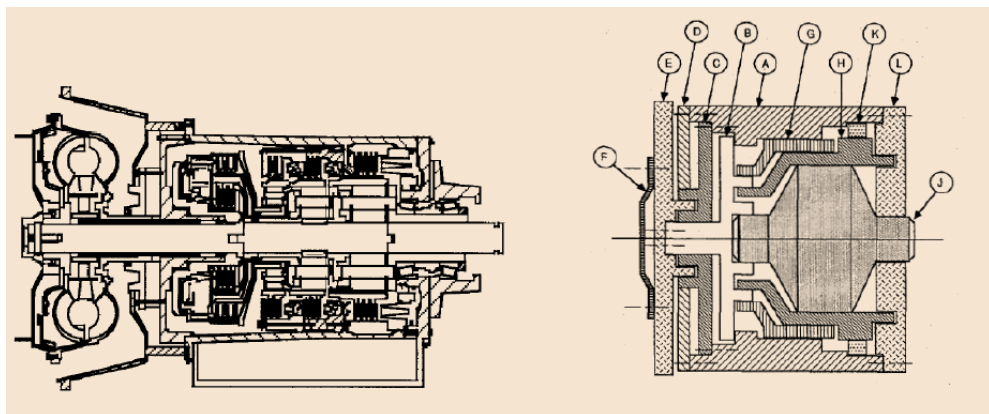


Figura 3.6 Esempio di Assembly for Industry (AFI) di De Fazio e Whitney.

Nella Figura 3.6 a sinistra viene rappresentato il disegno in sezione della trasmissione realistico, compreso di ruota posteriore mentre, nella stessa figura a destra, viene mostrata una possibile rappresentazione semplificata. Come si evince dalla stessa Figura 3.6 di destra sono presenti degli elementi di fissaggio come viti e dadi, ben visibili al loro asse di simmetria. Per semplicità di soluzioni e analisi, si suppone implicitamente la omissione di questi elementi di fissaggio, imponendo la condizione che, in fase di montaggio, se due di queste parti devono venire montate insieme, sarà necessario fissarle attraverso gli stessi elementi di fissaggio omessi.

Il relativo grafo dei collegamenti non orientato è mostrato in figura X, e viene ricavato direttamente dal disegno rappresentativo in maniera manuale senza troppe difficoltà. L'autore osserva che la parte K, avendo un grado di

libertà di translazione in direzione assiale, potrebbe entrare in contatto con la parte *L*. Nonostante ciò nel grafo relativo, rappresentato in Figura 3.7, non viene considerato tale contatto in quanto nessuna sequenza di assemblaggio valida che involva la parte *K* con la parte *L* potrebbe esistere ed essere valida senza chiamare in causa le parti *A* o *H*. Nello stesso modo in cui il metodo automatico è stato strutturato, l'esempio verrà analizzato partendo dalla totalità delle possibili combinazioni tra le parti, riducendo man mano le soluzioni possibili che potrebbero esistere chiamando in causa dei particolari tra i collegamenti.

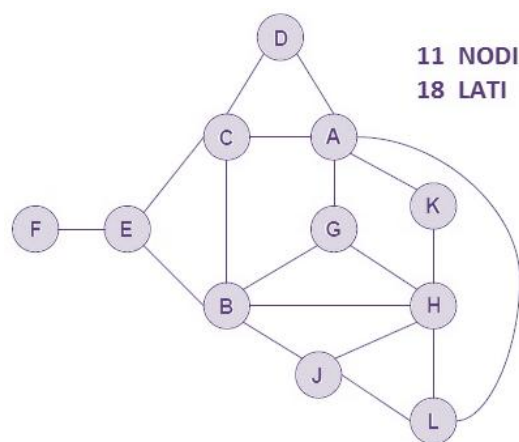


Figura 3.7 Grafo dei collegamenti dell'esempio AFI.

Così facendo, i primi due approcci, rappresentativi di tutte le possibili sequenze ottenibili attraverso la permutazione delle parti e la riduzione considerando il solo numero di collegamenti presenti, mostreranno solamente l'elevato numero di soluzioni che è possibile ottenere senza considerare nessun altro tipo di informazione estrapolabile dall'insieme. Entrambi gli approcci qui considerati, saranno analizzati con 10 nodi, dal momento che la riduzione del nodo *F* con il nodo *E*, non coinvolge le problematiche relative alla generazione delle sequenze di assemblaggio. La stessa ipotesi di semplificazione del problema viene assunta da Homem de Mello (Homem de Mello, et al., 1990) durante l'applicazione dei grafici AND/OR dello stesso esempio.

In questo contesto, solo il numero di soluzioni ottenibili attraverso l'esempio AFI verrà valutato con lo scopo di poter mettere in paragone i diversi metodi esposti rispetto a quello qui proposto. Per ulteriori informazioni quali la tipologia e l'elenco di sequenze ottenute si consiglia di fare riferimento alla stessa bibliografia. Per ultimo viene mostrata una prima approssimazione



delle soluzioni del problema attraverso l'applicazione della teoria dei grafi; in questo caso la ricerca dei tragitti hamiltoniani è svolta sia per il caso con 10 nodi che per il caso con 11 nodi applicando l'algoritmo di depth first search. A seguito di ciò, il nuovo approccio automatico sviluppato viene applicato al medesimo problema, mettendo in evidenza tutti i dati ed i passaggi svolti.

### 3.2.1 Tutte le sequenze possibili

Come primo avvicinamento al campo di soluzioni possibili, nessuna informazione riguardante le relazioni tra i componenti viene presa in considerazione, ma solamente il numero stesso di componenti con cui è possibile fare tutte le permutazioni possibili. Così facendo, si ottiene per 11 componenti una totalità pari a  $11! = 39'916'800$  combinazioni possibili; queste sarebbero il massimo numero di soluzioni che si potrebbe analizzare ed allo stesso tempo il limite del campo di ricerca per questi problemi. Oltre a queste combinazioni non potrebbe esistere un'ulteriore soluzione di arrangiamento differente degli elementi. Riducendo il nodo  $F$  con il suo unico nodo adiacente  $E$ , si ottiene che le soluzioni possibili, nel caso ora ci siano solamente dieci elementi, sarebbero pari a  $10! = 3'628'800$  soluzioni possibili. Proprio per la natura combinatoriale del problema, l'aggiunta di un solo elemento incrementa in maniera spaventosa il numero delle soluzioni possibili, in particolare quando il numero degli elementi in questione è considerevole.

### 3.2.2 Riduzione sul numero di collegamenti

La prima riduzione più naturale possibile del numero di soluzioni ottenute diventa quella di rispettare le "regole di collegamento" che persistono tra le parti secondo i collegamenti presenti nel grafo rappresentativo. Questo può essere fatto eliminando tutte quelle sequenze che mostrano un collegamento non presente nella lista dei collegamenti tra le parti. Facendo ciò, per questo particolare caso AFI, considerando il solo numero di 10 componenti (senza il nodo  $F$ ) si ottiene un totale di 449'376 soluzioni possibili. Si osserva quindi che la sola aggiunta delle informazioni di contatto preesistenti tra le parti porta ad una riduzione del 88% di soluzioni possibili sul numero totale di soluzioni ottenute precedentemente su 10 elementi attraverso le permutazioni. Nonostante questa enorme riduzione, si considera il campo delle soluzioni possibili ancora troppo vasto per un riscontro pratico.

### 3.2.3 Approccio bibliografico (metodo di De Fazio & Whitney)

De Fazio e Whitney spiegano che una ulteriore riduzione dell'ampiezza del campo di soluzioni può essere apportata mediante l'interrogazione di  $2 \times l = 36$  domande di precedenza all'utente, dove  $l = 18$  indica il numero di informazioni di contatto presenti. Una volta eseguite tutte le domande e fondate le regole di precedenza, è possibile implementare queste informazioni con un algoritmo al fine di generare delle soluzioni di assemblaggio fattibili, in accordo con le regole spiegate nel paragrafo 1.2.1.

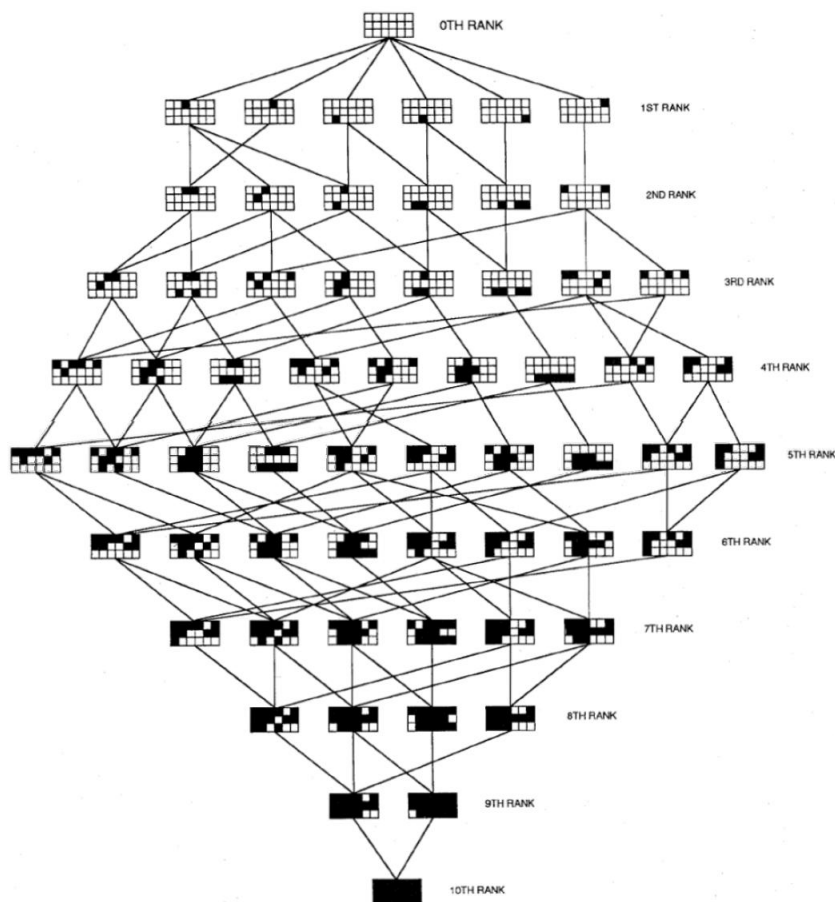


Figura 3.8 Rappresentazione grafica delle sequenze ottenute attraverso l'approccio De Fazio & Whitney.

Una volta create le sequenze, attraverso le precedenze sui collegamenti, è possibile rappresentare quest'ultime graficamente attraverso uno schema ad albero, come riportato nella Figura 3.8. L'esempio AFI in questione presenta

440 percorsi differenti partendo dal livello di selezione zero fino ad arrivare al decimo ed ultimo livello. A questo punto, De Fazio e Whitney propongono un'ulteriore riduzione di tutte le sequenze di assemblaggio fattibili applicando alcuni vincoli tecnici circa caratteristiche geometriche delle parti ottenendo infine 16 sequenze di assemblaggio fattibili rappresentate graficamente nello schema grafico di Figura 3.9.

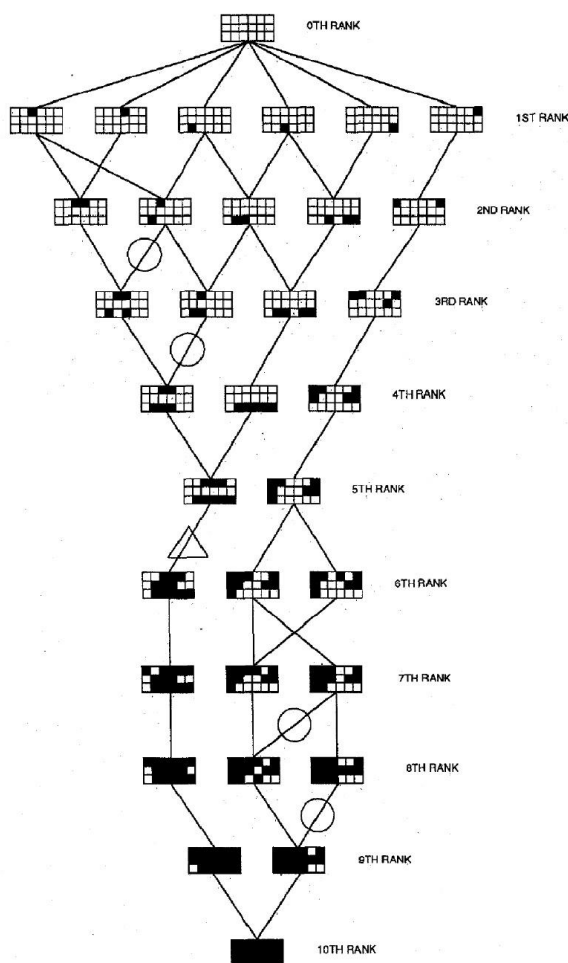


Figura 3.9 Rappresentazione grafica delle sequenze ottenute dopo la selezione attraverso vincoli tecnici all'esempio De Fazio & Whitney.

I segni a forma di cerchio e di triangolo presenti nello schema vogliono rappresentare un'ulteriore riduzione del numero di tragitti percorribili nelle sequenze dei collegamenti, a causa di compiti difficili da svolgere od infattibilità a livello pratico. Infine gli autori ottengono solo 2 sequenze di

assemblaggio fattibili, ma ottimali. Gli stessi inoltre concludono dicendo che il compito di generare 440 precedenze sui collegamenti ed il compito di doverle selezionare fino ad ottenere solo 16 sequenze valide ed infine due ottimali risulta comunque molto meno oneroso in termini di tempo e di costi che ottenere una sola sequenza facendo affidamento solamente sull'esperienza e su parametri ingegneristici tradizionali.

### 3.2.4 Approccio tramite la teoria dei grafi

Lo stesso esempio viene ora analizzato impiegando un algoritmo tratto dalla teoria dei grafi, ossia l'algoritmo di depth first search. Lo scopo è quello di trovare tutti i tragitti hamiltoniani presenti nel grafo dei collegamenti; questi percorsi hanno la peculiarità di passare per tutti i nodi una volta sola definendo così una sequenza lineare per l'assemblaggio.

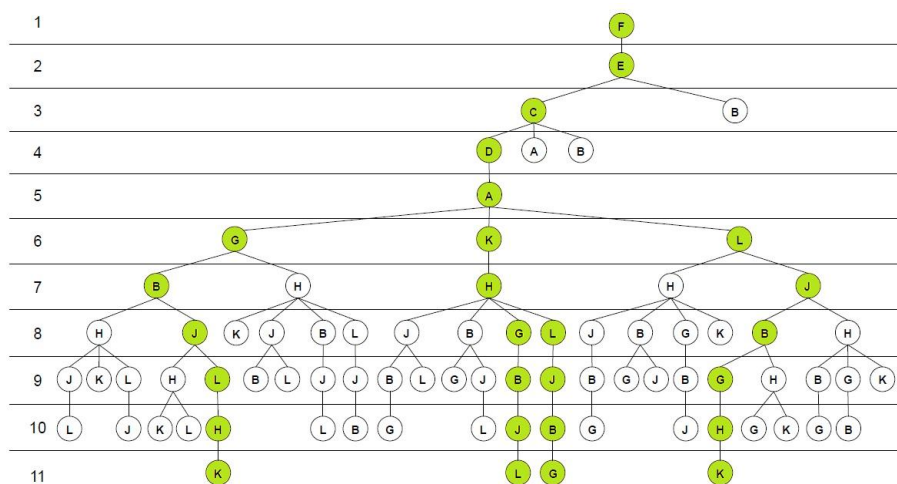


Figura 3.10 Ricerca dei tragitti Hamiltoniani per l'esempio di De Fazio e Whitney a 11 elementi.

Osservando che il problema della ricerca dei tragitti Hamiltoniani è un problema NP-completo e che non esiste una caratterizzazione del suo sviluppo, per il caso dell'esempio con 11 nodi, la ricerca deve per forza iniziare o finire dal nodo F. Una rappresentazione grafica dei risultati ottenuti applicando l'algoritmo DFS è mostrata in Figura 3.10. In questa situazione si hanno un numero totale di tragitti Hamiltoniani pari a 8, considerando che ognuno di essi può essere percorso in entrambe le direzioni. E' interessante notare che nessuno degli otto tragitti individuati rappresenta una sequenza di assemblaggio fattibile.

Così, con l'obiettivo di aumentare le possibilità di trovare delle soluzioni utili, viene riapplicato lo stesso algoritmo allo stesso esempio con 10 elementi (dove il nodo  $F$  è stato precedentemente ridotto). I risultati di questa operazione sono visibili in Figura 3.11.

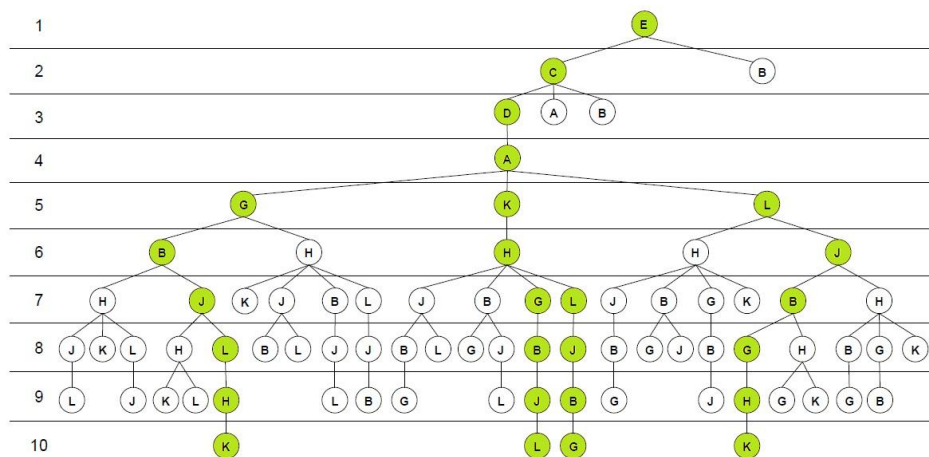


Figura 3.11 Ricerca dei tragitti Hamiltoniani per l'esempio di De Fazio e Whitney a 11 elementi.

Questa volta il numero di tragitti Hamiltoniani trovati dipende dal nodo con cui si decide di partire, in particolare si ottengono:

- 8 Tragitti Hamiltoniani partendo o finendo con il nodo  $E'$ .
- 18 Tragitti Hamiltoniani partendo o finendo con il nodo  $D$ .
- 26 Tragitti Hamiltoniani partendo o finendo con il nodo  $K$ .
- 18 Tragitti Hamiltoniani partendo o finendo con il nodo  $G$ .
- 8 Tragitti Hamiltoniani partendo o finendo con il nodo  $J$ .
- 6 Tragitti Hamiltoniani partendo o finendo con il nodo  $L$ .

Ovviamente, partendo o finendo da tutti gli altri nodi non elencati non si ottiene nessun tragitto Hamiltoniano valido. In questo modo si ottiene un totale di 84 tragitti Hamiltoniani, ma ciononostante nessuno di questi risulterà applicabile come una sequenza valida di assemblaggio.

Queste risultati ancora una volta sottolineano il fatto che questo approccio attraverso la teoria dei grafi non risulta utile nei problemi di pianificazione delle sequenze di assemblaggio, dal momento che ottenere una sequenza che involve tutte le parti di un assieme meccanico non significa ottenere una soluzione fattibile a livello pratico. Ancora una volta risulta la necessità di

dover creare dei criteri di scelta al fine di riuscire a scindere la totalità delle soluzioni fattibili da quel non realizzabili.

### 3.2.5 Metodologia automatica proposta

L'esempio caratteristico chiamato in causa da De Fazio e Whitney. Ossia la trasmissione di un automezzo pesante AFI risulta un ottimo modello per illustrare il funzionamento del software automatico creato per la generazione delle sequenze di assemblaggio fattibili. Di seguito verrà a tal proposito illustrato step-by-step il metodo impiegato sull'esempio utilizzato per i casi precedenti.

#### Identificazione dei contatti tra le parti

Come primo passaggio, si modella l'assieme meccanico attraverso un software di modellazione CAD, prendendo le caratteristiche geometriche necessarie del modello dalla bibliografia, e successivamente si interroga il software sui contatti che sussistono tra le parti presenti nell'assieme, pensate nella loro posizione di equilibrio statico. In Figura 3.12 è illustrato il modello virtuale ricavato con la relativa nomenclatura delle varie parti, mentre in Figura 3.13 viene mostrato il risultato dell'interrogazione del software CAD sul modello per estrapolare le informazioni di contatto.

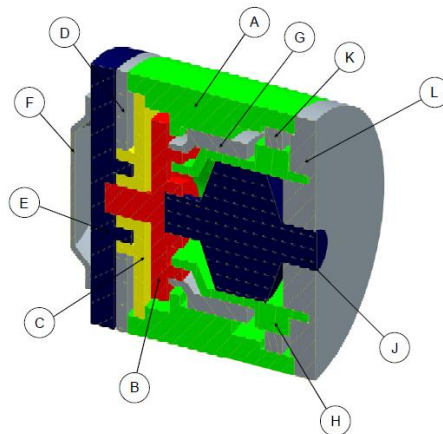
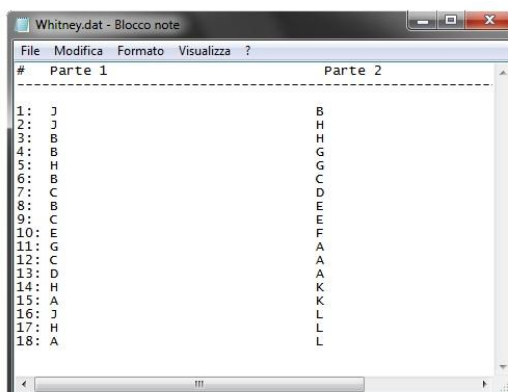


Figura 3.12 Modello virtuale dell'esempio AFI di De Fazio e Whitney.



#	Parte 1	Parte 2
1:	J	B
2:	J	H
3:	B	H
4:	B	G
5:	H	G
6:	B	C
7:	C	D
8:	B	E
9:	C	E
10:	E	F
11:	G	A
12:	C	A
13:	D	A
14:	H	K
15:	A	K
16:	J	L
17:	H	L
18:	A	L

Figura 3.13 Lista dei collegamenti dell'esempio AFI di De Fazio e Whitney.

### Generazione delle parti e del grafo dei collegamenti

Dalla lista dei contatti precedentemente ottenuta, viene ricavata la lista delle parti, ossia dei nodi, vengono effettuati i primi calcoli di classificazione su di essi e viene costruito e mostrato il relativo grafo non orientato dei collegamenti rappresentativo. Lo stesso grafo viene mostrato in Figura 3.14.

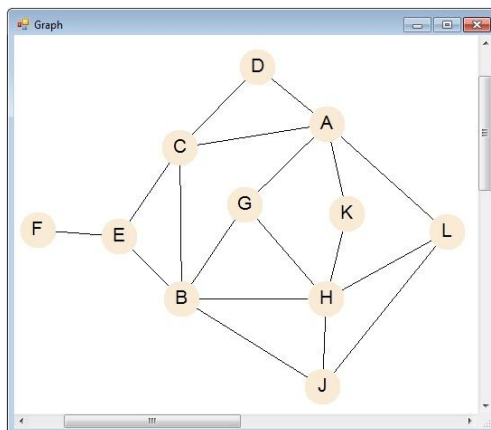


Figura 3.14 Grafo non orientato dei collegamenti dell'esempio AFI di De Fazio e Whitney.

### Riduzione iterativa dei nodi di grado uno

Una volta individuata la lista completa delle parti e i relativi gradi, si procede con la riduzione iterativa dei nodi di grado uno; per questo caso, solo il nodo *F* viene ridotto con l'adiacente nodo *E*, generando il primo

sottoassieme  $SAN11(1) = (E - F)$  composto da 2 elementi e caratterizzato da una sola possibile sottosequenza di assemblaggio, corrispondente alle:

$$2^{(k-2)} = 2^{(2-2)} = 1 \text{ possibile sottosequenza}$$

La Figura 3.15 mostra il sottoassieme generato dall'unione delle due parti, mentre la Figura 3.16 mostra il grafo ottenuto dalla riduzione appena effettuata.

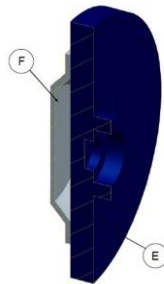


Figura 3.15 Sottoassieme  $SAN11(1) = F-E$ .

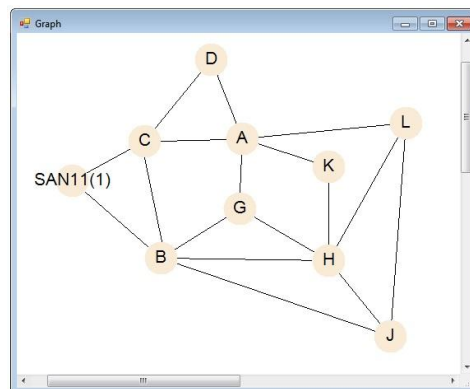


Figura 3.16 Grafo dei collegamenti esempio AFI dopo il primo passo di riduzione.

### Identificazione dei cicli indotti e classificazione dei cicli e dei nodi

Una volta ridotti tutti i nodi di grado unitario, l'algoritmo procede con la ricerca di tutti i cicli indotti presenti nel grafo dei collegamenti rappresentante il secondo livello di riduzione; 13 differenti cicli indotti vengono individuati, come è possibile notare dalla Figura 3.17 rappresentante la finestra di visualizzazione dei risultati riguardanti i cicli.



Allo stesso modo, per tutti i nodi vengono calcolati gli indici caratteristici al fine di individuare il miglior indice di centralità di un nodo, come mostrato in Figura 3.18. I relativi pesi degli indici vengono lasciati ad un valore standard, che su basi empiriche, corrisponde ad una casistica generica. In Tabella 3.5 vengono riportati relativi valori.

	Number of Cycle	Cycle name	Cycle Degree	Index of Cycle Degree	Sum of Node Centrality	Mean Node Centrality	Index Of Node Centrality	Cycle Connectivity	Index Of Cycle Connectivity	Index Of Cycle Importance
▶	1	J - H - B	5	0,07692308	0,444947	0,148315668	0,07563644	0,714285731	0,06535948	0,07263967
	2	J - L - A - G - B	5	0,07692308	0,7076438	0,141528755	0,072175324	1	0,09150327	0,08020056
	3	J - L - A - C - B	5	0,07692308	0,7621645	0,152432889	0,0777360946	1	0,09150327	0,0820541456
	4	J - L - H	4	0,06153846	0,3548846	0,118294865	0,06032675	0,5714286	0,052287586	0,0580509342
	5	B - G - H	5	0,07692308	0,495298922	0,165099636	0,08419575	0,714285731	0,06535948	0,07549277
	6	B - C - A - K - H	5	0,07692308	0,8753775	0,1750755	0,08928313	1	0,09150327	0,08590316
	7	B - C - A - L - H	5	0,07692308	0,932868659	0,186573729	0,09514687	1	0,09150327	0,08785774
	8	B - C - A - G	6	0,0923076943	0,6990052	0,1747513	0,0891177952	1	0,09150327	0,09097625
	9	B - SAN11(1) - C	5	0,07692308	0,288825929	0,09627531	0,0490974523	0,714285731	0,06535948	0,06379334
	10	H - K - A - G	5	0,07692308	0,617283165	0,154320791	0,0786988661	0,8333333	0,07625273	0,0772915557
	11	H - L - A - G	5	0,07692308	0,674774349	0,168693587	0,08602855	0,8333333	0,07625273	0,07973479
	12	H - L - A - K	5	0,07692308	0,6451079	0,161276981	0,08224631	0,8333333	0,07625273	0,07847404
	13	C - A - D	5	0,07692308	0,354790032	0,118263341	0,060310673	0,714285731	0,06535948	0,06753108
*										

Figura 3.17 Classificazione dei cicli individuati per l'esempio AFI.

	Position	Part name	Degree	Index Node Degree	Pertaining Cycle	Pertaining Cycles Ratio	Index Of Pertaining Cycles	Sum of Degree Adjacent Nodes	Mean Degree Adjacent Nodes	Index Mean Degree Adjacent Nodes	Clustering Coefficient CCF	Weighted Clustering Coefficient WCCF	Index Of Weighted Clustering Coefficient	Index Of Centrality
▶	1	J	3	0,0882353	4	0,307692319	0,07843137	13	4,33333...	0,1068200...	0,6666667	2	0,1558441...	0,03533456
	2	B	5	0,14705883	8	0,615384638	0,156862736	17	3,4	0,08381266	0,3	1,5	0,1168831...	0,203573674
	3	H	5	0,14705883	8	0,615384638	0,156862736	16	3,2	0,0788825	0,3	1,5	0,1168831...	0,206038758
	4	G	3	0,0882353	5	0,3846154	0,09803921	15	5	0,1232539...	0,333333343	1	0,0779220...	0,08568652
	5	C	4	0,117647059	6	0,461538464	0,117647044	14	3,5	0,08627773	0,333333343	1,33333337	0,1038961...	0,140207186
	6	D	2	0,05882353	1	0,07692308	0,019607842	9	4,5	0,1109285...	1	2	0,1558441...	-0,05495496
	7	SAN11(1)	2	0,05882353	1	0,07692308	0,019607842	9	4,5	0,1109285...	1	2	0,1558441...	-0,05495496
	8	A	5	0,14705883	9	0,6923077	0,176470578	14	2,8	0,0690221...	0,1	0,5	0,0389610...	0,2695378
	9	K	2	0,05882353	3	0,230769232	0,05882352...	10	5	0,1232539...	0	0	0	0,0560201
	10	L	3	0,0882353	6	0,461538464	0,117647044	13	4,33333...	0,1068200...	0,333333343	1	0,0779220...	0,113511279
*														

Figura 3.18 Classificazione dei nodi individuati per l'esempio AFI.

Tabella 3.5 Peso degli i degli indici utilizzato per l'esempio AFI.

	$W_{ND}$	$W_{PC}$	$W_{DAN}$	$W_{CCF}$	$W_{CD}$	$W_{MNC}$	$W_{CC}$
<b>Peso Indice</b>	1	1	0.5	0.5	1	1	1

La classificazione dei cicli e dei nodi viene effettuata per poter individuare quale sia il miglior sottoassieme da ridurre e su quale nodo base conviene generare le sottosequenze di assemblaggio dello stesso. Dalla tabella di Figura 3.18 si evince che il nodo *A* è classificato con l'indice di centralità maggiore, così la scelta del migliore sottoassieme ricade su tutti quei cicli che hanno in comune lo stesso nodo. Il ciclo, con in comune il nodo *A*, con il maggiore indice di importanza (dalla tabella di Figura 3.17) risulta essere quello caratterizzato dal gruppo *(A-C-B-G)* e prende il nome *SAC2I(1)*, come mostrato in Figura 3.19. Le sottosequenze che si genereranno in seguito alla riduzione di un ciclo composto da  $k = 4$  elementi, di cui uno solo viene considerato come nodo piattaforma (ossia il nodo *A*), saranno in numero pari a:

$$1 \cdot 2^{(k-2)} = 1 \cdot 2^{(4-2)} = 4 \text{ possibili sottosequenze}$$

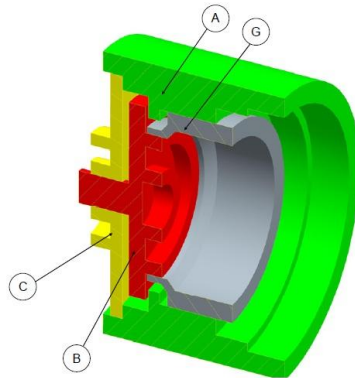


Figura 3.19 Sottoassieme *SAC2I(1) = (A - C - G - B)*.

Inoltre, se a questo livello di riduzione sono presenti dei cicli che non hanno componenti pertinenti al ciclo appena ridotto, gli stessi verranno analizzati e ridotti ad un ulteriore sottoassieme. In questo caso particolare, il ciclo caratterizzato dai nodi *(J-H-L)* rispetta i requisiti di riduzione e verrà perciò considerato come un nuovo sottoassieme che prende il nome *SA(JLH)* dai suoi elementi. Il nodo che verrà considerato come piattaforma, ancora una volta, sarà quello che presenterà il maggiore indice di centralità, ossia in questo determinato caso sarà il nodo *H*, portando così alla generazione di:

$$2^{(k-2)} = 2^{(3-2)} = 2 \text{ possibili sottosequenze}$$

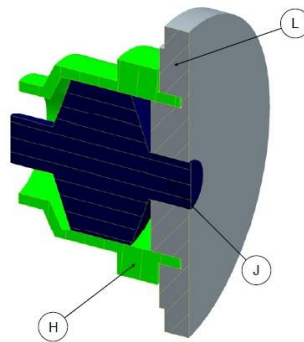


Figura 3.20 Sottoassieme  $SA(JLH) = (J-H-L)$ .

### Riduzione dei sottoassiemi

A questo punto, i due sottoassiemi  $SAC21(1)$  e  $SA(JLH)$  vengono ridotti ad un unico nodo che corrisponde al nodo piattaforma su cui è stata fatta la riduzione, evitando la formazione di lati doppi o cicli chiusi su se stessi (loop) come mostrato in Figura 3.21.

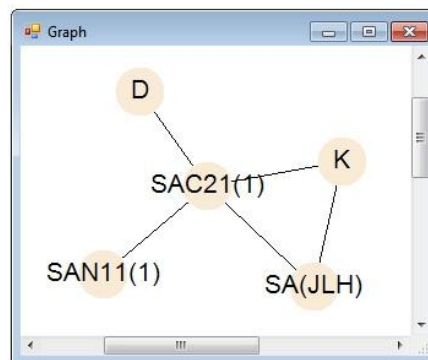


Figura 3.21 Grafo ridotto al 3° livello di riduzione.

### Riduzione iterativa dei nodi di grado 1

La procedura continua e si passa alla successiva iterazione dell'assieme ridotto che comincia con l'individuazione dei nodi di grado unitario. In questo caso vengono individuati due nodi di grado 1, ed in particolare il nodo  $D$  ed il nodo  $SANI(1)$ .

Dal momento che entrambi i nodi hanno lo stesso nodo adiacente in comune e quindi hanno lo stesso livello di importanza, il software genera due sottolivelli che sviluppa separatamente, come rappresentato in Figura 3.22.

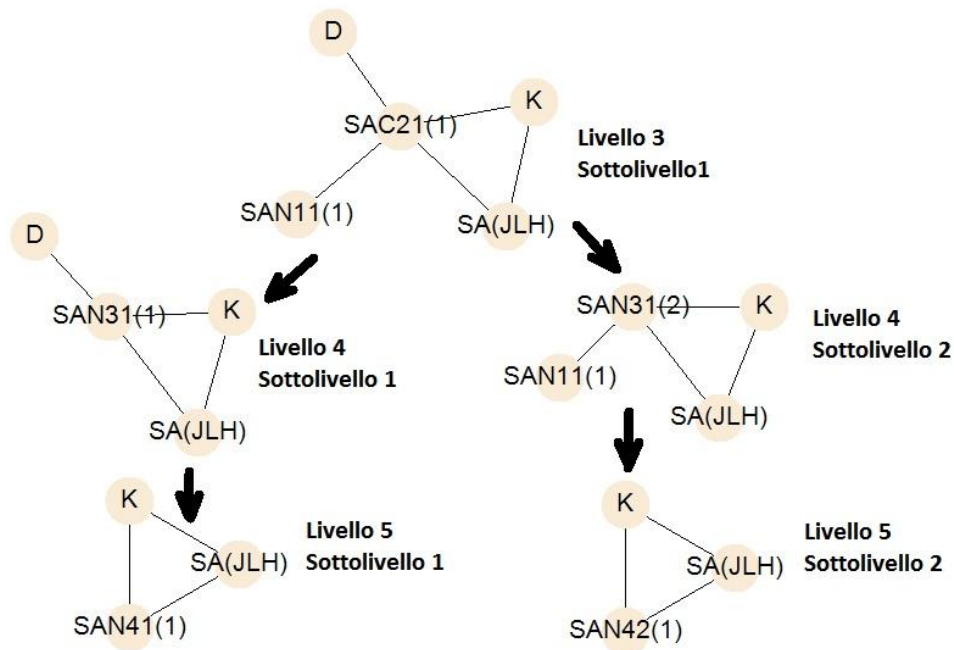


Figura 3.22 Gerarchia delle riduzioni dopo il livello 3 dell'esempio AFI.

Così facendo tutte le strade di riduzioni saranno valide e verranno portate avanti separatamente generando il doppio delle soluzioni dal livello 3 in avanti. Questo equivale a ridurre i due nodi di grado uno in un solo passaggio e memorizzare le 2 sottosequenze da quest'ultimo generato. In questo caso due sottolivelli vengono creati portando avanti separatamente le due sottosequenze, generando le medesime soluzioni.

### Generazione finale delle sequenze

Dopo i vari passaggi di riduzione, il sistema arriva al suo ultimo livello di riduzione, che in questo caso è costituito da due sottolivelli a causa della precedente dissociazione delle soluzioni  $SAN41(1)$  e  $SAN42(1)$ . Gli ultimi sottoassiemi ricavati sono composti da 3 elementi che costituiscono un ciclo chiuso, per questo motivo si avrà la generazione finale di:

$$2 \cdot 3 \cdot 2^{(k-2)} = 2 \cdot 3 \cdot 2^{(3-2)} = 12 \text{ Possibili sequenze di assemblaggio}$$

La Figura 3.23 sottostante rappresenta il modello virtuale scomposto nei tre sottoassiemi dati dall'ultimo livello di riduzione.

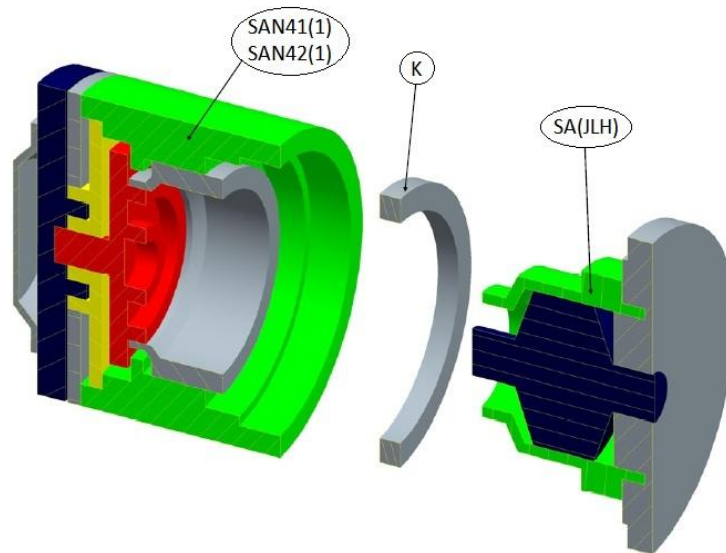


Figura 3.23 Sottoassieme finale (SAN41(1)/SAN42(1) – K –SA(JLH))

La seguente tabella riassume tutti i sottoassiemi generati ad ogni livello durante l'applicazione dell'algoritmo e le possibili sequenze finali (*SF*) che lo stesso genera al suo ultimo livello. Il numero totale delle sequenze possibili ricavate corrisponde, come si può facilmente intuire, alla moltiplicazione tra il numero delle sequenze finali generate ed il numero di sottosequenze generate per ogni sottoassieme ad ogni passaggio di riduzione.

In questo caso particolare si otterranno esattamente:

$$[(6 + 6) \text{sequenze finali} \cdot 2 \cdot 4] = 96 \text{ possibili sequenze di assemblaggio}$$

Queste non sono altro che le sequenze generate secondo il metodo qui proposto, che si basa esclusivamente sui collegamenti e sul numero di parti presenti nell'assieme.

Tabella 3.6 Rappresentazione degli assemblaggi per l'esempio AFI.

Livello	Componenti del sottoassieme	Nome sottoassieme	Sequenze di assemblaggio	
			Q.tà	Sequenza
2,1	<i>E,F</i>	<i>SAN11(1)</i>	1	<i>E&lt;F</i>
3,1	<i>A,C,B,G</i>	<i>SAC21(1)</i>	4	<i>A&lt;C&lt;B&lt;G</i>
				<i>A&lt;G&lt;B&lt;C</i>
				<i>A&lt;C&lt;G&lt;B</i>
				<i>A&lt;G&lt;C&lt;B</i>
	<i>H,L,J</i>	<i>SA(JLH)</i>	2	<i>H&lt;L&lt;J</i>
				<i>H&lt;J&lt;L</i>
4,1	<i>SAC21(1),SAN11(1)</i>	<i>SAN31(1)</i>	1	<i>SAC21(1)&lt;SAN11(1)</i>
4,2	<i>SAC21(1),D</i>	<i>SAN31(2)</i>	1	<i>SAC21(1)&lt;D</i>
5,1	<i>SAN31(1),D</i>	<i>SAN41(1)</i>	1	<i>SAN31(1)&lt;D</i>
5,2	<i>SAN31(2),SAN11(1)</i>	<i>SAN42(1)</i>	1	<i>SAN31(2)&lt;SAN11(1)</i>
6,1	<i>SAN41(1),K,SA(JLH)</i>	<i>SF</i>	6	<i>SAN41(1)&lt;K&lt;SA(JLH)</i>
				<i>SAN41(1)&lt;SA(JLH)&lt;K</i>
				<i>SA(JLH)&lt;K&lt;SAN41(1)</i>
				<i>SA(JLH)&lt;K&lt;SAN41(1)</i>
				<i>K&lt;SAN41(1)&lt;SA(JLH)</i>
				<i>K&lt;SA(JLH)&lt;SAN41(1)</i>
6,2	<i>SAN42(1),K,SA(JLH)</i>	<i>SF</i>	6	<i>SAN42(1)&lt;K&lt;SA(JLH)</i>
				<i>SAN42(1)&lt;SA(JLH)&lt;K</i>
				<i>SA(JLH)&lt;K&lt;SAN42(1)</i>
				<i>SA(JLH)&lt;SAN42(1)&lt;K</i>
				<i>K&lt;SAN42(1)&lt;SA(JLH)</i>
				<i>K&lt;SA(JLH)&lt;SAN42(1)</i>

Nessuna informazione sulla geometria delle parti o sulle precedenze di assemblaggio delle stesse è stata sfruttata durante la generazione delle attuali sequenze di assemblaggio. Il che significa che le sequenze generate, nonostante rispecchino fedelmente la teoria applicata, non è detto che da un punto di vista pratico siano fattibili. Questo aspetto fondamentale del problema richiede un'ulteriore valutazione dei risultati ottenuti attraverso un esame della fattibilità dei sottoassiemi e delle sequenze generate. Dal momento che, come già detto, non viene preso in considerazione nessun aspetto al di fuori dei soli collegamenti presenti, il metodo non tiene conto di possibili congruenze tra le geometrie delle parti e aree di operabilità per la messa insieme delle parti. Tramite i dati adoperati per questo tipo di analisi quindi, non è possibile valutare in maniera concreta la fattibilità pratica di una sequenza di assemblaggio. Ad ogni modo, come annunciato precedentemente, il numero di possibilità può venire ulteriormente ridotto valutando la fattibilità delle sottosequenze per ogni sottoassieme generato e valutando la stessa fattibilità delle sequenze ottenute tramite composizione dei relativi sottoassiemi.

### 3.3 Validazione attraverso un caso pratico

#### *Valvola a sfera*

La metodologia automatica implementata è stata successivamente valutata su un tipico assieme di natura meccanica, composto da un discreto numero di parti. L'esempio in questione, è quello di una valvola a sfera, costituita da 8 parti collegate reciprocamente in modo da comporre l'assieme completo; in Figura 3.24 viene rappresentato il modello virtuale modellato tramite software CAD. Il modello virtuale è stato semplificato di tutti quegli organi di collegamento che si presentano più volte e non forniscono delle caratteristiche essenziali nella descrizione dell'assieme e del suo funzionamento. Così, le 8 viti e gli 8 bulloni che si presentano sulle due flange esterne verranno omissi dall'analisi, mentre la vite centrale, il relativo dado e la maniglia verranno considerati, dal momento che una loro omissione renderebbe la descrizione dell'assieme incompleta.



Figura 3.24 Modello virtuale di una valvola a sfera.

A differenza del caso precedente, non viene indicata tutta la procedura passo per passo, ma vengono solamente riportati i risultati ottenuti attraverso l'applicazione del software sviluppato ed alcune considerazioni qualitative sui sottoassiemi creati. Di seguito, in Figura 3.25, viene mostrato il particolare del modello virtuale in sezione trasversale accompagnato dalla sottostante Tabella 3.7 che riporta l'elenco nominativo di ogni parte costituente l'assieme completo valvola a sfera.



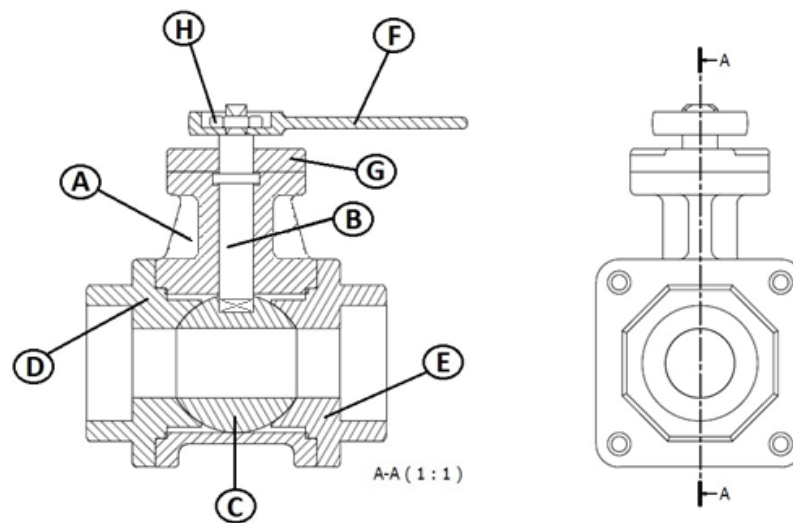


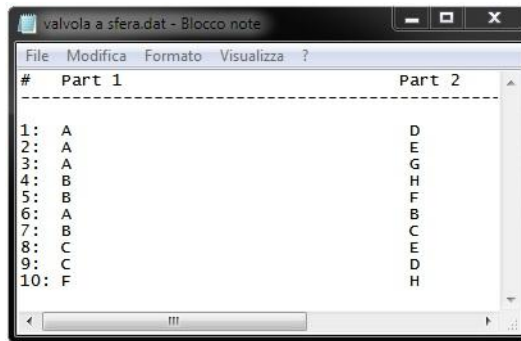
Figura 3.25 Particolare in sezione della valvola a sfera.

Tabella 3.7 Elenco nominativo delle parti costituenti valvola a sfera.

N °	Nominativo	Nome Parte
1	A	Corpo
2	B	Albero
3	C	Sfera
4	D	Flangia sinistra
5	E	Flangia destra
6	F	Chiave
7	G	Fermo
8	H	Dado

Come in tutti gli altri casi, una volta noto il modello virtuale, si procede con l'interrogazione del software di modellazione usato al fine di generare una lista dei collegamenti presenti tra le parti, dovuti ad un contatto o ad una interferenza nulla. In Figura 3.26 viene riportata la relativa lista dei collegamenti generata. La prima attività svolta dal metodo automatico implementato è la generazione del relativo grafo non orientato dei collegamenti presenti, la Figura 3.27 mostra la finestra che appare all'utente

relativa alla visualizzazione dello stesso grafo. Questa risulterà molto utile in quanto permetterà allo stesso di valutare la bontà dei dati di partenza ricavati in maniera veloce ed intuitiva.



#	Part 1	Part 2
1:	A	D
2:	A	E
3:	A	G
4:	B	H
5:	B	F
6:	A	B
7:	B	C
8:	C	E
9:	C	D
10:	F	H

Figura 3.26 Lista dei collegamenti relativi all'esempio valvola a sfera.

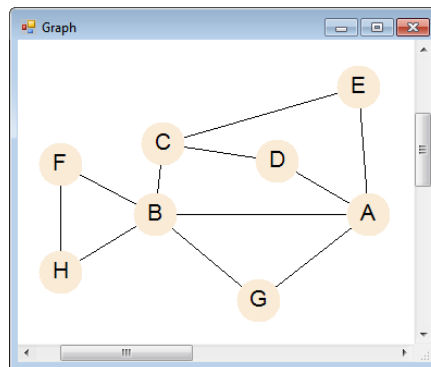


Figura 3.27 Grafo relativo della valvola a sfera.

Una volta fatto ciò, il metodo procede con la costruzione della lista delle parti presenti, con l'individuazione dei cicli indotti sul grafo e con la classificazione dei cicli e dei nodi attraverso il calcolo degli indici. Per il primo livello, ossia per l'assieme di partenza, si ottiene un sistema composto da 8 nodi e 5 cicli indotti; non tutti i cicli indotti sono direttamente visibili sul grafo dal momento che alcuni di essi sono impliciti alla disposizione di Figura 3.27. Quindi, si procede con la riduzione iterativa dei cicli e dei nodi di grado uno fino a che tutto il sistema non venga ridotto ad un unico elemento.

La Figura 3.28 e la Figura 3.29 mostrano la visualizzazione delle finestre che l'utente ottiene sullo schermo una volta che chiede di visualizzare i risultati

ottenuti al primo livello di riduzione. Per quanto riguarda il primo passo di riduzione, non essendoci nodi di grado unitario, il sistema cerca il nodo da considerare come piattaforma, che in questo caso è il nodo A, e successivamente classifica i cicli pertinenti a quel nodo con lo scopo di individuare quale o quali sottoassiemi corrispondenti creare. In questa particolare condizione si hanno due cicli, con lo stesso indice di importanza, attinenti al nodo considerato come piattaforma; in particolare il ciclo *A,B,C,D* ed il ciclo *A,B,C,E*. In Figura 3.30 vengono mostrati i corrispondenti sottoassiemi considerati.

Position	Part name	Degree	Index Node Degree	Pertaining Cycle	Pertaining Cycles Ratio	Index Of Pertaining Cycles	Sum of Degree Adjacent Nodes	Mean Degree Adjacent Nodes	Index Mean Degree Adjacent Nodes	Clustering Coefficient CCF	Weighted Clustering Coefficient WCCF	Index Of Weighted Clustering Coefficient	Index Of Centrality
1	A	4	0,181818...	4	0,8	0,222222...	11	2,75	0,10242...	0,16666...	0,6666667	0,08695...	0,33586...
2	D	2	0,090909...	2	0,4	0,111111...	7	3,5	0,13035...	0	0	0	0,1368433
3	E	2	0,090909...	2	0,4	0,111111...	7	3,5	0,13035...	0	0	0	0,1368433
4	G	2	0,090909...	1	0,2	0,05555...	9	4,5	0,16759...	1	2	0,26086...	-0,06776...
5	B	5	0,227272...	4	0,8	0,222222...	13	2,6	0,09683...	0,2	1	0,13043...	0,3093517
6	H	2	0,090909...	1	0,2	0,05555...	7	3,5	0,13035...	1	2	0,26086...	-0,04914...
7	F	2	0,090909...	1	0,2	0,05555...	7	3,5	0,13035...	1	2	0,26086...	-0,04914...
8	C	3	0,136363...	3	0,6	0,16666...	9	3	0,11173...	0	0	0	0,24716...

Figura 3.28 Classificazione dei nodi per l'esempio valvola a sfera.

Number of Cycle	Cycle name	Cycle Degree	Index of Cycle Degree	Sum of Node Centrality	Mean Node Centrality	Index Of Node Centrality	Cycle Connectivity	Index Of Cycle Connectivity	Index Of Cycle Importance
1	A - B - C - D	4	0,222222224	1,02921987	0,257304966	0,258902282	1	0,243902445	0,241675645
2	A - E - C - D	2	0,111111112	0,8302027	0,207550675	0,208839133	0,5	0,121951222	0,147300482
3	A - B - C - E	4	0,222222224	1,02921987	0,257304966	0,258902282	1	0,243902445	0,241675645
4	A - B - G	5	0,27777778	0,5774431	0,192481041	0,19367595	1	0,243902445	0,238452062
5	B - F - H	3	0,166666672	0,237566352	0,07918879	0,07968038	0,6	0,146341473	0,130896181

Figura 3.29 Classificazione dei cicli per l'esempio valvola a sfera.

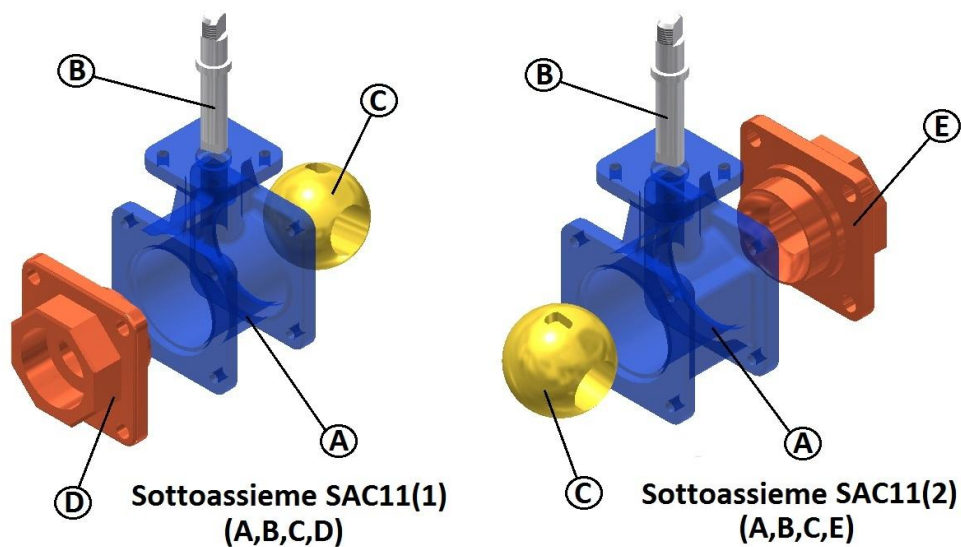


Figura 3.30 Sottoassiemi individuati al primo livello di riduzione per l'esempio valvola a sfera.

Come si può facilmente intuire, i sottoassiemi che vengono generati al primo livello di riduzione sono complementari tra loro; questo è dovuto alla particolare disposizione simmetrica tra le parti che compongono l'assieme. Infatti, coerentemente con la teoria dei grafi, anche il grafo dei contatti non orientato corrispondente presenta una simmetria dei nodi in questione, e quindi presenta due cicli indotti equipollenti che coinvolgono alternativamente la flangia di destra e la flangia di sinistra a parità delle altre parti. Secondo quanto detto nel relativo capitolo trattante la tecnica di generazione delle sequenze, essendoci due cicli con lo stesso valore, entrambi le strade di riduzione verranno considerate come valide e verranno portate avanti separatamente. Quindi, il secondo livello di riduzione presenterà due sottolivelli a se stanti altrettanto validi. Così facendo, i due sottoassiemi individuati vengono creati e le relative sottosequenze di assemblaggio degli stessi vengono generate. In particolare, per un ciclo composto da 4 elementi basato su un solo nodo piattaforma, si hanno:

$$1 \cdot 2^{(k-2)} = 1 \cdot 2^{(4-2)} = 4 \text{ Possibili sottosequenze di assemblaggio}$$

Dal momento che in questo caso i cicli individuati sono 2 ed entrambi composti da 4 nodi, al primo passaggio di riduzione si avrà la generazione di 8 sottosequenze di montaggio.

Il procedimento di individuazione viene portato avanti riducendo i cicli individuati ad un unico nodo e ricalcolando tutti gli indici per tutti i nodi e tutti i cicli ottenuti dal nuovo grafo non orientato.

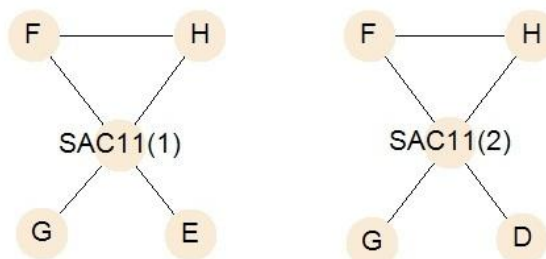


Figura 3.31 Grafo equivalente ricavato al secondo livello di riduzione (primo sottolivello a sinistra e secondo sottolivello a destra).

In Figura 3.31 vengono riportati i grafi non orientati dei collegamenti ottenuti nel secondo livello di riduzione. Dal momento che si ha, in entrambi i casi, la presenza di più nodi di grado unitario, la routine procede nella riduzione iterativa dei singoli nodi per entrambi i sottolivelli riportati. La Figura 3.32 mostra l'esploso dell'assieme con le parti ed i sottoassiemi che verranno presi in considerazione nel prossimo livello di riduzione.

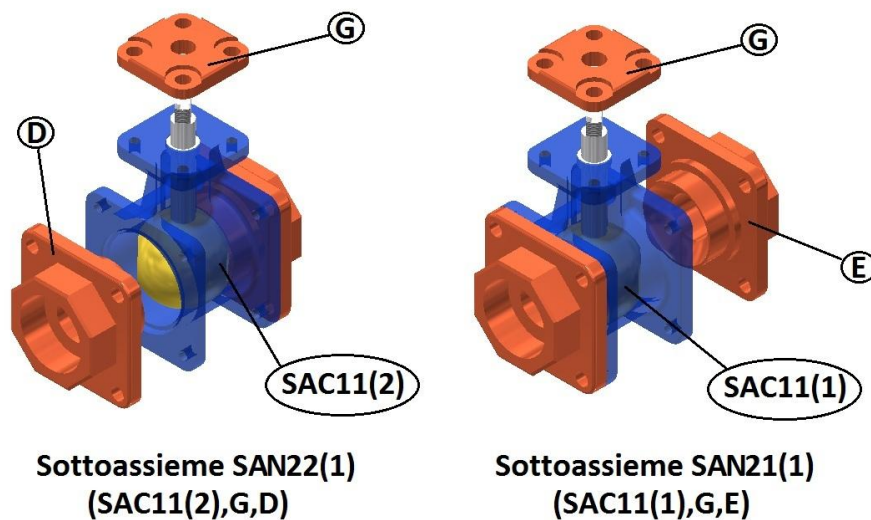


Figura 3.32 Sottoassiemi individuati al secondo livello di riduzione per l'esempio valvola a sfera.

In particolare si ha che il nodo  $G$ , ossia quello rappresentante il fermo, viene considerato in entrambi i sottolivelli, mentre alternativamente al caso in questione, viene coinvolta la flangia che non è ancora stata assemblata, ossia il nodo  $D$  o il nodo  $E$ . Essendo i due casi del tutto equivalenti, si avrà che per due nodi di grado unitario collegati direttamente al medesimo nodo base, il numero di sottosequenze generate sarà pari a 2 per ogni sottoassieme creato. I nodi di grado uno verranno così fusi insieme al nodo adiacente ed il nuovo grafo equivalente verrà creato. Il metodo iterativo procede nella successiva individuazione dei nodi di grado unitario e dei cicli indotti passando al successivo livello di riduzione creato. Si noti che per entrambe le strade di riduzione si arriva ad un ultimo equivalente dove, gli elementi che compongono il sottoassieme  $SAN21(1)$  o il sottoassieme  $SAN22(1)$  sono i medesimi come anche le parti mancanti da assemblare costituite dai nodi  $F$  e  $H$ . La differenza sostanziale che discrimina i due sottolivelli di riduzione è la diversa composizione con cui vengono create le sottosequenze per ogni sottoassieme che volta per volta viene generato. A tal proposito, la Figura 3.33 mostra un esploso dei sottoassiemi e delle parti prese in considerazione all'ultimo livello di riduzione.

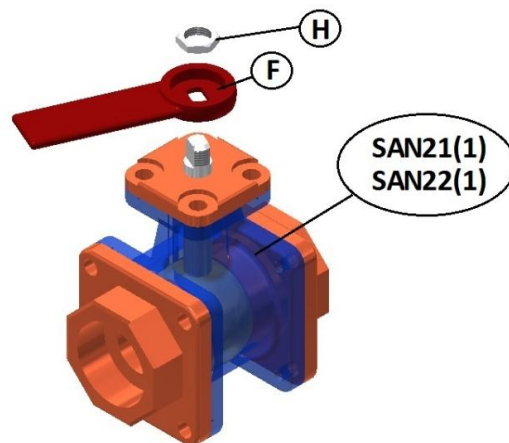


Figura 3.33 Sottoassiemi individuati negli ultimi livelli di riduzione per l'esempio valvola a sfera.

Essendo entrambi gli ultimi livelli caratterizzati da un unico ciclo indotto composto da 3 nodi, si avrà la generazione per ognuno di essi di:

$$3 \cdot 2^{(k-2)} = 3 \cdot 2^{(3-2)} = 6 \text{ Possibili sequenze finali di assemblaggio}$$

Dal momento che sono presenti due sottolivelli di riduzione, all'ultimo passaggio si avrà la generazione di 12 possibili sequenze finali di assemblaggio.

Tabella 3.8 Rappresentazione assemblaggi per l'esempio Valvola a sfera.

Livello, sottolivello	Componenti del sottoassieme	Nome sottoassieme	Sequenze di assemblaggio	
			Q.tà	Sequenza
1,1	A,B,C,D	SAC11(1)	4	$A < B < C < D$
				$A < B < D < C$
				$A < D < B < C$
				$A < D < C < B$
1,2	A,B,C,E	SAC11(2)	4	$A < B < C < E$
				$A < B < E < C$
				$A < E < C < B$
				$A < E < B < C$
2,1	SAC11(1),G,E	SAN21(1)	2	$SAC11(1) < G < E$
				$SAC11(1) < E < G$
2,2	SAC11(2),G,D	SAN22(1)	2	$SAC11(2) < G < D$
				$SAC11(2) < D < G$
3,1	SAN21(1),F,H	SF	6	$SAN21(1) < F < H$
				$SAN21(1) < H < F$
				$H < F < SAN21(1)$
				$H < SAN21(1) < F$
				$F < H < SAN21(1)$
				$F < SAN21(1) < H$
3,2	SAN22(1),F,H	SF	6	$SAN22(1) < F < H$
				$SAN22(1) < H < F$
				$H < F < SAN22(1)$
				$H < SAN22(1) < F$
				$F < H < SAN22(1)$
				$F < SAN22(1) < H$

Anche in questo caso, il numero totale delle sequenze possibili ricavate corrisponde, alla moltiplicazione tra il numero delle sequenze finali (*SF*) generate ed il numero di sottosequenze generate per ogni sottoassieme ad ogni passaggio di riduzione.

Per concludere, l'applicazione della metodologia automatica proposta sull'assieme valvola a sfera, fornisce un totale di:

$$[(6 + 6) \textit{sequenze finali} \cdot 2 \cdot 4] = 96 \textit{ possibili sequenze di assemblaggio}$$

Come nel caso precedente, il numero di possibilità può venire ulteriormente ridotto, valutando la fattibilità delle sottosequenze per ogni sottoassieme generato e valutando la stessa fattibilità delle sequenze ottenute tramite composizione dei relativi sottoassiami. Il capitolo successivo vuole argomentare in maniera più approfondita quanto appena detto.



## CAPITOLO 4

# SELEZIONE DELLE SEQUENZE GENERATE

### 4.1 Considerazioni sulle sequenze generate dal metodo proposto

La metodologia considerata, che in questa sede è stata implementata numericamente ed automaticamente al calcolatore, fornisce un utile strumento che genera una molteplicità di soluzioni di montaggio. Dal momento che, come già precedentemente affermato, il metodo si basa unicamente sulle informazioni di contatto preesistenti tra le parti in questione, non è detto che vengano create soltanto soluzioni di assemblaggio valide. Si consideri a tal proposito l'esempio bidimensionale proposto in Figura 4.1; si può facilmente intuire che il grafo non orientato dei collegamenti equivalenti sia composto da un unico ciclo chiuso composto da 4 elementi collegati reciprocamente l'uno con l'altro (come mostrato in Figura 4.2). Come è già stato precedentemente illustrato a pagina 75, il metodo fornisce per questo determinato caso, in accordo con la teoria di riduzione usata, un totale di 16 possibili sequenze di assemblaggio fattibili e non fattibili. In particolare, le sequenze che vengono generate attraverso l'applicazione del metodo, sono elencate nella Tabella 4.1. Sorge spontaneo notare che, in questa determinata situazione, esiste una particolare condizione di montaggio per cui la parte  $D$  non può essere montata prima che le restanti parti vengano assemblate tra loro; questo è dovuto alla particolare geometria delle parti che necessitano di venir montate reciprocamente secondo un determinato ordine cronologico. Se quest'ordine non venisse rispettato, si otterrebbe come risultato l'impossibilità di assemblare l'assieme completo.

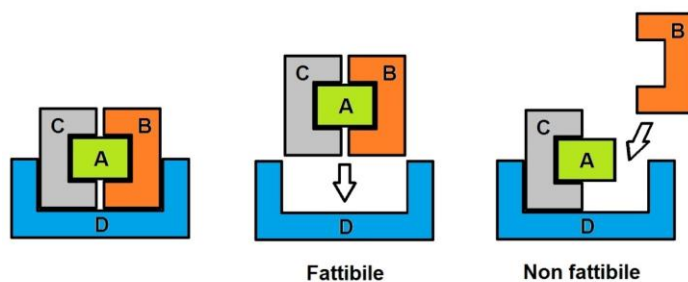


Figura 4.1 Esempio bidimensionale caratteristico e possibile ordine di assemblaggio.

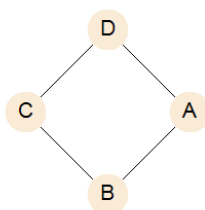


Figura 4.2 Grafo dei collegamenti equivalenti.

Tabella 4.1 Sequenze generate.

<b>N °</b>	<b>Sequenza</b>	<b>Fattibilità</b>
1	<b>A&lt;B&lt;C&lt;D</b>	<b>SI</b>
2	A<B<D<C	NO
3	A<D<B<C	NO
4	A<D<C<B	NO
5	<b>B&lt;A&lt;C&lt;D</b>	<b>SI</b>
6	B<A<D<C	NO
7	<b>B&lt;C&lt;A&lt;D</b>	<b>SI</b>
8	B<C<D<A	NO
9	<b>C&lt;B&lt;A&lt;D</b>	<b>SI</b>
10	C<B<D<A	NO
11	C<D<A<B	NO
12	C<D<B<A	NO
13	D<A<B<C	NO
14	D<A<C<B	NO
15	D<C<A<B	NO
16	D<C<B<A	NO

Logicamente, tramite l'applicazione dell'algoritmo, questa condizione, che richiama in se delle caratteristiche geometriche non coinvolte dal metodo, non viene considerata. Così, la serie di sequenze che viene generata, conterrà al suo interno anche quelle sequenze che ai fini pratici non saranno attuabili. In particolare, su un totale di 16 possibili sequenze generate, come mostrato in Tabella 4.1, risulterà che solo 4 di esse risultino fattibili, ossia tutte quelle sequenze che assemblino per ultima la parte chiamata *D*.

Nasce dunque l'esigenza di trovare un metodo selettivo che escluda categoricamente tutte quelle sequenze che, nonostante rispettino le ipotesi di riduzione prefissate dal metodo, non siano realmente fattibili ai fini pratici una volta valutate attraverso le reali caratteristiche di forma delle parti. Inoltre, si richiede che il metodo di selezione che verrà applicato riesca ad arrivare ad iterazione, ossia ad aver escluso completamente tutte le sequenze non fattibili, con il minor numero di informazioni di precedenza possibili. Si ricorda che la natura del metodo qui implementato, è del tutto differente dai metodi proposti da De Fazio e Whitney (De Fazio, et al. 1985) e da Bourjault (Bourjault, et al., 1984), dove la generazione viene attuata, volta per volta, attraverso l'interrogazione dell'utente sulla validità delle precedenze di collegamento imposte. Infatti, fino a questo punto, la generazione delle sequenze, ottenute con il metodo qui proposto, è stata svolta in maniera del tutto autonoma e senza che venga coinvolto l'intervento umano.

## **4.2 Tecniche di selezione delle sequenze fattibili**

Al fine di ampliare la bontà del metodo applicato, risulta necessario individuare un metodo ricorsivo che riesca a selezionare, nella totalità delle soluzioni precedentemente ricavate, solamente le sequenze di assemblaggio fattibili; questa selezione deve essere svolta basandosi esclusivamente sulle informazioni ottenute attraverso le domande poste all'utente. Dal momento che la metodologia di generazione delle sequenze di assemblaggio è stata implementata attraverso un software che svolge automaticamente tutti i passaggi preposti dallo stesso metodo, si vuole a questo punto sviluppare una tecnica selettiva sistematica che possa venir anch'essa implementata nello stesso software in un secondo momento. In questa maniera, sarà possibile ottenere uno strumento completo che fornisca un ausilio al progettista nella scelta delle sequenze, in modo tale che essi debbano solamente rispondere a delle domande di fattibilità che non richiedano esperienze in merito alla tecnica usata.

Il primo passo concettuale consiste nella scelta del tipo di domande da porre all'utente. Come ormai ben noto, il metodo basa la generazione delle sequenze selezionando volta per volta quali sottoassiemi risulta conveniente ridurre. Questo procedimento prosegue fino a quando risulti presente un unico sottoassieme composto dalla combinazione lineare di tutti i sottoassiemi precedentemente ricavati, dopodiché la generazione delle sequenze finali avviene sostituendo e combinando tutte le sottoparti che compongono l'assieme stesso. Proprio sulla base della potenza dell'algoritmo applicato, si suppone che la scelta migliore consista nel diversificare due diverse selezioni: la selezione delle sottosequenze per ciascun sottoassieme e la selezione delle sequenze finali generate all'ultimo livello di riduzione. In particolare, dal momento che le sequenze finali non sono altro che una semplice combinazione delle sottosequenze generate per i sottoassiemi, risulta conveniente selezionare in primo luogo i sottoassiemi, riducendo così il numero totale di domande poste all'utente. Sulla base di quanto appena detto, verrà esaminata separatamente la selezione dei sottoassiemi dalla selezione delle sequenze finali.

#### ***Fattibilità delle sottosequenze per i sottoassiemi generati***

Ogni volta che un sottoassieme viene individuato e ridotto, per esso vengono generate le corrispondenti sottosequenze di assemblaggio dello stesso. Il numero delle sottosequenze generate per ogni sottoassieme dipende dalla teoria di riduzione applicata e varia caso per caso a seconda del numero di parti che lo costituiscono. Per questo motivo non è possibile impostare delle domande generiche valide per tutti i casi, ma bisogna interrogare l'utente volta per volta su tutte le sottosequenze generate. Quindi, il numero totale di domande poste all'utente sarà pari alla somma di tutte le sottosequenze generate. È importante notare che questa procedura di domanda-risposta dovrà essere compiuta osservando l'ordine di riduzione eseguito, in maniera tale da evitare di porre più volte la stessa domanda. Riprendendo in considerazione l'esempio della Valvola Europa precedentemente trattato, la valutazione delle sottosequenze generate per ogni sottoassieme è mostrata in Tabella 4.2. Il totale di domande da porre sui sottoassiemi in questo particolare caso sarà pari alla somma delle sottosequenze generate, ossia 4.

Tabella 4.2 Selezione delle sottosequenze dei sottoassiemi per l'esempio Valvola Europa.

Nodo Base	Sottosequenza	Fattibilità
GUIDA	GUIDA-COPERCHIO-MOLLA	NO
	GUIDA-MOLLA-COPERCHIO	SI
COPERCHIO	COPERCHIO-MOLLA-GUIDA	SI
	COPERCHIO-GUIDA-MOLLA	NO

Una volta selezionata la fattibilità di tutte le sottosequenze generate, si procede con l'eliminazione di tutte quelle sequenze che presentano al loro interno quelle sottosequenze classificate come non fattibili. In particolare, a seguito della pulizia delle sottosequenze non fattibili, restano da caratterizzare  $2 \cdot 6 = 12$  possibili sequenze candidate come valide. Il fatto di interrogare prima l'utente sulla fattibilità dei sottoassiemi comporta una notevole riduzione sul numero totale di domande da porre al fine di eliminare tutte le sequenze non valide. Questo perché, proprio per la natura del metodo, le sottosequenze generate possono presentarsi in più sequenze finali, a maggior motivo se i sottoassiemi sono stati generati nei primi livelli di riduzione.

### ***Fattibilità delle sequenze finali generate***

Una volta identificata la bontà di tutte le sottosequenze generate resta da svolgere la valutazione delle sequenze finali rimanenti, in quanto non è detto che anch'esse siano tutte fattibili. La sezione delle sequenze finali di assemblaggio viene svolta in maniera differente, dal momento che si cerca di trovare un metodo univoco che elimini tutte le sequenze di assemblaggio non valide attraverso un numero di domande inferiore al numero delle sequenze stesse. Se così non fosse basterebbe valutare ogni sequenza singolarmente seguendo passo per passo tutte le attività di montaggio che la costituiscono. Il che comporterebbe una elevata mole di lavoro di selezione da parte del progettista. Infatti, per assiemi con un discreto numero di parti, il numero di sequenze generate potrebbe essere elevato e di conseguenza l'utente si troverebbe davanti ad un elevatissima mole di attività decisionali. Si identificherà così come ottimale, quella metodologia che con il minor numero di domande possibili riesca ad eliminare tutte e soltanto le sequenze di assemblaggio non fattibili.

A tal proposito, diverse metodologie applicative sono state testate, qualitativamente e quantitativamente, al fine di trovare la migliore. Di

seguito non verranno elencate tutte le strategie selettive provate, ma verrà illustrata solamente quella che a seguito di diverse prove su esempi pratici è risultata la migliore.

In questo caso, a differenza del caso precedente, si cerca una tipologia di domande standard che possa venir volta per volta applicata al caso specifico. In particolare si presuppone di imporre, in maniera ricorsiva delle domande di questo tipo:

**“ E’ possibile montare [ Discriminante ] su [ Corpo della domanda ] ? ”**

Dove il discriminante della domanda è sempre composto da un unico elemento, che sia esso un sottoassieme generato od una parte originaria, mentre il corpo della domanda può essere composto da  $n$  elementi che siano essi parti originarie o sottoassiemi generati.

Bisogna porre delle ulteriori precisazioni riguardanti il corpo della domanda, in particolare si denota che:

- il corpo della domanda si considera come un unico sottoassieme fittizio già rigidamente assemblato, in cui le parti costituenti non hanno possibilità di movimento reciproco. Per questo motivo non conta l’ordine che gli elementi hanno all’interno del corpo della domanda, ma solamente la loro presenza o meno.
- Il corpo della domanda può contenere da  $1$  a  $N - 1$  parti già montate tra di loro, dove  $N$  indica il numero totale delle parti che costituiscono l’assieme.

L’approccio selettivo sistematico proposto prevede l’imposizione di quesiti della tipologia appena esposta per ogni sequenza. Si comincia imponendo l’ultimo elemento della sequenza quale discriminante della domanda e tutti gli elementi non considerati della stessa sequenza quali il corpo della domanda ( intesi come un unico sottoassieme ). Se l’esito della domanda risulta positivo, si procede nella valutazione della sottosequenza composta dal corpo stesso della domanda, il procedimento prosegue fino ad arrivare a considerare solo i primi due elementi. Se, contrariamente, l’esito di una delle domande poste risulta negativo, si procede ad eliminare la sequenza considerata oltre a tutte le sequenze che presenterebbero a quel livello lo stesso discriminante e lo stesso corpo della domanda. Se la sequenza riceve tutte risposte con esito positivo, viene considerata fattibile e si procede a considerare la sequenza successiva. Per evitare di riporre delle domande uguali, ogni quesito viene memorizzato in modo tale che ogni volta che la domanda si ripresenti, la sua risposta venga fornita automaticamente.

### 4.2.1 Valutazione della tecnica di selezione proposta

#### Valutazione di un esempio pratico

Per comprendere meglio la tecnica selettiva proposta si consideri l'esempio di Figura 4.1 e le sue corrispondenti sequenze generate di Tabella 4.1.

Tabella 4.3 Selezione delle sequenze per l'esempio di Figura 4.1.

<i>N°</i>	<i>Sequenza</i>	<i>Domanda</i>	<i>Risposta</i>	<i>Numer o doman de</i>	<i>Fattibilità</i>
1	A<B<C<D	E' possibile montare D su (A,B,C)?	SI	1	SI
		E' possibile montare C su (A,B)?	SI	2	
		E' possibile montare B su (A)?	SI	3	
2	A<B<D<C	E' possibile montare C su (A,B,D)?	NO	4	NO
3	A<D<B<C				NO
4	A<D<C<B	E' possibile montare B su (A,D,C)?	NO	5	NO
5	B<A<C<D	E' possibile montare D su (A,B,C)?	SI		SI
		E' possibile montare C su (A,B)?	SI		
		E' possibile montare A su (B)?	SI		
6	B<A<D<C				NO
7	B<C<A<D	E' possibile montare D su (A,B,C)?	SI		SI
		E' possibile montare A su (B,C)?	SI	6	
		E' possibile montare C su (B)?	SI	7	
8	B<C<D<A	E' possibile montare A su (B,C,D)?	NO	8	NO
9	C<B<A<D	E' possibile montare D su (C,B,A)?	SI		SI
		E' possibile montare A su (C,B)?	SI		
		E' possibile montare B su (C)?	SI		
10	C<B<D<A				NO
11	C<D<A<B				NO
12	C<D<B<A				NO
13	D<A<B<C				NO
14	D<A<C<B				NO
15	D<C<A<B				NO
16	D<C<B<A				NO

La Tabella 4.3 illustra il metodo selettivo proposto per l'esempio di Figura 4.1. Con un totale di 8 domande all'utente è possibile selezionare automaticamente le uniche 4 sequenze di assemblaggio realmente fattibili tra tutte le 16 sequenze precedentemente generate. Il metodo selettivo così proposto, permette di selezionare tra tutte le sequenze generate solo quelle fattibili attraverso l'interrogazione dell'utente a domande in numero inferiore alla quantità delle sequenze stesse; permettendo così un risparmio di tempo e di sforzo di elaborazione. Inoltre, il metodo si presta ad una successiva implementazione al calcolatore che velocizzerebbe ulteriormente la procedura. È importante notare che il numero delle domande poste per arrivare all'iterazione del metodo non dipende dall'ordine considerato delle sequenze, ma solamente dalla tipologia delle sequenze create.

#### *Valutazione dell'esempio AFI di De Fazio e Whitney*

Si consideri a questo punto l'esempio AFI proposto da De Fazio e Whitney. Come è stato precedentemente visto, il numero totale delle sequenze generate attraverso l'applicazione dell'algoritmo per l'esempio in questione, è pari a 96. Come prima azione si valuta la fattibilità delle sottosequenze generate per tutti i sottoassiemi, il che corrisponde alla risposta di  $1 + 4 + 2 + 2 = 9$  domande di fattibilità.

Tabella 4.4 Selezione delle sottosequenze per l'esempio AFI di De Fazio e Whitney.

N °	Sottoassieme	Sottosequenza	Risposta	N ° Tot di Domande all'utente
1	SAN11(1)	E<F	SI	1
2	SAC21(1)	A<C<B<G	NO	2
		A<C<G<B	NO	3
		A<G<C<B	NO	4
		A<G<B<C	SI	5
3	SA(JHL)	H<L<J	NO	6
		H<J<L	SI	7
4	SAN31(1)	SAC21(1)<D<SAN11(1)	SI	8
		SAC21(1)<SAN11(1)<D	NO	9

Una volta scartate tutte le sottosequenze che non sono fattibili, si procede alla valutazione delle sequenze secondo la metodologia di selezione sistematica proposta, come mostrato in Tabella 4.5.



Tabella 4.5 Selezione delle sequenze per l'esempio AFI di De Fazio e Whitney.

N °	Sequenza	Domanda	Risposta	N ° Tot di domande all'utente	Fattibilità della sequenza
1	SAN31(1)-K-SA(JHL)	E' possibile assemblare SA(JHL) a (SAN31(1),K)?	SI	1	SI
		E' possibile assemblare K a (SAN31(1))?	SI	2	
2	SAN31(1)-SA(JHL)-K	E' possibile assemblare K a (SAN31(1), SA(JHL))?	NO	3	NO
3	K-SAN31(1)-SA(JHL)	E' possibile assemblare SA(JHL) a (SAN31(1),K)?	SI		SI
		E' possibile assemblare SAN31(1) a (K)?	SI		
4	K-SA(JHL)-SAN31(1)	E' possibile assemblare SAN31(1) a (K ,SA(JHL))?	SI	4	SI
		E' possibile assemblare SA(JHL) a (K)?	SI	5	
5	SA(JHL)-K-SAN31(1)	E' possibile assemblare SAN31(1) a (K ,SA(JHL))?	SI		SI
		E' possibile assemblare K a (SA(JHL))?	SI		
6	SA(JHL)-SAN31(1)-K	E' possibile assemblare K a (SAN31(1), SA(JHL))?	NO		NO

Tabella 4.6 Sequenze fattibili ottenute con il metodo proposto applicato all'esempio AFI di De Fazio e Whitney.

N °	Sequenza fattibile
1	A<G<B<C<D<E<F<K<H<J<L
2	K<A<G<B<C<D<E<F<H<J<L
3	K<H<J<L<A<G<B<C<D<E<F
4	H<J<L<K<A<G<B<C<D<E<F

In definitiva, si ha che attraverso l'interrogazione dell'utente su 14 domande di fattibilità si ottengono  $(1 \cdot 1 \cdot 1 \cdot 1 \cdot 4) = 4$  possibili sequenze di assemblaggio fattibili, come mostrato in

Tabella 4.6 . Dunque, questo approccio genera in maniera del tutto automatica 96 differenti sequenze di assemblaggio fattibili e non fattibili, dopodiché il numero viene successivamente ridotto a 4 possibili soluzioni fattibili, rispondendo a 14 domande di fattibilità.

Mentre, attraverso l'applicazione del metodo di De Fazio e Whitney, ponendo 36 domande a del personale esperto vengono prodotte 16 soluzioni di assemblaggio fattibili; queste sono rappresentate nella Tabella 4.7.

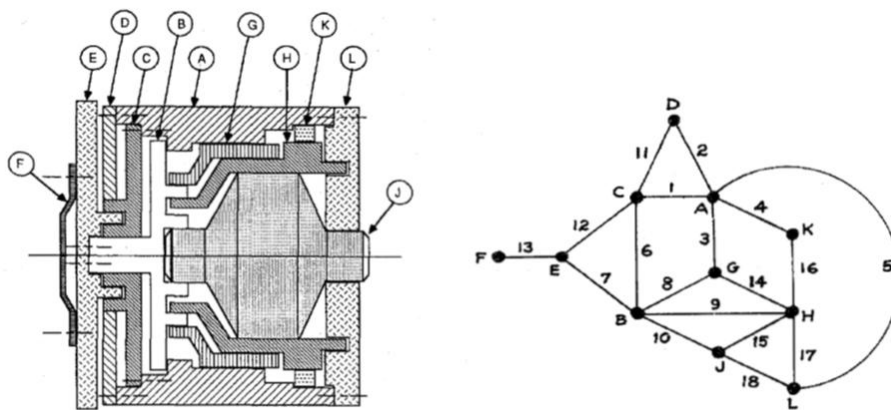


Figura 4.3 Esempio AFI De Fazio e Whitney e grafo con collegamenti numerati.

Si noti che la tipologia di soluzioni ricavata è differente, essendo le due tecniche di generazione delle sequenze di natura differente. Il metodo di De Fazio e Whitney fornisce le sequenze, secondo l'ordine dei collegamenti che è possibile effettuare volta per volta, mentre il metodo proposto fornisce una sequenza lineare delle parti da montare. Nonostante questo, è possibile rendere le due tipologie di risultati omogenee attraverso una semplice rielaborazione dei risultati. In particolare, riscrivendo i risultati ottenuti da De Fazio e Whitney secondo una sequenza lineare delle parti da montare è possibile paragonare i risultati ottenuti.

Tabella 4.7 Soluzioni fornite dall'applicazione del metodo De Fazio e Whitney.

	STEP 1	STEP 2	STEP 3	STEP 4	STEP 5	STEP 6	STEP 7	STEP 8	STEP 9	STEP 10	
<b>SEQUENZE</b>	1	3	4	14-16	15	5-17-18	8-9-10	1-6	2-11	7-12	13
	2	3	14	15	4-16	5-17-18	8-9-10	1-6	2-11	7-12	13
	3	3	14	4-16	15	5-17-18	8-9-10	1-6	2-11	7-12	13
	4	4	3	14-16	15	5-17-18	8-9-10	1-6	2-11	7-12	13
	5	14	3	4-16	15	5-17-18	8-9-10	1-6	2-11	7-12	13
	6	14	3	15	4-16	5-17-18	8-9-10	1-6	2-11	7-12	13
	7	14	15	3	4-16	5-17-18	8-9-10	1-6	2-11	7-12	13
	8	14	15	17-18	16	3-4-5	8-9-10	1-6	2-11	7-12	13
	9	15	14	3	4-16	5-17-18	8-9-10	1-6	2-11	7-12	13
	10	15	14	17-18	16	3-4-5	8-9-10	1-6	2-11	7-12	13
	11	15	17-18	14	16	3-4-5	8-9-10	1-6	2-11	7-12	13
	12	18	15-17	14	16	3-4-5	8-9-10	1-6	2-11	7-12	13
	13	6	1	2-11	7-12	13	3-8	4	9-14-16	10-15	5-17-18
	14	6	1	2-11	7-12	13	3-8	9-14	4-16	10-15	5-17-18
	15	6	1	2-11	7-12	13	3-8	9-14	10-15	4-16	5-17-18
	16	6	1	2-11	7-12	13	4	3-8	9-14-16	10-15	5-17-18

**Collegamenti**

Paragonando i risultati ottenuti è possibile trarre le seguenti conclusioni:

- il metodo di De Fazio e Whitney fornisce un numero maggiore di soluzioni fattibili; nonostante questo, il numero di interrogazioni all'utente in rapporto alle sequenze fattibili ottenute è maggiore per il metodo proposto.
- Le soluzioni ottenute attraverso il metodo qui proposto derivano dalla teoria dei grafi e per questo rappresentano sempre un percorso Hamiltoniano. Ciò non accade per le soluzioni ottenute da De Fazio e Whitney che si basano solamente sulle relazioni di precedenza imposte dal progettista.
- Il metodo qui proposto fornisce delle ulteriori soluzioni valide che il metodo di De Fazio e Whitney non individua.

Tabella 8 Sequenze fattibili ottenute da De Fazio e Whitney per l'esempio AFI.

N ° sequenza	Sequenza fattibile
1	A<G<K<H<J<L<B<C<D<E<F
2	A<G<H<J<K< L<B<C<D<E<F
3	A<G<H<K<J< L<B<C<D<E<F
4	A<K<G<H<J< L<B<C<D<E<F
5	G<H<A<K<J< L<B<C<D<E<F
6	G<H<A<J<K< L<B<C<D<E<F
7	G<H<J<A<K< L<B<C<D<E<F
8	G<H<J<L<K< A<B<C<D<E<F
9	J<H<G<A<K< L<B<C<D<E<F
10	J<H<G<L<K< A<B<C<D<E<F
11	J<H<L<G<K< A<B<C<D<E<F
12	J<L<H<G<K< A<B<C<D<E<F
13	C<B<A<D<E<F<G<K<H<J<L
14	C<B<A<D<E<F<G<H<K<J<L
15	C<B<A<D<E<F<G<H<J<K<L
16	C<B<A<D<E<F<K<G<H<J<L

Valutazione dell'esempio valvola a sfera

Si consideri l'esempio riportato nel capitolo precedente riguardante l'assieme valvola a sfera. Si era ricavato che tramite l'applicazione del metodo automatico proposto su tale esempio fornisce una totalità di 96 possibili sequenze di assemblaggio fattibili e non fattibili. Come negli altri casi, si vuole a questo punto valutare la fattibilità o meno delle sequenze di montaggio ricavate precedentemente.

Si procede dunque con la selezione iniziale delle sottosequenze generate per ogni sottoassieme creato; la tabella mostra la selezione effettuata manualmente delle sottosequenze considerate valide ed il numero di domande poste all'utente. Il numero di domande poste all'utente in questa fase è sempre pari al numero totale di sottosequenze generate.

Tabella 4.9 Selezione delle sottosequenze per l'esempio Valvola a sfera.

N °	Sottoassieme	Sottosequenza	Risposta	N ° Tot di Domande all'utente
1	SAC11(1)	$A < B < C < D$	NO	1
		$A < B < D < C$	NO	2
		$A < D < B < C$	NO	3
		$A < D < C < B$	SI	4
2	SAC11(2)	$A < B < C < E$	NO	5
		$A < B < E < C$	NO	6
		$A < E < C < B$	SI	7
		$A < E < B < C$	NO	8
3	SAN21(1)	$SAC11(1) < G < E$	SI	9
		$SAC11(1) < E < G$	SI	10
4	SAN22(1)	$SAC11(2) < G < D$	SI	11
		$SAC11(2) < D < G$	SI	12

Una volta scartate tutte le sottosequenze che non sono fattibili, si procede alla valutazione delle sequenze secondo la metodologia di selezione sistematica proposta, come mostrato in Tabella 4.10.

Tabella 4.10 Selezione delle sequenze per l'esempio AFI di De Fazio e Whitney.

N °	Sequenza	Domanda	Risposta	N ° Tot di domande all'utente	Fattibilità della sequenza
1	$SAN21(1) < F < H$	E' possibile montare H su (SAN21(1),F)?	SI	1	SI
		E' possibile montare F su SA21(1)?	SI	2	
2	$SAN21(1) < H < F$	E' possibile montare F su (SAN21(1),H)?	NO	3	NO
3	$H < F < SAN21(1)$	E' possibile montare SAN21(1) su (H,F)?	NO		NO
4	$H < SAN21(1) < F$				NO
5	$F < H < SAN21(1)$				NO
6	$F < SAN21(1) < H$	E' possibile montare H su (SAN21(1),F)?	SI	4	SI
		E' possibile montare F su SAN21(1)?	SI	5	
7	$SAN22(1) < F < H$				SI
8	$SAN22(1) < H < F$				NO
9	$H < F < SAN22(1)$				NO
10	$H < SAN22(1) < F$				NO
11	$F < H < SAN22(1)$				NO
12	$F < SAN22(1) < H$				SI

Si noti che, essendo il sottoassieme  $SAN21(1)$  composto dalle medesime parti del sottoassieme  $SAN22(1)$ , siccome nella ripetizione delle domande non ha importanza tenere conto dell'ordine degli elementi, una domanda che

presenterà nel corpo o nel discriminante il sottoassieme *SAN21(I)* sarà del tutto analoga ad una domanda che presenterà nel corpo o nel discriminante il sottoassieme *SAN22(I)*. In definitiva, si ottiene che attraverso l'interrogazione dell'utente su 17 domande di fattibilità si ottengono  $(4 \cdot 2 \cdot 1) = 8$  possibili sequenze di assemblaggio fattibili. Per concludere, questo approccio genera in maniera del tutto automatica, ancora una volta, 96 differenti sequenze di assemblaggio fattibili e non fattibili ed il numero di soluzioni viene successivamente ridotto a 8 possibili soluzioni fattibili, rispondendo a 17 domande di fattibilità.

Tabella 11 Sequenze fattibili ottenute con il metodo proposto applicato all'esempio Valvola a sfera.

N ° sequenza	Sequenza fattibile
1	A<D<C<B<G<E<F<H
2	A<D<C<B<E<G<F<H
3	F<A<D<C<B<G<E<H
4	F<A<D<C<B<E<G<H
5	A<E<C<B<G<D<F<H
6	A<E<C<B<D<G<F<H
7	F<A<E<C<B<G<D<H
8	F<A<E<C<B<D<G<H

## CAPITOLO 5

# CONSIDERAZIONI FINALI

### 5.1 Conclusioni

Questa tesi è stata sviluppata avendo come scopo principale quello di poter presentare e sviluppare, attraverso l'uso di un calcolatore, un approccio automatico e sistematico adoperabile in supporto al processo di pianificazione delle sequenze di assemblaggio, di un generico prodotto meccanico. Lo strumento che è stato così sviluppato e validato, permetterà al designer stesso di minimizzare il ricorso a decisioni soggettive riducendo gli sforzi di decision making durante l'elaborazione di un piano valido.

Inoltre, avendo a disposizione una moltitudine di soluzioni di assemblaggio, anziché una singola soluzione sviluppata dall'esperienza del personale addetto, si potrà dare una visione completa del possibile campo di soluzioni realizzabili, derivanti dalla natura combinatoria del problema.

L'approccio proposto permetterà così, sia al personale specializzato che a quello non specializzato, di ottenere automaticamente almeno un piano di montaggio applicabile ad un generico prodotto, composto da un discreto numero di parti. Le sequenze verranno generate basandosi unicamente sulle informazioni di contatto estraibili dal modello 3D-CAD dell'assieme considerato. Questo tipo di approccio risulta molto differente dalle metodologie convenzionali applicate nell'ASA, poiché non tiene in considerazione delle domande di precedenza fatte sulle attività di montaggio. Questa prerogativa porta con sé il notevole svantaggio per cui il metodo, pur generando i risultati cercati, produrrà sequenze non sempre valide a livello pratico, nonostante siano del tutto coerenti alla teoria applicata. Mediante una successiva analisi, in cui verrà richiesto l'intervento umano, sarà poi possibile filtrare il numero iniziale di soluzioni, rimuovendo quelle impossibili dalla totalità dei risultati ricavati. Si consideri inoltre che il grafo



non direzionato dei collegamenti non fornisce una rappresentazione univoca dell'assieme meccanico, in quanto due differenti assiemi meccanici potrebbero venir rappresentati dallo stesso grafo dei collegamenti. Dal momento che le sequenze si ottengono basandosi sulle sole informazioni presenti nel grafo, se due differenti assiemi meccanici presentano lo stesso grafo dei collegamenti, le sequenze generate per entrambi saranno le medesime.

Diversi studi di ricerca si sono dedicati ad analizzare la possibilità di attingere direttamente al database aziendale per ricavare le informazioni necessarie del prodotto. E' quindi importante avere una gestione avanzata dei dati relativi al prodotto in tutte le diverse fasi di sviluppo dello stesso, in modo tale che vengano sempre adottate delle scelte coerenti tra loro e nei risultati delle soluzioni ottenute. Tuttavia, gli approcci che in passato sono stati proposti dai ricercatori, raramente vengono applicati in ambito industriale, in quanto la loro complessità aumenta in maniera esponenziale con la complessità del sistema in analisi. Inoltre, questi richiedono un numero considerevole di interrogativi specifici, relativi alla tipologia di montaggio; questo causa una perdita di tempo nell'elaborare una risposta mentre, adottando delle scelte soggettive e basate sull'esperienza, ciò non accade.

Per concludere, queste metodologie basate su domande e risposte a specifici quesiti, richiedono il coinvolgimento di personale tecnico che abbia il massimo livello di indicazioni sul prodotto considerato. Di contro, si è mostrato che nell'approccio proposto, a qualsiasi livello di dettaglio purché sia nota la disposizione dei componenti, le sole informazioni di contatto tra le parti sono sufficienti a definire le sequenze di assemblaggio senza l'intervento di personale addetto.

## **5.2 Analisi dei risultati**

Come per qualunque altro approccio di pianificazione delle sequenze, il numero di risultati ottenuti è direttamente proporzionale al numero di componenti ed al numero di collegamenti tra esse. Nonostante per un assieme composto da un numero discreto di parti, si ottenga una considerevole quantità di sequenze, la rappresentazione dei risultati in modo implicito (ossia attraverso l'uso dei sottoassiemi generati) permette una facile rintracciabilità delle soluzioni irrealizzabili: questo è reso possibile dalla natura dello stesso metodo che basa la sua generazione delle sequenze sull'individuazione di possibili sottoassiemi.

E' necessario considerare che una buona parte dei parametri di progettazione soggettivi, non sufficientemente quantificabili da poter essere implementati metodicamente, quali quelli relativi al processo di produzione manifatturiero, vengono valutati nelle fasi finali dell'intera progettazione. Per essi, un loro improvviso cambiamento comporta una perdita di tempo nel dover rielaborare il progetto, causando una relativa perdita di profitto da parte dell'impresa.

Ogni volta che un assieme viene modellato, viene implicitamente supposta una relativa sequenza di assemblaggio. L'approccio considerato è indipendente da questo fattore, in quanto la tecnica di modellazione usata non viene coinvolta nel metodo. Di contro e come già detto, per questo motivo, il metodo non identifica solamente le sequenze fattibili. Nonostante ciò, a differenza di altri metodi, senza domande di precedenza si riesce a ridurre automaticamente il numero delle sequenze ottenute, garantendo tra esse la presenza di almeno una fattibile. Il metodo proposto permette una rapida analisi delle sequenze di montaggio relative ad un assieme. Questo permette di analizzare più volte il modello variando volta per volta alcuni componenti specifici e valutando la loro influenza sulle sequenze ottenute. Così facendo sarà possibile modificare, anche nelle prime fasi della progettazione alcuni parametri progettuali che garantiscano successivamente un assemblaggio ottimale. Questa analisi infatti, può essere anche utile nella valutazione della forma del prodotto e della geometria delle parti nelle fasi iniziali del ciclo di sviluppo del prodotto.

Nel caso il metodo non produca nessuna sequenza fattibile, sarà necessario controllare il modello considerato, verificando la presenza di condizioni di instabilità o di iperstaticità di una delle parti costituenti.

Il presente approccio di identificazione restituisce i risultati in tempi di calcolo contenuti, a patto che i sottoassiemi individuati volta per volta siano composti da un numero limitato di parti.

### **5.3 Sviluppi futuri**

L'approccio proposto apre la possibilità di una migliore integrazione della pianificazione di montaggio nelle le prime fasi della progettazione. Questo perché nessun parametro progettuale di dettaglio viene richiesto durante l'elaborazione dei risultati.

Inoltre, tramite degli ulteriori sviluppi, ci si aspetta che il metodo sia direttamente applicabile per mezzo di sistemi della gestione dei dati del prodotto (PDM) o della gestione del ciclo di vita del prodotto (PLM).

L'approccio proposto per la definizione automatica delle sequenze di montaggio è stato validato attraverso delle casistiche base ricorsive e attraverso degli esempi estratti dalle teorie presenti nella bibliografia. Successivamente si potrebbe prevedere di applicare il metodo su casi tipicamente industriali molto più complessi e presentare i risultati ottenuti a del personale esperto con lo scopo di sviluppare delle soluzioni sempre più concrete alla realtà costruttiva.

Un'ulteriore applicabilità del metodo sviluppato può essere il suo utilizzo nell'individuazione e nell'analisi dei sottoassiemi che è possibile considerare all'interno di un prodotto assemblato. Infatti, il metodo permette l'identificazione in modo univoco di cluster o di gruppi di parti con delle caratteristiche in comune. L'approccio di identificazione dei sottoassieme si presta ad una migliore configurazione industriale delle workstation in cui vengono introdotte più attività di montaggio con lo scopo di accelerare il processo di produzione.

Inoltre, al fine di ridurre il numero di sequenze infattibili che vengono automaticamente generate dal metodo, si potrebbe pensare di integrare la tecnica di interrogazione e risposta delle domande di precedenza all'interno dell'algoritmo. In questo modo è possibile indirizzare la generazione delle sequenze durante ogni passaggio di riduzione del grafo.

Dal momento che il software di generazione è stato sviluppato per la prima volta, sarebbe concepibile pensare che le routine sviluppate seguendo l'intuito dell'autore, possano venir successivamente analizzate e ottimizzate al fine di ridurre la mole di operazioni richieste al calcolatore e quindi il tempo di elaborazione dei risultati. In aggiunta a questo, un possibile studio delle tempistiche necessarie al conseguimento dei risultati permetterebbe una maggiore comprensione delle problematiche legate al metodo applicato.

Per concludere, siccome il metodo prima di essere applicato richiede che ogni organo di collegamento rigido (quali viti, ralle, bulloni, tiranti ecc.) venga omesso. Un aggiuntiva implementazione allo stesso, che prenderebbe in considerazione anche la loro influenza, fornirebbe una sua maggiore applicabilità in ambiti industriali.



## *Riferimenti bibliografici*

- [1] **Baldwin D.F. [et al.]** Assembly planning effectiveness using virtual reality [Journal] // IEEE Trans. Robotics and Automn. -1991. – 1 : Vol.7. – pp. 78-94.
- [2] **Banerjee E. [et al.]** Assembly planning effectiveness using virtual reality [Journal] // Presence: Teleoperators and Virtual Environments. – USA : Massachusetts Institute of Technology Press, April 1999 . – 2 : Vol. 8. –pp.204-217. – ISSN 1054-7460.
- [3] **Boothroyd Geoffrey and Knight Winston** Design for assembly [Journal] // IEEE Spectrum. – Settembre 1993. – pp. 53-55.
- [4] **Bordegoni Monica e Cugini Umberto** Haptic modeling in the conceptual phases of product design [Journal] // Virtual Reality. – 2006. –Vol. 9. – pp. 192-202.
- [5] **Borjault A.** Contribution à une approche méthodologique de l'assemblage automatisé: élaboration automatique des sequences opératoires. [Report] : PhD Thesis / L'univeristé de Franche-Compté. – France : [s.n.], 1984.
- [6] **De Fazio Thomas L. [et al.]** Aids for the design or choice of assembly sequences [Conference] // Proceedings 1989 IEEE International Conference on System, Man and Cybernetics. – Cambridge, MA, USA : Charles Stark Draper Lab. Inc., 1989. – Vol 1. – pp.61-70.
- [7] **De Fazio Thomas L. And Whitney Daniel E.** Simplified generation of all mechanical assembly sequences. [Journal] // IEEE Journal of robotics and automation. – [s.l.] : IEEE, December 1987. – 6 : Vols. Ra-3. – pp. 640-658.
- [8] **De Fazio Thomas L., Rhee Stephen J. And Whitney Daniel E.** Design-specific approach to design for assembly (DFA) for complex mechanical assemblies [Journal] // IEEE, Transaction on Robotics and Automation. – October 1999. – 5 : Vol. 15 . – pp. 869-880

- 
- [9] **Delchambre A.** A pragmatic approach to computer aided assembly planning. [Conference] // IEEE International Conference on Robotics and Automation. – 1990. – Vol. 3. – pp.1600-1605.
- [10] **Deo M.P.N. and Krishnamoorty M.S.** Algorithms for generating fundamental cycles in a graph [Journal] // ACM Transaction on Mathematical Software. – 1982. – 1 : Vol. 8. – pp. 26-42.
- [11] **Diestel Reinhard** Graph Theory [Book]. – New York : Springer-Verlag Heidelberg, 2005. – III.
- [12] **Eppstein David** Spanning Trees and spanners [Report] // Information and computer science ; University of California. – Irvine : [s.n.], 1996.
- [13] **Gibbs Norman E.** A cycle generation algorithm for finite undirected linear graphs [Journal] // Journal of the association for computing machinery. – October 1969. – 4 : Vol.16. – pp. 564-568.
- [14] **Gotlieb C.C. and Corneil D.G.** Algorithms for finding a fundamental set of cycles for an undirected linear graph [Journal] // Communications of the ACM. – December 1967. – 12 : Vol. 10. – pp.780-783.
- [15] **Homem de Mello Luiz S. And Lee Sukhan** Computer-Aided Mechanical Assembly Planning [Book]. – Boston : Kluwer Academic Publishers, 1991. – ISBN 0-7923-9205-1.
- [16] **Homem de Mello Luiz S. And Sanderson Arthur C.** AND/OR graph representation of assembly plans [Journal] // IEEE Transaction on Robotics and Automation. – April 1990. – 2 : Vol 6. – pp. 188-199.
- [17] **Homem de Mello Luiz S. And Sanderson Arthur C.** Representation of mechanical assembly sequences [Journal] // IEEE Transaction on Robotics and Automation. – 1991. – 2 : Vol 7. – pp. 211-237.
- [18] **Hwai-En Tseng, Chien-Chen Chang, Jia-Diann Li** Modular design to support green life-cycle engineering [Journal] // European Journal of Operational Research, Volume 29, Issue 1, April 1987, Pages 74–82.

- [19] **Kang Byungkyu** Design of survivable WDM network based on pre-configured protection cycle [Report] : Thesis of Master of Science / Edith Cowan University. – Western-Australia : [s.n.], 2006. – pp. 91.
- [20] **Lai Hsin-Yi and Huang Chin-Tzww** A systematic approach for automatic assembly sequence plan generation [Journal] // The international Journal of Advanced Manufacturing Technology . – London : Springer-Verlag, 8 September 2004. – 9-10 : Vol. 24. – pp.. 752-763.
- [21] **Pahl Gerhard [et al.]** Engineering Design – A Systematic Approach [Book] / ed. Springer-Verlag. – London,2007.- Third.
- [22] **Ritchie J.M, Dewar R.G. e Simmons J.E.L.** The generation and practical use of plans for manual assembly using immersive virtual reality [Journal] // Proceedings of the Institution of Mechanical Engineer / ed. Ltd. Mechanical Engineering Publications. – London : [s.n.], 1999. <5 : Vol. 213 – pp.461-474.
- [23] **Schank Thomas and Wagner Dorothea** Approximating clustering coefficient and transitivity [Journal] // Journal of graph algorithms and applications. – 2005. – 2 : Vol. 9. – pp. 265-275.
- [24] **Sharpe John E.** Computer Tools for Integrated Conceptual Design [Journal] // Design Studies.-Great Britain : Elsevier Science, 1995.- Vol. 16. Pp 471-488.
- [25] **Ulrich Karl and Eppinger Steven** Product Design and Development [Book] / ed. Hill Mc.-2000.-ISBN 0-07-116993-8.
- [26] **Viganò R. and Osorio-Gomez G.** A computer tool to extract feasible assembly sequence from a product CAD model, in automate way.// International Conference on Innovative Methods in Product Design – Proceedings of IMProve, San Servolo, Venice (Italy),2011. pp. 248-253.
- [27] **Viganò R. and Osorio-Gomez G.** Assembly sequence definition through hierarchical subassembly approach.// In Proceedings of TMCE 2010, Ancona, Italy, 2010. pp. 951-960.
- [28] **Watts Duncan J. and Strogatz Steven H.** Collective dynamics of “small worlds” network [Journal] // Nature. – 4 June 1998. – Vol. 393. – pp. 440-442.

- 
- [29] **Wesley M. A. [et al.]** A geometric modeling system for automated mechanical assembly. [Journal] // IBM J.Res and Dev.. – 1980. – 1 : Vol. 24. – pp. 64-67.
- [30] **Whitney D.E. [et al.]** Designing Assemblies [Journal] // Research in Engineering. – London : Springer-Verlag,1999. – Vol. 11 pp. 229-253.
- [31] **Whitney D.E. [et al.]** Introducing knowledge-based engineering into an interconnected product development process [Conference] // Proceedings of the 1999 ASME design engineering technical conferences. – Las Vegas, Nevada : [s.n.], 1999. – pp. 1-10.