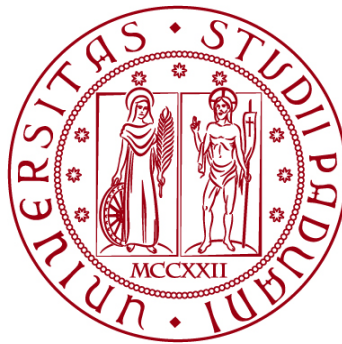


**UNIVERSITÀ DEGLI STUDI DI PADOVA**  
DIPARTIMENTO DI INGEGNERIA CIVILE, EDILE E AMBIENTALE  
*Department Of Civil, Environmental and Architectural Engineering*

Corso di Laurea Magistrale in Ingegneria Edile - Architettura



**TESI DI LAUREA**

**Ottimizzazione tramite algoritmi genetici nel design  
parametrico. Lo studio del Toshiba IHI Pavilion di Kishō  
Kurokawa.**

Relatore:  
Chiar.mo PROF. ANDREA GIORDANO  
Correlatori:  
Chiar.mo PROF. MANUEL ALEJANDRO,  
RÓDENAS LÓPEZ

Laureando: LEONARDO ZAGO  
matricola 1195447

**ANNO ACCADEMICO 2023-2024**



# INDICE

<b>1. Introduzione.....</b>	<b>7</b>
<b>2. La geometria delle superfici.....</b>	<b>11</b>
2.1. Il concetto di superficie, definizione e primi cenni storici.....	12
2.2. La rappresentazione geometrica delle superfici.....	15
2.3. La genesi configurativa.....	17
<b>3. Fondamenti dell'ottimizzazione.....</b>	<b>21</b>
3.1. Introduzione e concetti di base dell'ottimizzazione.....	22
3.2. I modelli dell'ottimizzazione.....	25
3.2.1. Il modello matematico.....	29
3.3. Determinazione della soluzione ottimale: gli algoritmi iterativi.....	31
3.4. Ottimizzazione multi-obiettivo.....	32
3.5. Diversi tipi di ottimizzazione.....	34
3.5.1. Ottimizzazione dimensionale.....	36
3.5.2. Ottimizzazione di forma.....	37
3.5.3. Ottimizzazione topologica.....	39
<b>4. Cambiamenti nella progettazione attraverso l'approccio del design parametrico.....</b>	<b>45</b>
4.1. Definizione e concetti di base del parametro.....	48
4.1.1. Il processo parametrico.....	52
4.2. Storia e caratteristiche del design parametrico in architettura.....	56

4.3.	Benefici e limitazioni della progettazione parametrica.....	68
4.3.1.	I benefici del design parametrico.....	69
4.3.2.	Le limitazioni del design parametrico.....	70
<b>5.</b>	<b>L'utilizzo di nuove tecnologie per la progettazione parametrica.....</b>	<b>73</b>
5.1.	Modellazione algoritmica.....	76
5.1.1.	Definizione e concetti di base.....	78
5.1.2.	Algoritmi in sistemi nodali.....	85
5.2.	Gli algoritmi genetici.....	87
5.3.	Panoramica dei software parametrici principali.....	97
5.3.1.	Interfaccia e funzionalità principali di Grasshopper.....	100
5.3.2.	Strumenti e plug-ins per l'ottimizzazione mono-obiettivo.....	104
5.3.3.	Strumenti e plug-ins per l'ottimizzazione multi-obiettivo.....	110
5.3.4.	Integrazione di Karamba3D nei processi di progettazione parametrica.....	117
<b>6.</b>	<b>L'ottimizzazione del Toshiba IHI Pavilion: il caso studio.....</b>	<b>121</b>
6.1.	Il contesto storico dell'Expo '70 a Osaka, Giappone.....	123
6.2.	Kisho Kurokawa.....	130
6.2.1.	La filosofia del Movimento Metabolista: una visione organica.....	132
6.3.	Toshiba IHI Pavilion: design architettonico e visualizzazione grafica tramite Grasshopper.....	133
6.3.1.	La modellazione parametrica del Toshiba IHI Pavilion e la sua ottimizzazione tramite algoritmi genetici.....	141
6.3.1.1.	La costruzione del teatro.....	142
6.3.1.2.	La costruzione del tetraedro.....	146

6.3.1.3.	La costruzione della “foresta” di tetraedri.....	148
6.3.1.4.	La costruzione dei punti di ancoraggio fra tetraedri e teatro e l'utilizzo del plug-in Karamba 3D.....	154
6.3.1.5.	L'ottimizzazione tramite il plug-in Wallace-i.....	157
6.3.2.	Studio delle soluzioni e risposta critica.....	159
<b>7.</b>	<b>Conclusioni.....</b>	<b>163</b>
<b>8.</b>	<b>Appendice.....</b>	<b>167</b>
8.1.	Tavola 1.....	168
8.2.	Tavola 2.....	169
8.3.	Tavola 3.....	170
8.4.	Tavola 4.....	171
8.5.	Tavola 5.....	172
<b>9.</b>	<b>Bibliografia.....</b>	<b>173</b>
<b>10.</b>	<b>Ringraziamenti.....</b>	<b>179</b>



# 01 • Introduzione

Come è avvenuto anche in altri campi applicativi, l'utilizzo dei nuovi strumenti informatici nell'architettura e nell'ingegneria non solo ha velocizzato e facilitato molte operazioni che precedentemente richiedevano calcoli faticosi e molto tempo, ma ha per certi aspetti contribuito a trasformare proprio i processi progettuali. È questo l'aspetto che la presente ricerca intende indagare nelle sue articolazioni teoriche e storiche per poi giungere a uno studio applicativo.

L'analisi della geometria delle superfici costituisce in questo percorso una specie di premessa metodologica, che garantisce una conoscenza critica degli strumenti informatici e delle loro applicazioni e di conseguenza un loro utilizzo corretto ed efficace.

L'altro aspetto che è stato poi preso in analisi è l'evoluzione dell'ottimizzazione nel campo dell'ingegneria, che ha avuto un impatto significativo specialmente con l'avvento del computer nel XX secolo. Questa tecnologia, infatti, ha rivoluzionato l'analisi ingegneristica, permettendo la conversione di complessi sistemi di equazioni differenziali in problemi algebrici risolvibili attraverso calcolatori. L'ottimizzazione, intesa come la ricerca di un "ottimo" o di una soluzione che massimizzi i benefici e minimizzi i costi, è diventata così una pratica comune nell'ingegneria come anche in numerosi aspetti della vita quotidiana. Come è stato osservato, la necessità di ottimizzare percorsi, risorse e processi è intrinsecamente legata al comportamento umano, manifestandosi in scelte consapevoli o inconsapevoli di miglioramento continuo.

L'ottimizzazione è articolata in quattro fasi fondamentali: la definizione del problema, l'identificazione delle variabili decisionali, la formulazione del modello matematico e la risoluzione del problema. Questi passaggi strutturano un

processo iterativo di miglioramento, dove le variabili decisionali vengono manipolate per ottenere un risultato ottimale. Questo concetto si riflette in vari campi, dall'ingegneria all'architettura, dove l'evoluzione degli strumenti di progettazione ha permesso una rappresentazione sempre più precisa e flessibile delle idee progettuali.

A questo punto si è cercato di mostrare come il passaggio dal disegno tradizionale al design parametrico rappresenti un cambio di paradigma nella progettazione architettonica. L'uso di software avanzati, come Grasshopper per Rhinoceros, ha aperto nuove possibilità, permettendo una modellazione complessa e un'analisi accurata prima della realizzazione fisica delle strutture. Questo approccio ha facilitato la sperimentazione e l'adozione di soluzioni innovative, che altrimenti sarebbero state difficili da realizzare con metodi tradizionali.

Ho voluto mettere in evidenza come il design parametrico e la modellazione algoritmica stiano trasformando il modo di concepire e realizzare progetti, consentendo un controllo maggiore e più preciso dei modelli tridimensionali e delle informazioni. Questo approccio, basato sull'integrazione tra design e computazione, permette di risolvere problemi complessi in modo più efficiente, ottimizzando risorse e tempi. Nel vasto campo della scienza informatica, la modellazione algoritmica è fondamentale, poiché consente di tradurre problemi complessi in rappresentazioni formali risolvibili mediante algoritmi. Il lavoro degli ingegneri e architetti, automatizzando compiti ripetitivi e permettendo la generazione di molteplici soluzioni, viene facilitato. Ho preso in considerazione in modo più ravvicinato gli algoritmi genetici, un sottogruppo degli algoritmi evolutivi, che sono tecniche ispirate alla selezione naturale, utilizzate per risolvere problemi complessi tramite la creazione e l'evoluzione di popolazioni di soluzioni candidate. Questi algoritmi operano cercando di migliorare progressivamente le soluzioni, selezionando le soluzioni migliori in base alla loro "fitness," e introducendo variazioni genetiche, gli algoritmi genetici esplorano efficacemente lo spazio delle soluzioni. L'approccio a obiettivo singolo negli algoritmi genetici limita l'ottimizzazione a una sola funzione di fitness. Tuttavia, nei problemi di ottimizzazione multi-obiettivo, si cercano soluzioni non dominate,



che non possono essere migliorate in tutti gli aspetti rispetto ad altre soluzioni. Questo metodo permette di considerare contemporaneamente molteplici criteri, riflettendo meglio la complessità dei problemi reali.

Come esito conclusivo di questa ricerca si è cercato di applicare le osservazioni teoriche fin qui sviluppate a un caso preciso, che permetta di verificare l'efficacia delle nuove strategie progettuali. In questo senso il Toshiba IHI Pavilion, progettato da Kisho Kurokawa per l'Expo '70 in Giappone, rappresenta un caso studio significativo. Questo padiglione, un emblema del movimento Metabolista, ha anticipato molte delle pratiche di architettura sostenibile moderne. La sua progettazione modulare e l'utilizzo di materiali riciclabili hanno reso possibile un'architettura adattabile e trasformabile nel tempo. L'analisi di questo caso attraverso strumenti contemporanei permette di esplorare nuove modalità di ottimizzazione e di comprendere meglio l'impatto delle tecnologie digitali sulla progettazione, come si vedrà nel capitolo 6.



# 02 • La geometria delle superfici

La comprensione della "geometria delle superfici" appare fondamentale nel contesto di questa tesi di laurea, rappresentando un capitolo importante e imprescindibile per il successivo sviluppo e approfondimento degli argomenti trattati. La geometria delle superfici costituisce il fondamento concettuale su cui si erige l'intero lavoro, fornendo le basi teoriche essenziali per una corretta analisi e interpretazione dei fenomeni studiati, con particolare attenzione al caso studio descritto nel capitolo sesto. La sua inclusione iniziale si giustifica non solo per il suo ruolo di precursore logico, ma anche perché fornisce strumenti concettuali e metodologici che si rivelano indispensabili nella trattazione delle tematiche successivamente affrontate. In questo contesto, presentare la geometria delle superfici all'inizio della tesi garantisce una comprensione profonda e coerente dell'intero *corpus* di ricerca, fornendo così un solido fondamento per l'analisi e la discussione dei risultati ottenuti nei capitoli finali.

Le superfici, soprattutto nei progetti più moderni, sono spesso di una complessità elevata e ne consegue che non è facile riuscire a decifrarne la struttura geometrica. Per fare ciò, infatti, è necessario provvedere ad una completa conoscenza della genesi geometrica delle superfici stesse. Lo studio e la conoscenza delle superfici non è utile solamente per uno scopo meramente didattico, ma stimola la creatività, utile in campo progettuale, e appare indispensabile all'architettura esistente per una sua corretta osservazione, comprensione e per una migliore comunicazione. Proprio la comunicazione costituisce un punto fondamentale in campo architettonico e si posiziona alla base del sapere di un professionista. La comunicazione avviene grazie ad una buona rappresentazione grafica (disciplina fondamentale in diversi campi e in particolar modo nell'architettura e nell'ingegneria civile, cosa che verrà discussa in questo capitolo e nei capitoli successivi), la quale si configura per la cultura

occidentale come elemento di sintesi tra la realtà e l'immaginazione. Non solo, la rappresentazione grafica costituisce il lavoro teorico, l'analisi critica, la comunicazione e la documentazione, senza la quale è possibile la perdita dei valori estetici e poetici dell'opera. La rappresentazione grafica è un potente strumento per semplificare, comunicare ed esplorare informazioni complesse, contribuendo così a una comprensione più profonda e a una presa di decisioni più informata.

## 2.1. Il concetto di superficie, definizione e primi cenni storici

Stando al dizionario Treccani, il termine *superficie* ((meno com. *superficie*) s. f. [dal lat. *superficies*, comp. di *super-* e *facies* «faccia»] (pl. *-ci*, disus. *-cie*)) possiede numerosissime interpretazioni che garantiscono una complessità elevata della sua definizione e certificano che il termine *superficie* è indispensabile per ogni riflessione di tipo teorico in campi anche diversi dall'architettura. La definizione più comune viene data così: "il contorno di un corpo come elemento di separazione della regione dello spazio occupata dal corpo da quella non occupata; oppure, con riferimento a determinati corpi, la faccia esterna, più o meno piana e a due dimensioni". È possibile però definire *superficie* con accezioni specifiche: per esempio, nelle tecnologie meccaniche è sinonimo di *lavorazione*; in matematica è figura geometrica a due dimensioni che nasce per astrazione dalla nozione intuitiva di contorno di un corpo, avente le caratteristiche di una lamina pensata priva di spessore; e ancora, in fisica con riferimento a una relazione  $f(x, y, z)=0$  che descrive gli stati di un sistema governato dai parametri  $x, y, z$ ; e così via in marina, in aeronautica, nelle scienze della terra, in agraria. Come si è potuto evincere il termine *superficie* non si riferisce solamente un concetto, non è solamente ciò che noi possiamo vedere e toccare con mano. Il termine *superficie* contiene in sé anche definizioni come *orma*, *tipo* (dal greco "impronta", "traccia"). Possiamo dire quindi che la *superficie* è un elemento fisico e

geometrico, ma racchiude in sé anche molte altre espressioni complesse in varie discipline scientifiche.

Un altro aspetto fondamentale che riguarda la superficie è il fatto che possiede come qualità quella di configurare e creare spazi tramite interrelazioni tra superfici reali e non, elemento cardine che costituisce l'architettura, poiché spesso, anche se non direttamente visibile e tangibile, lo spazio domina intorno a noi e determina il nostro umore, le nostre sensazioni, le nostre emozioni e il nostro spirito. Gran parte di ciò che noi otteniamo dall'architettura, non solo da un punto di vista materico ma con una più ampia concezione del termine, artistico e spirituale, lo otteniamo grazie alle relazioni che sorgono fra gli spazi: “ quando noi costruiamo non facciamo altro che distaccare una conveniente quantità di spazi, chiuderlo e proteggerlo e tutta l'architettura sorge da questa necessità” (Scott J., *L'architettura dell'umanesimo*, Bari, 1978). Altro punto su cui soffermarsi è che anche il vuoto può essere assunto come una superficie o uno spazio come l'insieme delle relazioni tra più superfici; basti pensare come cambia l'aspetto di un edificio solo cambiando le possibili conformazioni delle aperture, progettando i pieni e i vuoti di una facciata. In egual modo i vuoti tra gli stessi edifici creano una propria identità determinando forme, geometrie, densità e relazioni.

Le prime formulazioni sulle origini della geometria avvengono tra il XIX e il XVII secolo a.C., testimoniati da antichissimi papiri dove vengono documentate per la prima volta alcune formulazioni di problemi per il calcolo di aree e semplici figure. Successivamente si possono rintracciare le prime definizioni geometriche e di superficie in epoca egizia, quando, con la necessità di determinare e misurare i terreni, nascono anche i primi studi di geometria: si ebbe la necessità di sapere la grandezza, quindi l'area, del proprio campo e che forma avesse. Infatti, la geometria nasce proprio per il bisogno di misurare in campo amministrativo e insieme anche per il bisogno di rappresentare la realtà anche se solamente nella dimensione piana, almeno per il momento. Insieme agli Egiziani anche i Babilonesi iniziarono a considerare la geometria come un sapere importante sempre per l'esigenza

di conoscere forme e misure per problemi quotidiani e reali. Colori i quali portarono ad un altro livello lo studio e la conoscenza geometrica furono i Greci con i suoi grandi matematici. Essi svincolarono lo studio della materia dai problemi di carattere unicamente pratico per uno studio più approfondito. Il grande passo che costituì un cambiamento radicale e che portò allo studio della geometria come lo conosciamo oggi fu saper tradurre problemi concreti in problemi geometrici puramente astratti, rendendo di fatto la trattazione universale. I Greci ebbero un ulteriore merito che costituisce un avanzamento nella conoscenza della geometria, cioè quello di risolvere i problemi e costruire le definizioni tramite il cosiddetto procedimento dimostrativo. I principali pensatori greci furono Talete e Pitagora (spetta a loro il primo studio dei poliedri regolari, fra questi anche il tetraedro, geometria principale del caso studio affrontato nel capitolo sesto).



*Figura 1: Una delle prime rappresentazioni del libro tredicesimo di Euclide dove vengono descritti i poliedri regolari pervenuto fino a noi è il manoscritto di Stephen the Clerk per Areta di Patraso, a Costantinopoli nell'888 d.C. Il manoscritto ora risiede nella Biblioteca Bodleian, Università di Oxford. In questa particolare pagine viene descritto il procedimento per la costruzione del tetraedro regolare inscritto in una circonferenza, elemento di rilievo nel caso studio presentato nel capitolo sesto.*

Poi Platone con i solidi platonici, Euclide, che articola i propri contributi in tredici libri pubblicati ad Alessandria tra il IV e il III sec. a.C., e ancora Archimede di Siracusa, Appollonio di Perga e numerosi altri geometri minori. Dopodiché vi sarà un lunghissimo intervallo di tempo (durato circa diciassette secoli) nel quale la trattazione geometrica rallenterà fino a fermarsi. Tale intervallo di tempo, però, non fu del tutto negativo: infatti tutto questo tempo fu necessario per assimilare certi concetti e rendere universale, diffusa e compresa tutta la conoscenza accumulata con i grandi pensatori precedenti.

## 2.2. La rappresentazione geometrica delle superfici

Come si è potuto vedere, il tema della rappresentazione geometrica ricopre un importantissimo snodo nella mente di un architetto/ingegnere e, come si potrà notare con i numerosi rimandi a questa materia, è molto significativo anche in questo lavoro. È giusto allora approfondire il tema dedicandogli un breve paragrafo.

La rappresentazione geometrica raccoglie intrinsecamente all'interno della sua definizione non solo tutti quei valori che sono di semplice visione, ma anche una serie di valori che non sono direttamente tangibili come valori culturali, teorici e storici. Come già detto, quindi, la geometria e la sua rappresentazione sono uno strumento imprescindibile per l'architetto/ingegnere per due principali motivi: il primo perché permette una corretta e soprattutto efficace visualizzazione di oggetti, anche tridimensionalmente molto complicati, nello spazio e il secondo motivo perché la rappresentazione geometrica è rilevante in quanto può ispirare la creatività riuscendo nell'intento di unire due discipline, l'arte e la scienza, proprie dell'architetto/ingegnere. Questo parallelismo tra le due discipline Scienza e Arte è emblematico perché dimostra lo strettissimo collegamento che è presente anche tra Rappresentazione e Geometria e tra la figura dell'architetto e dell'ingegnere.

La "scientificizzazione", se così si può definire, per una giusta rappresentazione della geometria descrittiva, prima dei lavori di Gaspard Monge (1746-1818), ha inizio solo durante il Rinascimento. Infatti proprio durante questa epoca nascono i primi studi della prospettiva e l'esigenza di una giusta rappresentazione di essa. L'applicazione del sapere matematico in ambito figurativo avviene da parte di due forti personalità: Girard Desargues (1591-1661) e Blaise Pascal (1623-1662). Il primo matematico che si occupò però in maniera approfondita delle superfici fu Pierre de Fermat (1601-1665), il quale, oltre a definire alcuni principi generali interessanti, si soffermò sulla costruzione delle superfici coniche. Dopo questi pionieri matematici, si assistette all'epocale ascesa di Gaspard Monge, una figura di spicco che lasciò un'impronta indelebile nello sviluppo della matematica e, in particolare, nella rappresentazione della geometria. Monge si distinse per il suo eccezionale contributo alla geometria descrittiva, una disciplina di cui egli stesso pose le basi. Monge nel suo principale scritto *Géométrie descriptive* ha lasciato un'eredità duratura nel campo della geometria descrittiva, poiché ha sviluppato il metodo fondamentale per la rappresentazione di punti e altri elementi dello spazio attraverso la doppia proiezione ortogonale. In particolare si occupò della rappresentazione di una superficie variabile sia di posizione che di forma, rappresentandola tramite un infinito sistema di linee in essa contenute. Questo innovativo approccio di Monge ha rivoluzionato la capacità di visualizzare e comprendere le forme tridimensionali, offrendo un sistema coerente e preciso per tradurre concetti spaziali complessi in rappresentazioni bidimensionali. Il metodo delle proiezioni ortogonali di Monge è diventato uno strumento indispensabile in diverse discipline, dalla progettazione ingegneristica all'architettura, consentendo una comunicazione visiva chiara e accurata di concetti geometrici. Durante l'Ottocento un'altra personalità di rilievo negli studi sulla rappresentazione geometrica fu Henry Jean Victor Poncelet (1788-1867), il fondatore della geometria proiettiva, che porterà al metodo delle proiezioni centrali. Anche in Italia ci furono grandi pensatori che contribuirono agli studi della geometria descrittiva e alle superfici: alcuni di questi furono Vincenzo Flauti (1782-



1863), Giuseppe Tramontini (1768-1852), Antonio Maria Bordoni (1788-1860), Giuseppe Bruno (1828-1893) e Luigi Cremona (1830-1903).

Grazie a tutti questi pensatori è stato possibile liberare la prospettiva da tutte quelle precedenti limitazioni e con la nuova metodologia essa diventa uno studio matematico con un potenziale enorme di rappresentare verosimilmente figure nello spazio tridimensionale. I metodi creati sono anche ora adottati per la rappresentazione di oggetti nello spazio, soprattutto nella progettazione architettonica, dando una base scientifica alla rappresentazione geometrica capace di ricreare la realtà.

Negli ultimi due secoli altri fattori hanno sviluppato la prassi architettonica e ingegneristica: la nascita di nuove geometrie non euclidee e l'avvento del computer. In particolare l'uso dei software di progettazione permette di applicare e utilizzare tutti i dati necessari in modo immediatamente disponibile, ma col rischio di procedere per automatismi e di perdere, almeno in parte, il controllo di quanto si sta operando. D'altro canto proprio i software di progettazione utilizzano quelle conoscenze geometriche come le proiezioni ortogonali, l'assonometria, ecc., che devono rimanere un patrimonio di sapere per il progettista.

### 2.3. La genesi configurativa

La genesi configurativa delle superfici risulta essere in architettura necessaria in quanto evita tutti i possibili fraintendimenti dello studio analitico geometrico. Conoscere la genesi configurativa di una superficie inoltre ci permette di costruire una rigorosa classificazione di superfici più o meno complesse, sempre orientata per una giusta rappresentazione. Prima di esporre i due possibili modi per classificare una superficie, va indicata preliminarmente la definizione più recente e appropriata di superficie: *una superficie è il luogo di tutte le posizioni, che assume successivamente nello spazio una linea mobile, che cambia di posizione, secondo una legge*

*determinata e continua*. Per *legge determinata* si intende una legge matematica la quale riesce ad individuare la *linea mobile*, detta *generatrice*, in una qualsiasi posizione che può compiere durante il moto. La generatrice poi dovrà appoggiarsi su uno o più linee fisse, che vengono dette *direttrici*; tutte le superfici apparterranno ad uno stesso gruppo, detto *famiglia*, quando si modificano solo le direttrici mantenendo immutate le generatrici. Quindi, fatta questa breve introduzione, le due classificazioni che possono essere effettuate sono:

- rispetto alla natura della generatrice: in questo caso le superfici sono dette *superfici rigate* se la generatrice è una linea retta, mentre *non rigate* se la generatrice, al contrario, è una linea curva.
- rispetto alla legge del movimento di essa: in questo caso sono dette *superfici di rotazione*, cioè superfici che si creano da una generatrice che ruota, trasla (*superficie di traslazione*) o, insieme, sia ruota che trasla (*superficie di rototraslazione*) ad un asse rettilineo.

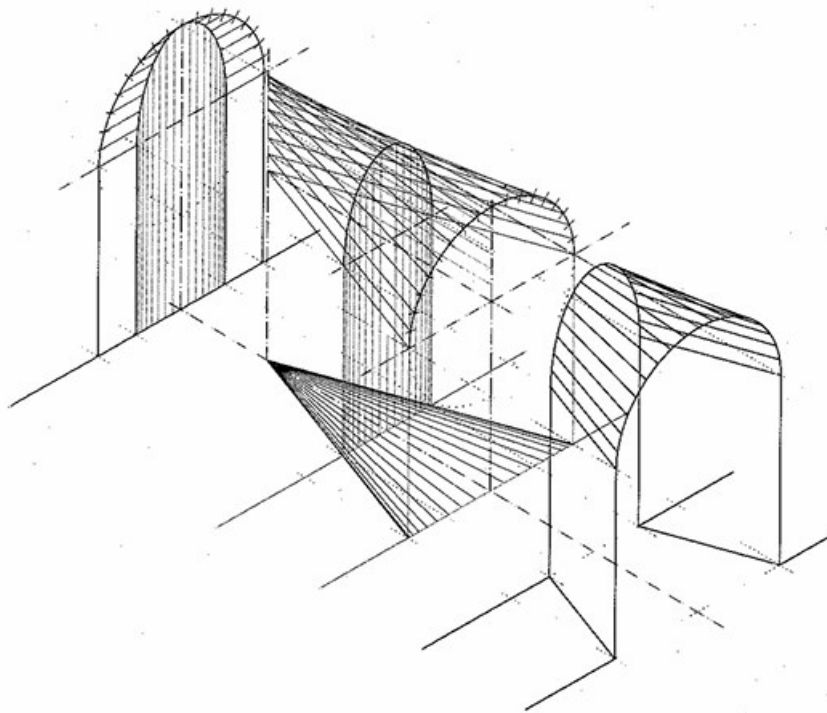
È anche vero però che una stessa superficie può essere classificata in più modi cambiando semplicemente o la generatrice o la legge del movimento.

Le superfici rigate, come detto, sono caratterizzate da una generatrice che è una linea retta; esse possono essere suddivise a loro volta in superfici sviluppabili e superfici gobbe (o sghembe o non sviluppabili). Ora senza entrare nello specifico delle definizioni di queste sottocategorie e degli sviluppi di queste tipologie di superfici non utili al fine di questo lavoro, vengono esposti gli esempi di superfici rigate:

- Coni e cilindri
- Iperboloide iperbolico
- Paraboloidi iperbolici
- Cono gobbo
- Corno di vacca (volta a sbieco)
- Conoide
- Cilindroide

- Elicoide
- Poliedri

Tra gli esempi citati, si affronterà solo uno a titolo esemplificativo, poiché la spiegazione dettagliata di tutti risulterebbe eccessivamente lunga e superflua: il conoide. Esso è una superficie gobba creata grazie ad una generatrice rettilinea la quale tocca sia una direttrice rettilinea che curvilinea, e rimane sempre parallela ad un piano direttore. Può essere sia conoide retto che conoide obliquo a seconda che la direttrice rettilinea è perpendicolare o obliqua al piano direttore. Il conoide è l'intradosso di una volta che copre uno spazio trapezoidale o la superficie dell'arco di ingresso in una torre cilindrica.



*Figura 2: Genesi configurativa di un conoide (da A. Giordano, Cupole volte e altre superfici. La genesi e la forma, UTET, Torino 1999).*

Le superfici di rotazione sono figure geometriche generate dalla rotazione di una curva o di una forma intorno a un asse. Questo tipo di superfici ha la caratteristica di mantenere la stessa forma mentre ruotano attorno a un punto

fisso, creando così una forma tridimensionale. Alcuni esempi di superfici di rotazione sono:

- Superficie cilindrica di rotazione, superficie conica di rotazione, iperboloide ad una falda di rotazione
- Ellissoide di rotazione
- Iperboloide di rotazione
- Paraboloide di rotazione
- Toro
- Qualsiasi altra superficie ottenuta dalla rotazione attorno ad un asse di una curva piana o sghemba

# 03 • Fondamenti dell'ottimizzazione

L'ottimizzazione è una disciplina che interessa una vasta gamma di settori, dall'ambito matematico a quello economico, ma in particolar modo, soprattutto negli ultimi decenni, ha occupato un ruolo cruciale nell'ambito dell'ingegneria. Anche nel campo ingegneristico, quindi, si utilizzano tecniche di ottimizzazione in una vasta gamma di rami: per esempio, per quanto riguarda la gestione delle risorse nell'ingegneria gestionale, in diversi settori produttivi come l'*automotive* e l'aerospaziale per migliorare l'efficienza meccanica dei componenti, per i processi produttivi nell'ingegneria meccanica, e, anche, per la progettazione di strutture nell'ingegneria edile e nell'architettura<sup>1</sup>. In sintesi, l'ottimizzazione riguarda la maggior parte delle scienze esatte e questo strumento si attua attraverso l'applicazione e la determinazione di modelli utili volti a rendere più efficaci soluzioni ottimali. Per una definizione più ampia possibile ma di facile lettura l'ottimizzazione è definita come un modello intuitivo che va a determinare processi decisionali complessi: avendo un ventaglio di scelte disponibili e un principio di valutazione si vuole ottenere la scelta che risulta più vantaggiosa rispetto al principio assegnato (C. Arbib, 2010). E ancora, è possibile definire l'ottimizzazione come la determinazione della soluzione che contemporaneamente soddisfa al meglio due parametri spesso discordanti fra loro. Alcuni esempi sono resilienza e agilità in un manufatto, elevata efficienza e ridotti consumi in un veicolo, prontezza di adattamento alle variazioni della domanda ed efficienza economica in un sistema produttivo di beni o servizi: complessivamente, costi contenuti e benefici adeguati. Come si vedrà più ampiamente nei paragrafi successivi l'ottimizzazione può essere facilmente

---

<sup>1</sup> I recenti progressi tecnologici e la ricerca architettonica contemporanea hanno tentato di fondere i due campi, coniugando forma e statica, creatività e calcolo strutturale in modo che è sempre meno chiara la distinzione fra i ruoli dell'ingegnere e dell'architetto.

associata al volere e all'interesse di un singolo individuo. Infatti la teoria dell'ottimizzazione non si discosta dal concetto di decisione, ovvero di espressione per una preferenza consapevole o non consapevole (C. Arbib, 2010). L'ottimizzazione si configura come uno strumento indispensabile che fornisce un approccio sistematico e scientifico<sup>2</sup> per massimizzare l'efficienza e minimizzare gli sprechi in molteplici contesti, migliorare le prestazioni dei sistemi e promuovere lo sviluppo di soluzioni valide e sostenibili. Nello specifico, l'ottimizzazione nel campo della progettazione civile e dell'architettura ci consente di comprendere, spiegare e, in definitiva, creare configurazioni spaziali ottimali che riducono il numero di cicli di progettazione e l'utilizzo delle risorse, oltre a limitare gli sprechi; tutto ciò può portare ad un nuovo modo di pensare la progettazione.

### 3.1. Introduzione e concetti di base dell'ottimizzazione

Già a partire dal XVIII secolo furono studiati metodi per massimizzare l'ottimizzazione di strutture e componenti nel campo dell'ingegneria attraverso l'utilizzo di strumenti analitici. Questo obiettivo però non è stato raggiunto fino agli anni '50 e '60 del XX secolo data la complessità dei calcoli da studiare (A. Tedeschi, 2016). A partire da questi anni, infatti, con l'introduzione del computer c'è stata una rivoluzione per l'analisi ingegneristica<sup>3</sup>. Questa

---

<sup>2</sup> Per ciò che si è detto sino ad ora si potrebbe pensare che la pratica dell'ottimizzazione possa avere una sua utilità e interesse solamente in relazione ad argomenti matematici o comunque ad argomenti che si possono in qualche modo numerare e quantificare. È possibile però dire che numeri e più in generale tutte le materie scientifiche sono solo uno "strumento". Infatti si può pensare alla ricerca delle soluzioni ottimali grazie agli insiemi all'interno dei quali vengono stabilite regole per mettere in relazione e in ordine anche oggetti o decisioni non rappresentabili tramite numeri (C. F. Manara e G. Lucchini, 2013).

<sup>3</sup> Nonostante questo, il termine ottimizzazione è stato accolto solo recentemente nei dizionari: ad esempio nel Dizionario Enciclopedico Italiano compare con il supplemento del 1974. Dal Dizionario Enciclopedico Italiano - Istituto della Enciclopedia Italiana. Roma – vol. VII – 1970:

innovazione ha reso possibile convertire complessi sistemi di equazioni differenziali in problemi algebrici risolvibili mediante calcolatori. Questa trasformazione ha aperto la strada a un approccio pratico ed operativo alla risoluzione di problemi di ottimizzazione, inizialmente espressi in maniera teorica.

Analizzando in maniera attenta la vita di tutti i giorni l'idea di ottimizzazione non è poi così difficile da riscontrare nel comportamento umano in quanto ottimizzare è una tecnica molto vantaggiosa per portare a compimento moltissime azioni in maniera più veloce, attenta, fruttuosa, produttiva, in sintesi: efficace. Un esempio molto chiaro è, per recarsi sul posto di lavoro, cercare la strada meno trafficata. Difatti questa può essere considerata un'azione ottimizzata, anche se semplice, in quanto si sceglie la via migliore su diverse possibili opzioni. In definitiva, quando si desidera ridurre al minimo i costi o massimizzare i profitti, si utilizza consapevolmente o inconsapevolmente un metodo di ottimizzazione. Arrivati a questo punto si può dire in maniera del tutto generale che con la parola ottimizzazione si definisce il raggiungimento di una posizione di ottimo, ossia del massimo

---

*ottimizzare* v. tr. [der. di ottimo]. – Rendere ottimo, portare a una condizione o a un risultato che siano considerati i migliori possibili. In particolare nella matematica e nelle sue applicazioni con riferimento a un problema le cui soluzioni dipendano da uno o più parametri o funzioni variabili, scegliere questi in modo che la corrispondente soluzione sia la migliore in relazione a un determinato fine (v. ottimizzazione in Suppl.).

*ottimizzazione* s. f. [der. di ottimizzare]. – Il raggiungimento di una situazione ottima, ossia del migliore risultato possibile nelle condizioni date o in relazione a un determinato fine. Il termine è utilizzato spec. nella matematica (v. oltre) e, in genere, nel linguaggio scientifico e tecnico. Matematica. – Nella matematica applicata sono chiamati problemi di o. i problemi consistenti nella ricerca del massimo e del minimo di funzioni o più in generale di funzionali, liberi oppure vincolati. La ricerca delle tecniche più adatte alla soluzione di problemi di o. ha ricevuto un grande impulso negli ultimi anni, con l'avvento dei calcolatori elettronici, che permettono di risolvere numericamente, in modo esatto o approssimato, problemi che prima, per la loro complessità, o per il numero delle variabili e delle funzioni implicate, si sottraevano a ogni trattazione.

risultato possibile con i termini dati o in relazione a un determinato fine (C. Arbib, 2010).

I problemi di ottimizzazione possono essere schematizzati all'interno di quattro fasi principali per la realizzazione del processo di ottimizzazione che all'interno presentano tre elementi fondamentali (A. Tedeshi, 2016):

- I. Prima fase: Definizione del problema. Identificare chiaramente il problema da risolvere e gli obiettivi da ottimizzare è il punto di partenza cruciale per la formulazione di un modello di ottimizzazione. Questo coinvolge un'analisi approfondita della situazione o del processo in questione al fine di comprendere appieno le sfide e gli obiettivi desiderati.
- II. Seconda fase: Identificazione delle variabili decisionali: determinare le variabili che possono essere regolate per ottenere il risultato ottimale. Vengono definite le condizioni al contorno che controllano l'ottimizzazione.  
In questa fase è riscontrabile il primo elemento che è l'insieme di variabili, ovvero la quantità i cui valori possono essere manipolati per ottimizzare l'obiettivo. Un esempio è il percorso che un veicolo deve seguire attraverso una rete di traffico: tutte le possibili scelte sono le variabili decisionali.
- III. Terza fase: Definizione di una o più funzione obiettivo: creare una funzione matematica che rappresenti l'obiettivo da massimizzare o minimizzare in relazione alle variabili decisionali.  
Il secondo elemento fondamentale, dunque, è la funzione obiettivo (o funzione di fitness o, ancora, funzione dell'ottimo) che deve essere massimizzata o minimizzata. L'obiettivo può essere il tempo di arrivo di un veicolo a una destinazione specifica.
- IV. Quarta fase: Formulazione dei vincoli: stabilire le restrizioni o i vincoli che limitano le soluzioni accettabili per il problema.



Il terzo elemento di un problema di ottimizzazione è un insieme di vincoli, che sono restrizioni sui valori che le variabili possono assumere. Ad esempio, per un veicolo i possibili vincoli di arrivare a destinazione sono quelli di percorrere solamente strade circolabili evitando tutte quelle al senso di marcia opposto o strade pedonali.

### 3.2. I modelli dell'ottimizzazione

In generale, un modello rappresenta un'astrazione selettiva di un sistema reale. In altri termini, un modello viene progettato per analizzare e comprendere da un punto di vista astratto il funzionamento di un sistema concreto, del quale contiene solo gli elementi ritenuti rilevanti ai fini dell'indagine svolta. Sulla base delle loro caratteristiche è possibile suddividerli in iconici, analogici e simbolici. Inoltre un'altra possibile suddivisione è quella che riguarda la natura aleatoria dei modelli in stocastici e deterministici. In modo del tutto comune i modelli di ottimizzazione vengono così definiti (C. Vercellis, 2008):

dato un insieme ammissibile  $S$ , contenente un numero finito o infinito di soluzioni ammissibili  $s \in S$ , e una funzione obiettivo  $f: S \rightarrow R$  che associa a ogni  $s \in S$  un numero reale  $f(s)$ , il problema di ottimizzazione associato alla coppia  $(S, f)$  richiede di determinare la soluzione ammissibile che rende minima la funzione obiettivo, ed è espresso simbolicamente come:

$$\min f(s).$$

La funzione  $f(s)$  rappresenta la funzione obiettivo,  $S$  è l'insieme delle soluzioni possibili (denominato ammissibile) e  $s$  è il vettore delle variabili di progetto. Generalmente, il vettore  $s$  è una variabile vettoriale  $n$ -dimensionale, esprimibile come  $s = (s_1, s_2, \dots, s_n)$ , e quindi la funzione obiettivo  $f(s)$  risulterà

una funzione di  $n$  variabili reali  $s = (s_1, s_2, \dots, s_n)$ . Gli elementi  $s^*$  e  $S$  che risolvono il problema, ovvero che minimizzano vengono detti *soluzioni ottimali globali*, o anche *soluzioni ottimali* o *soluzioni di minimo*. Il corrispondente valore della soluzione obiettivo  $f^* = f(s^*)$  viene detto *valore ottimo globale* o anche *ottimo* o *minimo*. Bisogna sottolineare però come non sempre è possibile una soluzione ottimale per il problema o dove esiste non è detto che sia unica. Ci sono casi in cui la funzione obiettivo per il raggiungimento dell'obiettivo desiderato, per esempio un guadagno monetario, invece di essere minimizzato deve essere massimizzato. In maniera del tutto analoga allora la relazione assume la forma:

$$\max f(s).$$

Ne consegue facilmente che il problema di ottimizzazione può essere ricondotto sia ad un problema di minimo che un problema di massimo cambiando semplicemente il segno della funzione senza che muti il risultato (C. Vercellis, 2008); infatti:

$$\max_{s \in S} f(s) = -\min_{s \in S} f(-f(s)).$$

Così facendo, la ricerca della soluzione ottima può essere sempre ricondotta alla ricerca del minimo della funzione obiettivo  $f(s)$ .

Per comprendere fino in fondo il concetto di ottimizzazione massima e minima si prosegue delineando due esempi concreti:

- Ottimizzazione Massima: Investimento in Azioni

Si immagina di avere una somma di denaro da investire nel mercato azionario, l'obiettivo sarà quello di massimizzare il profitto ottenuto dall'investimento. La funzione obiettivo in questo caso è il profitto che dipende

dalla scelta delle azioni e dal momento in cui si vendono, tale problema si risolve cercando di trovare la combinazione ottimale di azioni che rende massimo il profitto atteso.

- Ottimizzazione Minima: Riduzione dei Costi di Produzione

Si consideri una fabbrica che produce automobili, l'obiettivo da raggiungere dell'azienda sarà minimizzare i costi di produzione mantenendo una certa qualità dei veicoli. La funzione obiettivo in questo caso è il costo totale di produzione, che dipende dalle decisioni su materiali, processi di produzione, energia utilizzata, ecc. tale problema si risolve trovando la combinazione di risorse e processi che permette di produrre le automobili al costo più basso possibile senza compromettere la qualità.

Il concetto di equivalenza tra massimizzazione e minimizzazione illustra come i problemi di ottimizzazione possano riguardare sia la massimizzazione che la minimizzazione di una funzione obiettivo. Tuttavia, è possibile convertire un problema di massimizzazione in uno di minimizzazione e viceversa semplicemente cambiando il segno della funzione obiettivo.

Accanto alla nozione di ottimo globale è opportuno definire il concetto più debole di ottimo locale. Supponiamo che l'insieme ammissibile  $S$  sia dotato di un sistema di intorni  $T$ . Gli elementi  $s^L \in S$  per cui vale la relazione:

$$f(s^L) \leq f(s) \quad \forall s \in T(s^L)$$

vengono detti *soluzioni ottimali locali*, o *soluzioni di minimo locale*, rispetto al sistema di intorni  $T$ . Il concetto di minimo locale è importante in quanto dimostra che localmente una soluzione esiste ed è minima, ma non è sempre detto sia l'unica. Infatti, leggendo la seguente funzione, abbiamo tre diverse soluzioni di minimo, ma solo una di queste è quella globale.

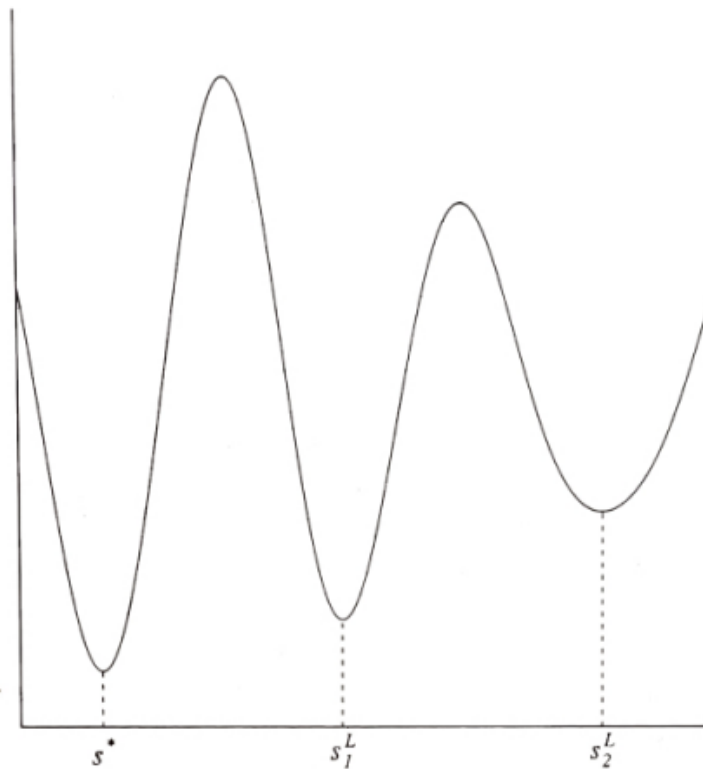


Figura 3: Minimi locali e minimo globale di una funzione (C. Vercellis, Ottimizzazione: teoria, metodi, applicazioni, McGraw-Hill, 2008, pag. 19).

Un'ulteriore considerazione da fare è che la funzione obiettivo  $f(s)$  da minimizzare, o massimizzare, è spesso soggetta a vincoli. I problemi vincolati mostrano un'applicabilità più elevata nella rappresentazione di processi decisionali complessi dal momento che i sistemi reali prevedono solitamente diversi vincoli necessari per rappresentare tutte le condizioni che definiscono l'ammissibilità del problema stesso. Un'ultima osservazione deve essere fatta per quanto riguarda la funzione obiettivo: infatti, a seconda della costruzione del problema, è possibile definire ulteriormente l'incognita di ottimizzazione lineare se la funzione obiettivo  $f(s)$  e le funzioni di vincolo sono lineari, viceversa viene definito di ottimizzazione non lineare se essi risultano non lineari. Questi ultimi, a loro volta, possono essere distinti in problemi di ottimizzazione intera, se le variabili del vettore  $s$  sono vincolate ad assumere valori interi, e di ottimizzazione convessa, se le funzioni risultano convesse.

### 3.2.1. Il modello matematico

Nella teoria dell'ottimizzazione occupano una posizione di rilievo i modelli matematici di ottimizzazione. Anticamente i problemi che noi oggi definiamo come problemi di ottimizzazione matematica erano chiamati modelli di programmazione matematica, ma questo termine potrebbe implicare un'errata associazione con i linguaggi di programmazione informatica (C. Vercellis, 2008). In effetti, storicamente le prime analisi della programmazione matematica possono essere ricondotte alle ricerche mirate alla costruzione di modelli matematici dell'economia a metà del XX secolo. Oggi si utilizza quasi esclusivamente il termine ottimizzazione per praticità e convenzione. Nei modelli di ottimizzazione matematica le funzioni per determinare l'ottimo e i vincoli che caratterizzano le soluzioni possibili per tali funzioni sono espressi attraverso equazioni e disequazioni matematiche. I modelli di ottimizzazione matematica, che hanno un grande potenziale applicativo grazie alla flessibilità delle loro formulazioni e all'efficacia dei loro metodi di soluzione, si presentano in diverse forme: tra queste ci sono l'ottimizzazione lineare, l'ottimizzazione intera e l'ottimizzazione non lineare. Come fatto nella sezione precedente, anche per il modello matematico si indicheranno le definizioni e le caratteristiche. Sempre da C. Vercellis, *Ottimizzazione: teoria, metodi, applicazioni*, McGraw-Hill, 2008:

si indichi come  $s = (s_1, s_2, \dots, s_n)$  il vettore delle variabili di decisione nello spazio  $n$ -dimensionale  $R$ , e come  $f(s)$  una funzione reale definita in  $R$ ,  $f : R^n \rightarrow R$ , che costituisce la funzione obiettivo. Sono inoltre assegnate le funzioni reali  $g_i(s)$ ,  $i = 1, 2, \dots, m$ , a loro volta definite in  $R$ ,  $g_i : R^n \rightarrow R$ , che costituiscono i vincoli. Un problema di ottimizzazione matematica è definito come:

$$\begin{aligned} & \min f(s) \\ \text{s.a } & g_i(s) \leq 0 \quad i = 1, 2, \dots, m. \end{aligned}$$

Per quanto riguarda il calcolo dei valori rispettivamente di ottimo globale e locale sempre Carla Vercellis ci dà una definizione la quale non si discosta di molto rispetto a quelle date per la risoluzione di un modello generico. Quindi, per il valore ottimo globale: una soluzione ammissibile  $s^* \in K$  viene detta soluzione ottimale globale se soddisfa la relazione:

$$f(s^*) \leq f(x) \quad \forall s \in K,$$

e il corrispondente valore  $f^* = f(s^*)$  viene denominato *valore ottimo globale*. Invece, dato un numero  $\delta > 0$ , i vettori  $x^L \in K$  per cui vale la relazione:

$$f(x^L) \leq f(x) \quad \forall s \in K : \|s - x^L\| < \delta$$

vengono detti *soluzioni ottimali locali*.

Come visto prima, a seconda delle diverse tipologie e caratteristiche della funzione obiettivo nonché delle variabili in gioco si possono ottenere differenti classi di problemi di ottimizzazione matematica. Di seguito le principali classi (C. Vercellis, 2008):

- Ottimizzazione lineare: il problema viene definito tale se la funzione obiettivo  $f(s)$  e le funzioni  $g_i(s)$ ,  $i = 1, 2, \dots, m$ , sono lineari.
- Ottimizzazione intera: questo problema, importante anch'esso per lo studio teorico e applicativo, si ottiene quando le variabile del vettore  $s$  possono essere solamente un numero intero.

- Ottimizzazione binaria o booleana: è un particolare problema di ottimizzazione intera con l'aggiunta che il vettore  $s$  può assumere solo valori 0 o 1.
- Ottimizzazione convessa: Il problema viene detto di ottimizzazione convessa se la funzione obiettivo  $f(s)$  e le funzioni  $g_i(s), i = 1, 2, \dots, m$ , sono convesse. Queste particolari tipologie di funzioni sono maggiormente interessanti in quanto hanno la caratteristica di poter ottenere l'ottimo globale di ogni soluzione ottimale locale. Vengono definite così: una funzione si dice convessa se il suo diagramma si trova al di sotto (o: al di sopra) della corda sottesa a due punti comunque scelti. Si osservi che una funzione lineare risulta simultaneamente convessa e concava.
- Ottimizzazione stocastica: se alcuni dei parametri nelle funzioni  $f(x)$  e  $g_i(s), i = 1, 2, \dots, m$ , rappresentano variabili casuali soggette a distribuzioni di probabilità.

### 3.3. Determinazione della soluzione ottimale: gli algoritmi iterativi

Per determinare la soluzione di un problema di ottimizzazione si utilizzano i cosiddetti algoritmi di ricerca i quali andranno rivolti agli opportuni modelli del problema di ottimizzazione. Come descritto nei paragrafi precedenti, in base all'impostazione del problema e quindi al numero e al tipo di variabili e vincoli l'ottimizzazione può essere di vario genere e in base al problema posto esistono altrettanti algoritmi risolutivi. Tra tutte le costruzioni degli algoritmi quella più preponderante e utilizzata è la ricerca della posizione di ottimo tramite schemi e processi iterativi. I processi iterativi si basano sulla generazione di una successione  $\{x_k\}$  di possibili soluzioni di modo che all'interno di questa sequenza ci sia una soluzione ottimale oppure più probabilmente che la soluzione converga ad una soluzione ottimale. Di conseguenza otterremo due diversi percorsi per raggiungere la soluzione; nel

primo caso la soluzione si ottiene quando l'algoritmo si ferma e trova tale soluzione ottimale; nel secondo caso, invece, l'algoritmo continua il calcolo finché non raggiunge l'approssimazione della soluzione ottimale ritenuta accettabile. Tale approccio mira a migliorare progressivamente un sistema o un processo attraverso la successiva iterazione di fasi di progettazione, implementazione e valutazione. Nel contesto dello sviluppo software, ad esempio, l'iterazione è essenziale per adeguare il prodotto alle esigenze emergenti e perfezionare le sue funzionalità. Ogni ciclo iterativo consente di ricevere feedback, identificare potenziali miglioramenti e apportare modifiche al sistema. Gli algoritmi di ottimizzazione vengono valutati in base alla loro efficienza computazionale, che rappresenta il numero di iterazioni necessarie per raggiungere una condizione di arresto in base alla dimensione del problema considerato. Inoltre, nel caso dei metodi di approssimazione, viene analizzato anche il grado di approssimazione al valore ottimo (C. Vercellis, 2008). Per la determinazione delle soluzioni ottimali esiste un'ulteriore importante tipologia di algoritmo che è la classe degli algoritmi genetici (ma si approfondirà questo tema nel capitolo quinto).

### 3.4. Ottimizzazione multi-obiettivo

Precedentemente si sono studiati e definiti in generale i modelli di ottimizzazione e la loro risoluzione tramite algoritmi iterativi dotati di un unico obiettivo tale da essere ottimo ed efficace se viene massimizzata una funzione di guadagno e minimizzata una funzione di costo. In altre parole, assumiamo implicitamente che si possa determinare un criterio unico per valutare l'efficacia delle decisioni prese. Nella realtà, però, l'assunzione di un unico criterio come funzione obiettivo è un po' restrittiva. Infatti è più comune che i processi decisionali siano complessi avendo così più possibilità di traguardo, o meglio, che tale traguardo sia l'insieme intricato di molteplici criteri. Ognuno di noi sperimenta ogni giorno come le decisioni siano il risultato di compromessi tra diversi criteri di valutazione: per fare un esempio,



decidiamo di acquistare un'autovettura se il suo prezzo rientra nel budget allocato, se il suo design è soddisfacente, se le prestazioni sono ritenute accettabili, se il confort è adeguato alle nostre aspettative. Questo esempio mostra come, per arrivare ad una decisione, si passa prima attraverso un processo decisionale che è caratterizzato da molteplici obiettivi e di conseguenza si può dire che una decisione risulta più o meno preferibile ad un'altra a seconda di un compromesso tra tutte le variabili in esame del problema.

È normale chiedersi a questo punto se sia possibile analizzare un problema di ottimizzazione multi-obiettivo tramite un calcolo di un modello mono-obiettivo effettuando una conversione. Allora il problema viene posto come segue: indichiamo come  $f(x)$  e  $g(x)$  due funzioni obiettivo distinte, supponendo che esse rappresentino indicatori di valutazione che si desidera minimizzare. Il problema di determinare una soluzione che minimizza sia la funzione  $f(x)$  sia la funzione  $g(x)$  è evidentemente mal posto sotto il profilo matematico, dal momento che possono esistere coppie di soluzioni  $x_1, x_2$  per le quali  $f(x_1) < f(x_2)$  e  $g(x_2) < g(x_1)$ . Si è mostrato quindi come non è possibile portare un problema multi-obiettivo ad uno mono-obiettivo in quanto non è possibile determinare quale tra  $x_1$  e  $x_2$  sia più conveniente.

Per ricondurre problemi con obiettivi molteplici a modelli di ottimizzazione a singolo obiettivo esistono comunque due metodi validi: l'ottimizzazione multi-obiettivo senza priorità e l'ottimizzazione multi-obiettivo con priorità. La prima consiste nell'attribuire alle decisioni per scegliere l'ottimizzazione efficace pesi diversi così da esprimere importanze diverse, similmente a quello che si fa per le medie ponderate. Per esempio, se considero due funzioni obiettivo, il problema può essere così formulato (C. Vercellis, 2008):

$$\min \theta_f f(x) + \theta_g g(x),$$

dove  $\theta_f$  e  $\theta_g$  rappresentano pesi relativi degli obiettivi  $f(x)$  e  $g(x)$ . È evidente che anche questa formulazione risulta critica in quanto non è quasi mai oggettivo il peso di una funzione obiettivo e risulta quasi impossibile tradurre gli indicatori di valutazione con misure idonee concettualmente. L'ottimizzazione multi-obiettivo con priorità, invece, meglio rappresenta la realtà, in quanto in questa formulazione la decisione viene definita da una gerarchia di priorità tra gli obiettivi considerati. Date sempre due funzioni obiettivo  $f(x)$  e  $g(x)$ , supponiamo che  $f(x)$  abbia una priorità maggiore rispetto a  $g(x)$ . Possiamo quindi formulare il problema di ottimizzazione a obiettivo singolo come (C. Vercellis, 2008):

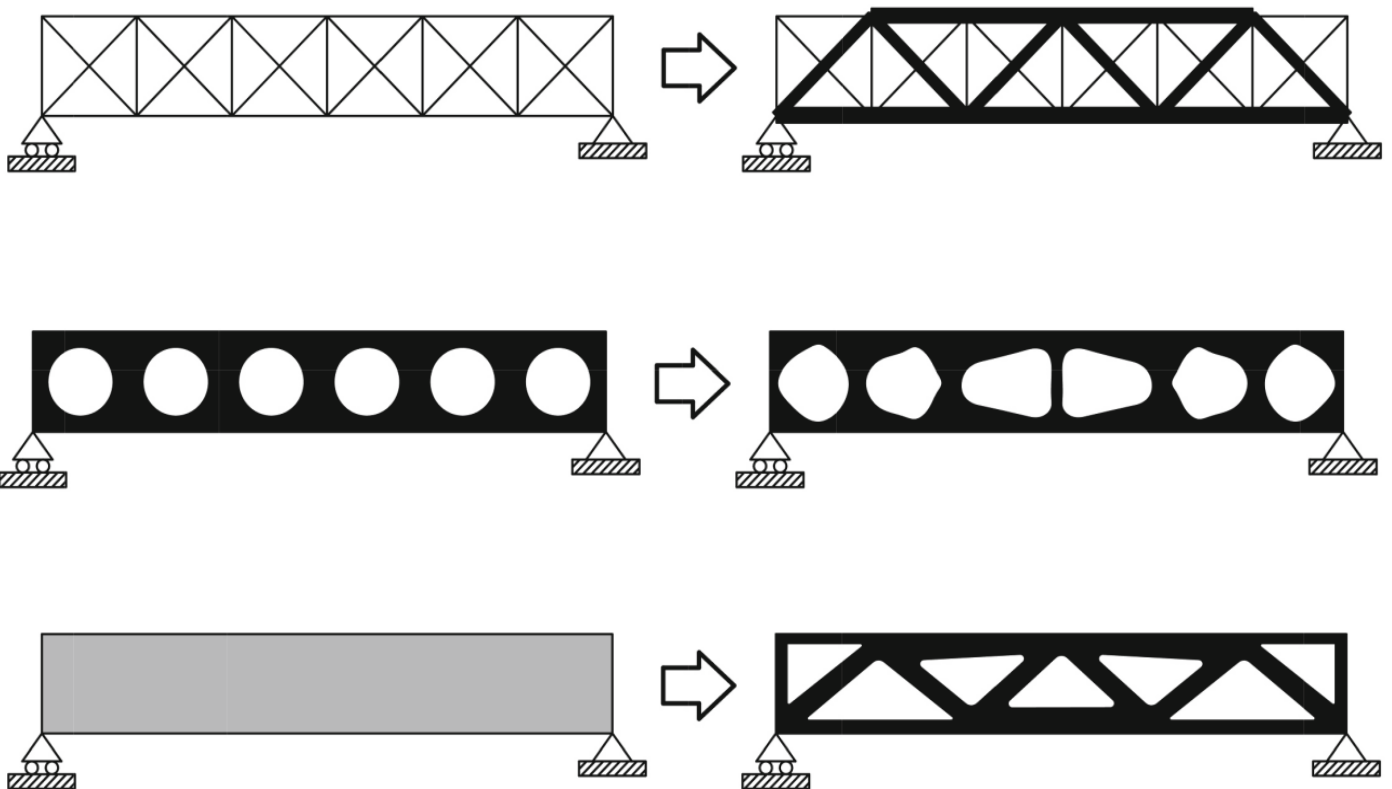
$$\min f(x) \quad H=\{x \in K : g(x) \leq G\}$$

dove il parametro numerico  $G$  rappresenta una soglia di valore ritenuta accettabile per l'obiettivo  $g(x)$ . Per questo problema possono verificarsi due situazioni: la prima è che il problema non possiede soluzioni ammissibili in quanto  $G$  risulta essere troppo contenitivo; la seconda è che il problema ammette soluzione cui corrisponde il valore ottimale  $z_f^*$ .

### 3.5. Diversi tipi di ottimizzazione

Esistono tre distinti paradigmi di ottimizzazione applicati nel contesto ingegneristico: l'ottimizzazione di forma, di dimensione e quella topologica. Tutte e tre queste tipologie di ottimizzazione ricercano e analizzano, come spiegato precedentemente con l'ottimizzazione in campo matematico, la ricerca della soluzione ottimale rispetto ai tre elementi fondamentali: le variabili, i vincoli e la funzione obiettivo. Questi tre differenti percorsi di affrontare l'ottimizzazione condividono la stessa procedura logica, ma sono intellettualmente e algebricamente differenti. In particolare nel campo

dell'ottimizzazione strutturale, essa si intende come la ricerca mirata a individuare la forma, le dimensioni e la tipologia ottimali di un componente specifico o di una struttura, con l'obiettivo di migliorarne la resistenza o la rigidità, nonché di ridurre il peso, la massima flessione o i costi di produzione. Ecco spiegato il perché lo studio di questa materia sia così rilevante. La ricerca e l'applicazione pratica, poi, di queste tecniche contribuiscono al progresso tecnologico e al miglioramento delle soluzioni esistenti (A., Giordano, P. Borin, et al., 2022). Le tre diverse tipologie di ottimizzazione si possono sintetizzare, prendendo in esempio una trave, nella seguente immagine in modo chiaro:



*Figura 4: Tre categorie di ottimizzazione strutturale. a) Ottimizzazione del dimensionamento, b) Ottimizzazione della forma e d) Ottimizzazione della topologia (A. Giordano, F. Panarotto, R. A. Bernardello, P. Borin, From Historical and Theoretical Analysis of Representation and Geometry to Topology for Structural Optimization, M.A. Ródenas-López, J. Calvo-López, M. Salcedo-Galera, Architectural Graphics, EGA 2022).*

### 3.5.1. Ottimizzazione dimensionale

Si può dire che l'ottimizzazione dimensionale sia la più semplice delle tre esistenti in quanto consiste nel solo modificare la geometria, in questo caso la dimensione, del componente strutturale. L'ottimizzazione dimensionale riguarda la creazione di una struttura ottimizzata partendo dalle dimensioni massime che potrebbe avere, aspetto noto come il suo spazio di esistenza (il suo ingombro). Questa fase coinvolge un processo iterativo in cui si rimuove il materiale in eccesso fino a che la struttura soddisfa specifiche condizioni vincolanti. In ambito strutturale, tali vincoli possono riguardare il volume, la rigidità, la frequenza naturale, la massima tensione ammissibile e altri parametri. Riprendendo l'esempio della trave visto in precedenza, nell'ottimizzazione delle dimensioni (o del dimensionamento) la variabile di progetto è la dimensione degli elementi strutturali, cioè l'area della sezione trasversale dei componenti della capriata. Quindi, date le informazioni di base iniziali quali il dominio, la forma e la geometria e le proprietà degli elementi come lo spessore, l'area e il materiale, l'ottimizzazione dimensionale riesce a definire quale sia il dimensionamento più efficace per la realizzazione della suddetta trave. L'ottimizzazione dimensionale entra in gioco quando la progettazione di una struttura è praticamente completa, rendendo difficile l'applicazione nelle fasi iniziali del processo progettuale. Tuttavia risulta estremamente vantaggiosa per rifinire ulteriormente un disegno già esistente. In sostanza, è una strategia per perfezionare la dimensione e le proporzioni della struttura ed è per questo che l'ottimizzazione dimensionale, seppur quella più semplice, avviene cronologicamente dopo quella di forma e quella topologica (A. Giordano, P. Borin et al., 2022).

### 3.5.2. Ottimizzazione di forma

Abbiamo visto che l'ottimizzazione in campo ingegneristico avveniva seppur in maniera più inconsapevole o con tecniche ancora troppo primitive nei secoli scorsi. Un chiaro esempio in tal senso è il lavoro dello scienziato R. Hooke (1635 – 1703) che durante la sua vita ebbe una parentesi da architetto. Nonostante non avesse avuto esperienze precedenti, Hooke riuscì a studiare un metodo anticipato dell'ottimizzazione di forma. Infatti, oltre a numerose nuove idee come nuovi dettagli costruttivi o nuovi strumenti per il rilevamento, Hooke dette importanti contributi alla scienza delle costruzioni: nel 1675 lo scienziato inglese scoprì la relazione di una catenaria: la proprietà di avere in ogni suo punto una distribuzione uniforme del suo peso totale e questo tipo di curva è stata spesso utilizzata per realizzare manufatti e strutture architettoniche<sup>4</sup>. Negli ultimi decenni con l'avvento di nuovi mezzi informatici questa tecnica è stata ampiamente sviluppata: alcuni esempi di cosa si è potuto realizzare verranno ampiamente descritti nel capitolo seguente.

Quindi in un problema di ottimizzazione di forma si vuole arrivare al raggiungimento della forma della struttura in esame migliore possibile seguendo e rispettando, come visto precedentemente nell'ottimizzazione dimensionale e come si vedrà nell'ottimizzazione topologica, i vari contorni, vincoli e variabili scelti. Pertanto, l'obiettivo è determinare la forma migliore del dominio sconosciuto, che a sua volta assume il valore della variabile di progetto. Possiamo quindi partire da una configurazione geometrica iniziale e raggiungere, grazie a processi che possono essere di aumentare o diminuire la forma degli elementi, una geometria

---

<sup>4</sup> Per esempio nella cupola di St Paul a Londra progettata da Robert Hooke, negli archi ideati per la prima volta con questa forma da Antoni Gaudí e nel Gateway Arch a Saint Louis progettato dall'architetto Eero Saarinen e dall'ingegnere Hannskarl Bandel.

ottimizzata. Bisogna evidenziare però che l'ottimizzazione di forma e anche quella di tipo dimensionale non possono consentire la creazione di nuove parti o, ancora, la generazione di fori sulla superficie della trave o di una qualsiasi geometria presa in considerazione. Si può, cioè, aspettarsi un cambiamento di forma o di dimensione degli elementi come possono essere il diametro di un foro, il raggio di curvatura o qualunque altra misura caratteristica del dominio. Nell'ottimizzazione della forma, la variabile di progetto rappresenta la forma o il contorno di una parte del confine del dominio strutturale. Essendo i parametri geometrici le variabili del sistema, in questo problema si trova la soluzione per metodi iterativi cioè quei metodi caratterizzati dalla ripetizione di una o più operazioni elementari fino al raggiungimento del limite prefissato di volte o all'arrivo dell'obiettivo descritto nel paragrafo precedente. Sia l'ottimizzazione dimensionale che quella di forma possono avere lo stesso percorso logico descritto dal seguente schema:

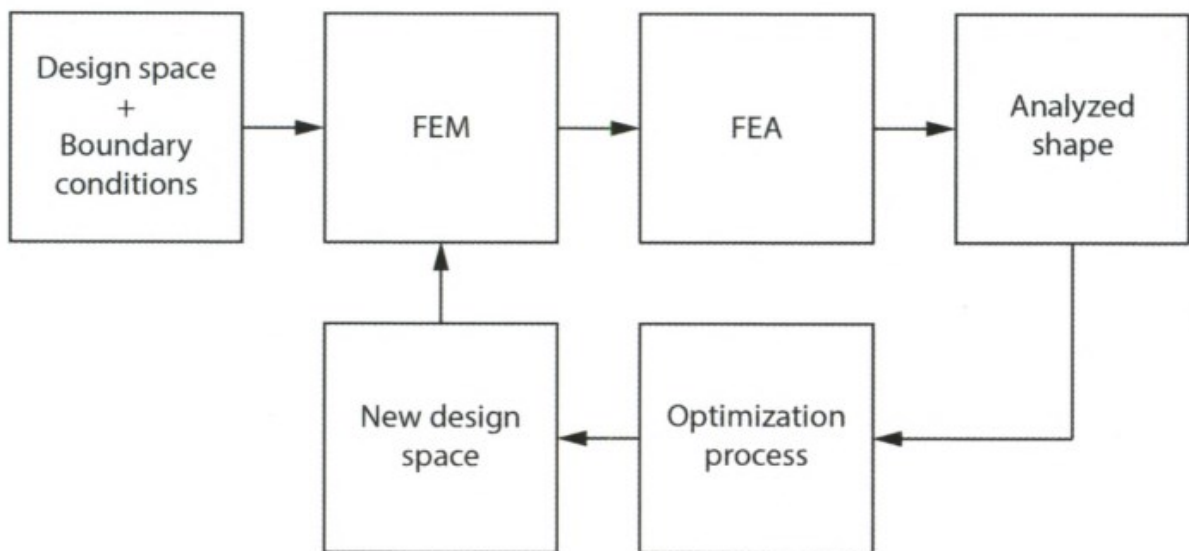


Figura 5: Diagramma logico dell'ottimizzazione di forma e dimensione (A. Tedeschi, S. Andreani, F. Wirz, AAD\_Algorithms-aided design: Parametric strategies using grasshopper, Le Penseur Publishe, 2016, pag. 406).

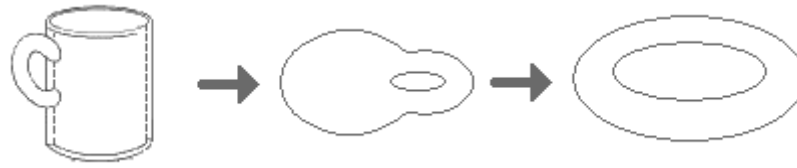
### 3.5.3. Ottimizzazione topologica

La topologia come concetto generale è una branca della matematica e della geometria che studia il rapporto tra la forma geometrica, che è ben nota a tutti, e le deformazioni che tale geometria subisce. Quindi, in generale, per la topologia non è assolutamente importante che forma e dimensione abbiano una geometria o quali siano le caratteristiche geometriche come i gradi degli angoli e le misure dei lati di una ipotetica geometria, in quanto non influiscono sulla topologia del sistema, ma importano solo la connessione e la continuità. Per esempio, se prendiamo in considerazione figure geometriche quali un quadrato, un cerchio e un'ellisse, esse in geometria euclidea sono tre figure ben distinte con le proprie caratteristiche, mentre nella topologia non c'è alcuna differenza perché tutte e tre le forme hanno curve chiuse semplici (connessione e continuità).

I due punti principali della topologia sono (A. Tedeschi, et al., 2016):

- I. Il principio di continuità dovuto alla deformazione; anche quando si perdono le caratteristiche metriche e proiettive, ci sarà una corrispondenza biunivoca tra i punti della figura originale e quelli della figura trasformata.
- II. Il principio dell'omomorfismo, che afferma che le deformazioni che causano penetrazioni o sovrapposizioni di materiale non sono ammissibili dopo la deformazione.

Seguendo i due principali concetti che stanno alla base della geometria topologica possiamo notare come, partendo da una tazza, si può arrivare ad un toro e come topologicamente i due solidi non cambiano:



*Figura 6: Una deformazione continua di una tazza di caffè in un toro. Le deformazioni continue vengono formalizzate nelle nozioni di omeomorfismo e omotopia.*

Per studiare l'ottimizzazione nel campo delle geometrie e delle forme si sono utilizzati i principi della topologia matematica. Come già visto in precedenza, il percorso per ricercare l'ottimo all'interno del campo dell'ottimizzazione iniziò nel XIX secolo, ma solo nel 1992, con il consolidamento dell'avvento di nuovi strumenti di calcolo, fu presentato il primo algoritmo realmente funzionante da Xie e Steven con il nome di Evolutionary Structural Optimization (ESO). L'implementazione della metodologia di ottimizzazione topologica in diverse discipline ingegneristiche si configura come una potente risorsa per notevoli miglioramenti, sia in termini di efficienza economica che di eccellenza nel design. Tali progressi rivestono un ruolo cruciale nella competizione globale, imponendo alle industrie contemporanee la necessità di adeguarsi a standard di costo più efficienti e di elevare la qualità progettuale. La soluzione che comporta l'ottimizzazione topologica è quella di poter arrivare alla migliore conformazione possibile di materiale all'interno di un dominio selezionato. In altre parole l'ottimizzazione topologica, definita anche in modo abbreviato OT, è la forma più generale e, quindi, anche quella più efficace delle ottimizzazioni descritte finora e solitamente avviene cronologicamente prima delle altre due. La sua potenzialità è quella di essere in grado di modificare e determinare, sempre sotto i rispettivi vincoli e variabili, posizione, numero e interconnessione dei vuoti e dei pieni al suo interno, non come avviene nelle ottimizzazioni di tipo formale e dimensionale dove, rispettivamente, vengono ottimizzate solo la forma e la dimensione mentre la configurazione spaziale di partenza risulta inalterata (S., Oh, et al., 2019).



Entrando un po' nello specifico, il calcolo matematico dell'OT con variabili di progetto continue si può approcciare in diverse maniere. I principali quattro approcci sono denominati: approccio delle densità, approccio dei derivati topologici, approccio a set di livello e approccio del campo di fase. Inoltre, per calcoli ancora più approfonditi, esistono anche ulteriori approcci come: gli approcci discreti e gli approcci lagrangiani e ottimizzazione combinata di forma e topologia. Non essendo l'obiettivo di questo lavoro lo studio delle varie tipologie di calcolo dell'OT, si andrà a discutere solamente l'approccio delle densità che risulta quello più comune. Bisogna fare un'ulteriore precisazione: il miglior metodo da utilizzare varia di caso in caso a seconda delle caratteristiche del problema e quindi c'è più di un modo di interpretazione o comprensione della soluzione. In altre parole, essa potrebbe avere diverse sfaccettature o significati, e la sua interpretazione potrebbe variare. Nella metodologia OT, quindi, la soluzione ottima che formerà il design ottimale è quella che meglio distribuisce la densità del materiale<sup>5</sup>  $\rho$  fittizio rispetto alla densità del materiale iniziale definito come  $\rho_0$ . Nel 1989 Bendsøe e successivamente altri suggerirono il SIMP (Simplified Isotropic Material with Penalization) che in un primo momento è stato inteso come un modo semplice ma artificiale di ridurre la complessità dell'approccio ricercando questa distribuzione di densità in un intervallo compreso tra 0 e 1, dove 0 è lo 0% di densità (vuoto) e 1 è il 100% di densità (totalmente pieno).

---

<sup>5</sup> È cruciale sottolineare che ci si riferisce alla densità strutturale, non al peso o alla massa dell'elemento, bensì alla sua capacità di conferire rigidità all'intero componente.

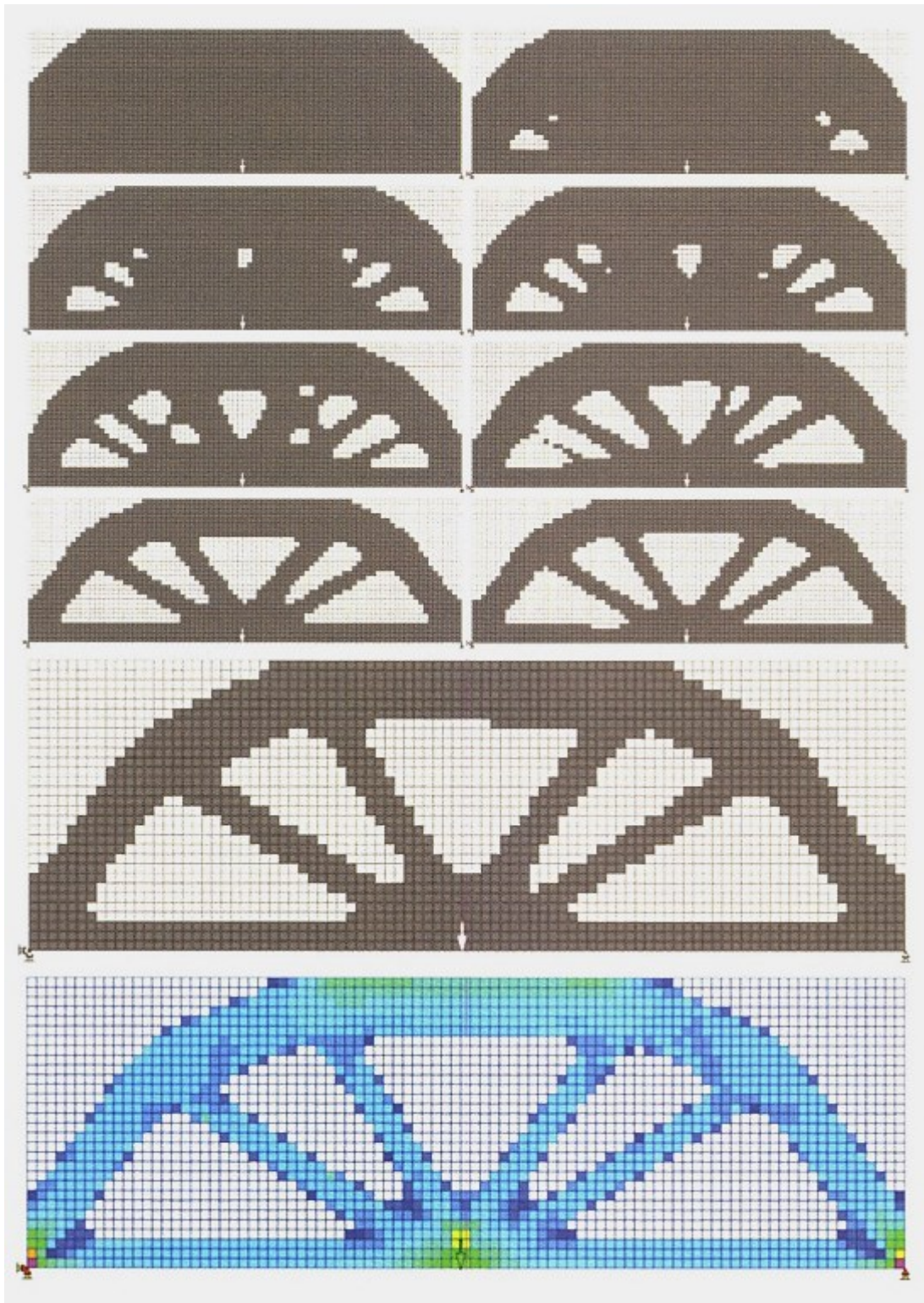


Figura 7: Ottimizzazione di una trave tramite topologia (A. Tedeschi, S. Andreani, F. Wirz, AAD\_Algorithms-aided design: Parametric strategies using grasshopper, *Le Penseur Publishe*, 2016, pag. 416).

Nell'approccio SIMP la relazione tra la variabile di progetto densità e la proprietà del materiale è data dalla legge:

$$\min C(\rho) = U^T f = U^T KU$$

$$\text{soggetto a : } \begin{cases} KU = f \\ V(\rho) < V_0 \\ 0 \leq \rho \leq 1 \end{cases}$$

In cui  $K$  è la matrice di rigidità globale,  $U$  e  $f$  rappresentano rispettivamente il vettore degli spostamenti e il vettore dei carichi esterni e  $\rho = [\rho_1, \rho_2, \dots, \rho_n]$  è il vettore delle variabili di progetto dato dalle densità dei materiali. L'espressione  $V(\rho) < V_0$  rappresenta il vincolo sulla quantità di materiale,  $V_0$  è il limite ammissibile (volume di progetto). Nel SIMP modificato, la densità che è direttamente associata al modulo di Young può essere espressa come:

$$E(\rho_i) = g(\rho_i)E_0 = \rho^p E_0, \quad g(\rho_i) = \rho^p$$

dove  $p$  è il parametro di penalizzazione ed  $E_0$  è il modulo di Young del materiale.

Per  $p = 1$  il problema di ottimizzazione corrisponde al cosiddetto problema del "variable-thickness-sheet" (Bendsøe e Sigmund, 1999), che in realtà, per l'obiettivo è un problema convesso con una soluzione unica. Tuttavia, per lo stesso obiettivo,  $p > 1$  penalizza gli spessori o le densità intermedie e quindi favorisce le soluzioni 0-1. La scelta di  $p$  troppo bassa o troppo alta provoca un'eccessiva scala di grigi o una convergenza troppo rapida verso i minimi locali; il "numero magico" che assicura una buona convergenza verso soluzioni quasi 0-1 è  $p = 3$ . Questo è stato successivamente confermato come il numero che assicura realizzabilità fisica di elementi

con densità intermedie (Bendsøe e Sigmund, 1999). È importante sottolineare che l'effetto di penalizzazione funziona solo in presenza di un vincolo di volume o di qualche vincolo che limita indirettamente il volume. Esistono numerosi altri schemi di interpolazione del materiale a una variabile che hanno tutti lo stesso scopo: fornire un'interpolazione continua tra il solido e il vuoto con penalizzazione dei valori intermedi di densità. Un'alternativa allo schema di penalizzazione implicita SIMP è l'utilizzo di una penalizzazione esplicita. Questo approccio prevede l'aggiunta del termine:

$$\alpha \int_{\Omega} \rho(1 - \rho)dV$$

alla funzione obiettivo (Allaire e Francfort, 1993; Allaire e Kohn, 1993), dove  $\alpha$  è un fattore di correzione. Per quanto visto finora, risulta comprensibile che i problemi di OT siano abbastanza complessi da risolvere numericamente ed è per questo motivo che ormai vengono impegnati i metodi iterativi e gli algoritmi genetici (i quali verranno ampiamente discussi nei capitoli successivi) per arrivare alla soluzione del problema. In queste metodologie di calcolo, infatti, vengono iterativamente rimosse porzioni di densità inefficace e passaggio dopo passaggio aggiornate le densità efficaci.

# 04 ■ Cambiamenti nella progettazione attraverso l'approccio del design parametrico

Come si è visto nel capitolo precedente, durante il processo di ottimizzazione diventa spesso importante, o comunque vantaggioso, riformulare il problema sotto l'aspetto parametrico, cioè impostare parametrizzando i set di variabili, la funzione obiettivo e i vincoli. A questo proposito è importante analizzare in profondità il concetto di parametro, come compare e cosa significa, e l'approccio all'architettura del design parametrico.

Gli architetti, ma più in generale anche i progettisti di un qualsivoglia lavoro, hanno avuto da sempre l'esigenza di disegnare ciò che progettavano in quanto, per condividere il proprio concetto con gli altri e offrire loro le conoscenze necessarie per trasformarlo in realtà, bisogna dapprima rappresentarlo. Quindi, il processo di concretizzazione di un'idea necessariamente implica la sua rappresentazione mediante immagini grafiche o mezzi simili. Tale rappresentazione, inoltre, non si limita solo a una comunicazione verso terzi, ma serve anche come strumento di controllo per il progettista stesso sul concetto iniziale. Nel corso del tempo tale rappresentazione si modifica in parallelo ai nuovi stili e ai nuovi studi. Per citare alcuni esempi, la prospettiva tramite Giotto prima e Brunelleschi poi, durante l'ultimo scorcio del Medioevo e l'epoca rinascimentale, e la geometria proiettiva nel Modernismo hanno rivoluzionato il concetto di rappresentazione e quindi il mondo del disegno. Il disegno che definiamo tradizionale ha però dei limiti, quello degli strumenti per poterlo sviluppare, che sono da secoli la carta, gli utensili da disegno, il righello e il compasso, che non permettono una libertà e un'efficienza ampia, e soprattutto costituiscono un processo additivo (A. Tedeschi et al., 2014). In questa tipologia

di processi non è possibile gestire le connessioni e le relazioni che possiedono i vari elementi. Questo porta a dire che il disegno tradizionale non è corretto a causa del mezzo, gli strumenti non sono rigorosi, ma bisogna affidarsi al progettista/disegnatore. In sintesi si può dire che il disegno non è un mezzo infallibile o intelligente perché si basa su standard e convenzioni dettati dall'uomo, altro elemento altrettanto non infallibile. Possiamo riassumere come il processo additivo per il disegno sia un metodo restrittivo in questi due punti:

- I. Il processo del disegnare si discosta di molto rispetto ai naturali meccanismi cognitivi del processo creativo dove si stabilizzano connessioni e non piuttosto si aggiungono informazioni.
- II. Il disegnare tradizionalmente non prevede l'aggiunta delle nozioni fisiche. Nel disegno ci sono dei grossi limiti per rappresentare alcuni importanti effetti fisici che invece sono rilevanti per la creazione di forme e geometrie.

Questi limiti e i problemi che implica il disegno tradizionale tramite processo additivo sono stati superati grazie all'avvento delle macchine di calcolo e dei primi computer. Tali limitazioni sono state superate, ma è bene specificare che non sono state risolte in quanto software tipo CAD<sup>6</sup> (computer-aided design) hanno migliorato il processo rendendolo più efficace e velocizzato, ma non hanno intaccato radicalmente il metodo di progettazione in modo logico. Dagli anni '80 ad oggi un costante lavoro di ricerche accademiche da parte di studiosi e programmatori ha cercato di sviscerare il mondo dei software digitali cercando sempre più di oltrepassare le limitazioni di editing delle applicazioni grazie alla costruzione di script, algoritmi e calcoli, quindi, in sostanza, manipolando e programmando in qualche modo il software da dentro. Questo processo evolutivo dei software e il fatto che ci si è resi conto delle potenzialità che possono avere certi programmi si possono notare anche grazie al ruolo del programmatore che

---

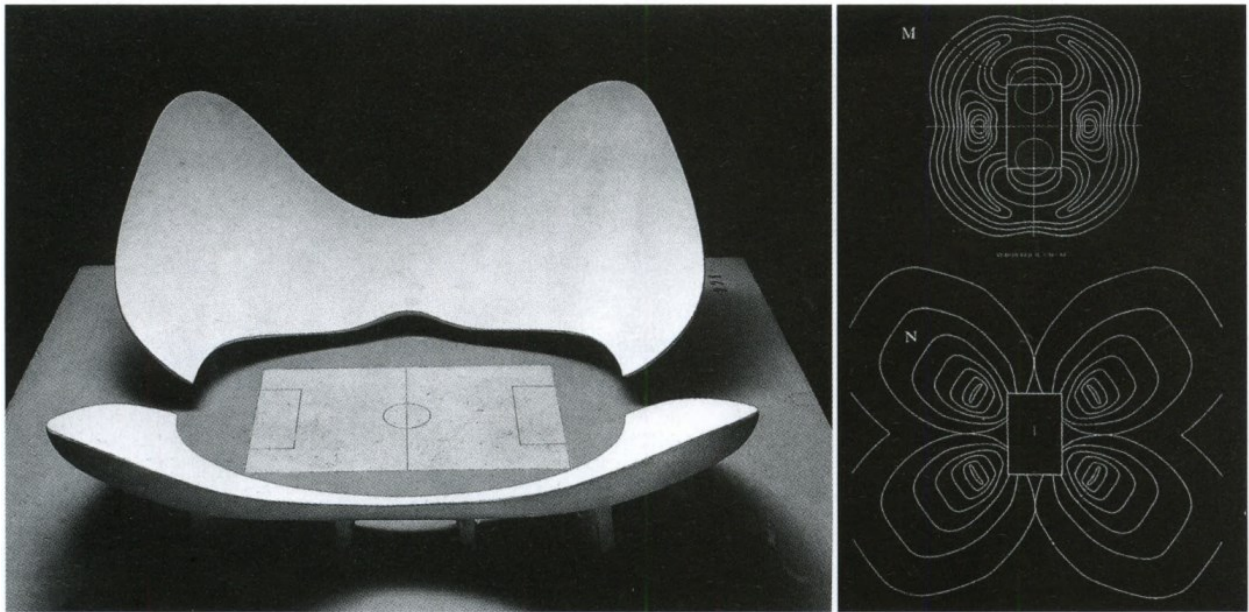
<sup>6</sup> L'informatico americano Ivan Sutherland sviluppò lo Sketchpad, definito come "A Machine Graphical di comunicazione", creando il primo programma interattivo di progettazione assistita da computer (CAD).

è sempre più determinante nella società moderna, contribuendo all'innovazione, all'efficienza e al progresso tecnologico. La domanda di competenze di programmazione è in costante aumento, e il lavoro dei programmatori ha un impatto tangibile sulla nostra vita quotidiana e sullo sviluppo della società. Tutto ciò ha portato a capire che si possono gestire in maniera migliore certe complessità di problemi un tempo difficilmente o spesso per nulla risolvibili grazie a questo approccio alla progettazione. Infatti negli ultimi anni è incrementato di molto lo sviluppo e l'interesse di forme molto particolari e di geometrie altrettanto complesse ed irregolari che verranno mostrate nel paragrafo "Storia e caratteristiche del design parametrico in architettura". Per queste tipologie specifiche, la complessità del comportamento strutturale sotto sistemi che esercitano forze esterne e la difficoltà di condurre analisi spesso limitano la progettazione stessa, portando i progettisti a valutare l'uso di sistemi di progettazione assistita parametrica che consentono di esplorare la differenziazione all'interno delle stesse soluzioni di concept design. Questa metodologia di modellazione si fonda su linguaggi di programmazione che formulano istruzioni in una forma eseguibile dal computer attraverso una sequenza di passi: ci si riferisce all'algoritmo, tema importantissimo per questo studio di tesi che verrà però affrontato nel capitolo quinto "L'utilizzo di nuove tecnologie per la progettazione parametrica".

Affrontando il tema dei cambiamenti nella progettazione attraverso l'approccio del design parametrico è impossibile non parlare dell'architetto italiano Luigi Moretti. Nel contesto della dodicesima Triennale di Milano del 1960, Moretti presentò idee e progetti che riflettevano le tendenze e le sfide architettoniche dell'epoca. La dodicesima Triennale di Milano era un importante evento espositivo che attraeva progettisti e artisti da tutto il mondo. In quell'anno, l'architettura moderna stava esplorando aspetti come l'uso innovativo dei materiali, la funzionalità e la ricerca di soluzioni estetiche contemporanee. Moretti, con il suo approccio eclettico, contribuì alla soluzione di queste problematiche con progetti che integravano elementi di tradizione e modernità.

La partecipazione di Moretti evidenziò la sua prospettiva distintiva sull'architettura e la sua capacità di unire concetti tradizionali con l'innovazione.

Moretti riconobbe subito le potenzialità del computer nel design. L'introduzione dei computer nel design da parte di Moretti e l'interfaccia grafica di Sutherland hanno segnato una rivoluzione nella tecnologia della progettazione architettonica e hanno migliorato gli strumenti dell'architetto (A. Tedeschi, et al., 2016).



*Figura 8: Ricerca "Architettura Parametrica". Esposizione alla Triennale di Milano, 1960. Soluzione per uno stadio di calcio e diagrammi (A. Tedeschi, S. Andreani, F. Wirz, AAD\_Algorithms-aided design: Parametric strategies using grasshopper, Le Penseur Publishe, 2016, pag. 20).*

#### 4.1. Definizione e concetti di base del parametro

Come evidenziato nella conclusione del paragrafo precedente, è evidente come ad oggi sia essenziale l'utilizzo di nuovi mezzi e strumenti per superare i limiti del disegno tradizionale. Lo strumento più innovativo è sicuramente il computer che viene utilizzato sia in fase di progettazione concettuale che nella fase di analisi tramite una metodologia parametrica, cioè capace di modificare il progetto nonché in grado di guardare tutta una serie di alternative in maniera simultanea. Questo metodo riesce in maniera semplice a mostrare



tutta la serie di possibilità che i progettisti possono ottenere dal disegno parametrico e velocemente poter visualizzare le correzioni. In sintesi si può dire che il progettista o disegnatore può manipolare, con una serie di sequenze logiche che verranno mostrate in seguito, il disegno a proprio piacimento senza sottostare alle limitazioni non solo della matita e della carta del disegno tradizionale, ma anche quelle dei software di tipo CAD. Questo approccio al disegno rappresenta una svolta molto interessante per il mondo del design e della pianificazione del prodotto perché permette di mettere in connessione una serie di processi sia logici che industriali intervenendo su uno specifico aspetto che si intende modificare ottimizzando il processo. Inoltre questo modello prevede un lavoro multidisciplinare in cui si fondono decisioni tecniche ed estetiche, il che fornisce una generazione flessibile e collettiva del progetto, implicando un diverso atteggiamento degli architetti nei suoi confronti, affidando compiti creativi ai processi e al team progettuale attraverso l'uso di vari sistemi digitali. Le tecniche parametriche nella progettazione architettonica stabiliscono nuove alternative allo sviluppo formale, consentendo un approccio chiaro e integrato a diversi aspetti, rappresentando così una capacità creativa con elaborazione tecnica, in cui il ruolo dell'architettura viene in primo piano, guidando soluzioni anziché generarle, e le condizioni di soluzione vengono definite e i risultati vengono selezionati. Il funzionamento dell'approccio parametrico si basa sulla relazione tra le varie parti del progetto: il progettista, infatti, costruisce il suo progetto a partire da queste relazioni e modifica queste relazioni in base alla valutazione e alla selezione dei risultati ottenuti. In questo modo si aumenta la possibilità di esaminare le varianti senza dover rifare ogni volta il lavoro di rappresentazione. Ciò richiede un cambiamento dal punto di vista logico del processo di rappresentazione e progettazione del disegno e del lavoro che si vuole svolgere, perché bisogna, ancora prima di iniziare, avere in mente la definizione delle relazioni, della logica che rende coerente il proprio progetto, come fase essenziale del processo di progettazione. In un modello parametrico ogni entità ha dei parametri associati. Questi parametri controllano le proprietà geometriche, come la lunghezza, la larghezza,

l'altezza, il raggio, ecc., controllano anche la posizione nel modello e il modo in cui le geometrie si relazionano tra loro l'una con l'altra. I parametri possono essere modificati di volta in volta dal progettista per creare le geometrie desiderate allontanandosi da quello che è il processo di progettazione tradizionale. In quest'ultimo si usa il metodo top-down, che significa avere una preparazione per ciò che deve essere fatto, cioè avere una forma prestabilita che viene sottoposta a una realtà o a un ordine. La progettazione parametrica, invece, utilizza il metodo bottom-up, che significa creare una logica basata su di relazioni specifiche, che possono essere modificate in base alle esigenze sociali e contestuali.

Entrando nello specifico del significato di parametro, nel campo del design di tipo parametrico, esso è quel processo logico che è interessato non tanto alla forma finale che la soluzione prenderà, ma piuttosto alle connessioni che si creano tra le parti, ovvero tra i dati, che corrispondono ai parametri scelti dall'operatore, e le operazioni matematiche, che corrispondono ai componenti. Decisi i parametri opportunamente e l'algoritmo matematico, cioè il mezzo che lega i parametri alle operazioni matematiche e definisce come devono interagire fra loro, si genererà, tramite un software, una forma, una geometria, un modello, ecc. Il risultato di questo processo sarà un modello tridimensionale che non è vincolato in modo definitivo, cioè potrà variare in un istante cambiando di volta in volta i parametri inseriti o alcune connessioni fra loro. Questa, come detto anche precedentemente, è la vera potenzialità: un modello geometrico 3D che è legato da relazioni logiche matematiche capace di aggiornarsi in qualsiasi momento al variare dei dati, rimanendo comunque coerente con le leggi stabilite all'inizio del lavoro. In questo modo è possibile arrivare alla soluzione che più si avvicina all'idea creativa capendo e controllando la risposta che ci è fornita dal software. Tutto ciò induce a fare due considerazioni principali: la prima è il fatto che la progettazione tramite parametri è un tipo di modellazione tridimensionale che si differenzia dalla modellazione tradizionale perché si basa sulla messa in relazione delle componenti, o parti del modello, tra di loro o con numeri o tramite caratteristiche che vengono appunto definiti parametri; la seconda è

che l'approccio alla costruzione del design cambia prospettiva. Infatti, in un approccio alla progettazione e al disegno che definiamo tradizionale si ha una soluzione a cui si vuole raggiungere e per farlo il processo è costituito da uno sviluppo dell'idea di base e dal risultato, mentre nel design parametrico la soluzione finale non esiste o, per meglio dire, ne esistono talmente tante generate dalla creatività che non ci si limita a sceglierne una prevista fin dal principio, ma si può scegliere la migliore data dal processo della progettazione. La soluzione nel design parametrico diventa così il percorso stesso per raggiungerla e il compito del progettista è quello di progettare solo le condizioni affinché tale risultato finale si possa generare, rappresentando qualcosa di prevedibile ma non di previsto. Questo permette di avere un processo che è "libero", nel quale i progettisti sono flessibili e possono tornare indietro nel percorso in qualunque momento rielaborando le regole per ottenere soluzioni e risultati aperti. È possibile ottenere molteplici soluzioni, se non infinite, che soddisfano comunque l'idea di partenza e il modello risponde ai cambiamenti adattandosi o riconfigurandosi ai nuovi valori assegnati ai parametri senza cancellarli o ridisegnarli. Non si parte più, quindi, dalle geometrie che vogliamo ottenere, ma dalle regole e dai processi logici che servono per arrivare a tali geometrie e forme, le quali non sono prescritte, ma rese deformabili fino alla fine del processo quando il progettista sceglierà, in base a tutto il percorso fatto, alle connessioni scelte e ai dati ottimali, la conformazione che più si adatta e che più piace tra tutte le possibilità.

È possibile, in sintesi, analizzare quali sono i punti che caratterizzano maggiormente questo diverso approccio alla progettazione: in primo luogo l'architetto o l'ideatore definisce quali sono le regole che sottostanno alla creazione del modello tridimensionale. Queste regole possono essere modificate in qualsiasi momento, trasformando anche il risultato finale, ma senza intaccare in nessun modo le relazioni che le legano. Infine è possibile sviluppare diverse forme di soluzione anche contemporaneamente al modello principale. Proprio a proposito di quest'ultimo punto va detto che il sistema parametrico consente, avendo ben conformato le regole del processo, di avere un numero illimitato di possibili alternative di design, le quali possono

essere generate in maniere rapida e veloce ma soprattutto in modo parallelo, il che significa che è possibile vedere, contemporaneamente alla modifica, le diverse soluzioni che vengono fornite anche in modo visivo e tridimensionale. Come è facile capire, il processo parametrico e lo sviluppo tecnologico vanno di pari passo ed è solo grazie all'innovazione tecnologica che è possibile effettuare questo tipo di approccio al design in quanto diventa possibile calcolare velocemente le soluzioni delle geometrie e le analisi visualizzando con estrema precisione e in tempo reale le diversificazioni effettuate sugli algoritmi (M. A. Schnabel, 2007).

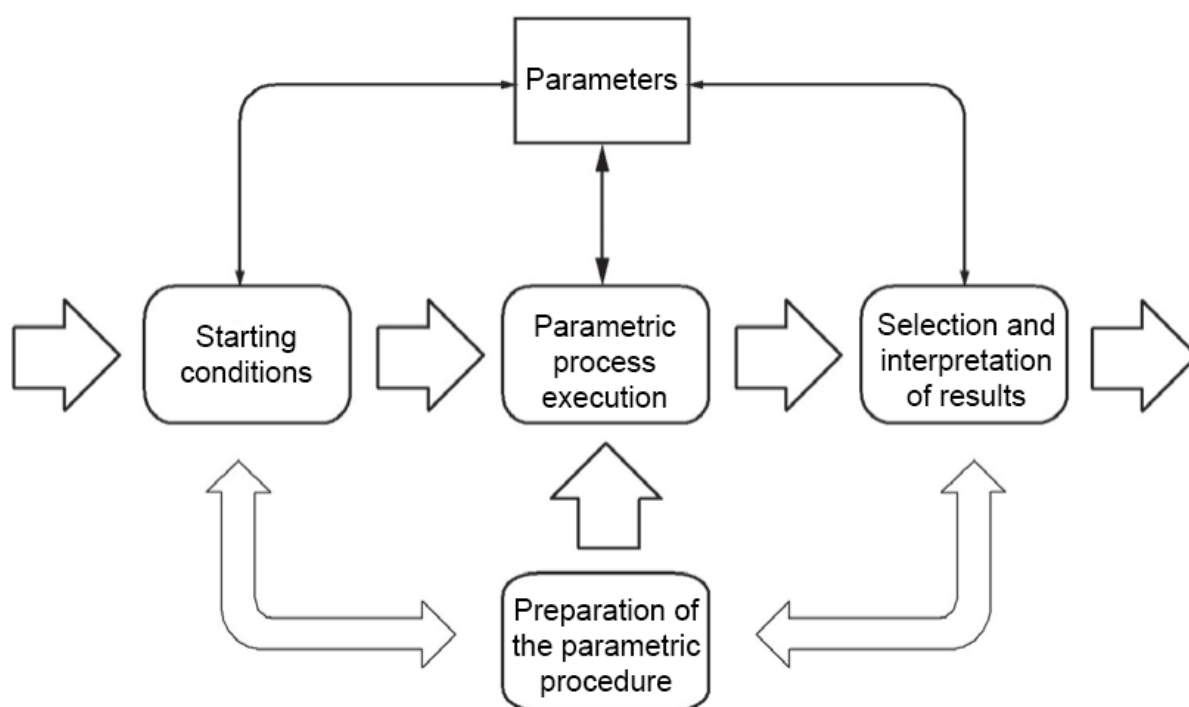
#### 4.1.1. Lo sviluppo del parametrico

Secondo alcuni studi importanti (Davis et al., 2011) tutto il percorso logico e pratico dell'approccio del design parametrico si può suddividere in quattro punti principali e la loro differenziazione contribuisce al loro sviluppo e alla loro generalizzazione. I quattro passi fondamentali si possono così distinguere:

- I. Definizione delle condizioni iniziali
- II. Preparazione del processo parametrico
- III. Esecuzione della procedura
- IV. Selezione e interpretazione dei risultati

Spesso tutte queste attività e questi dati sono rimescolabili fra loro nel senso che questo percorso è solamente logico ma non è fisso in quanto, per esempio, una volta ottenuto il risultato finale è possibile tornare indietro in una delle fasi precedenti per modificare alcuni dati per ottenere qualcosa di diverso e più vicino alle necessità del progettista. Inoltre un altro aspetto

fondamentale da considerare è quello dei parametri, i quali non rientrano nei quattro passi fondamentali perché toccano implicitamente tutti gli aspetti del processo. I parametri hanno diverse funzioni, quelle di esprimere intervalli, limiti e configurazioni specifiche. Lo stesso modello parametrico può fornire risultati diversi a seconda dei parametri che lo controllano.



*Figura 9: Svolgimento del design parametrico (R. G. Alvarado, A. Lyon, Diseño paramétrico en Arquitectura; método, técnicas y aplicaciones, ARQUISUR, 1, 2013).*

Per entrando nello specifico dei quattro passi della progettazione parametrica è opportuno analizzarli ora uno ad uno.

Le condizioni iniziali in un processo di progettazione parametrica sono i valori iniziali assegnati ai parametri del modello: questi valori influenzano la forma e le caratteristiche del design generato. Infatti, nella progettazione

parametrica, i parametri sono regole o vincoli che definiscono il comportamento del modello. Le condizioni iniziali possono essere variate anche in un momento già avanzato del processo, in quanto, come detto precedentemente, è possibile tornare sui “propri passi” scartando le opzioni che non raggiungono i requisiti o non corrispondono alle possibilità effettive. Le condizioni iniziali sono fondamentali perché determinano lo stato iniziale del sistema e influenzano il modo in cui il design si evolve in risposta alle modifiche dei parametri nel corso del processo. In sostanza, sono il punto di partenza che fornisce il contesto in cui il modello inizia a evolversi. Per questi motivi la scelta o la definizione accurata delle condizioni iniziali è essenziale per ottenere risultati desiderati e controllare il processo di progettazione nel suo insieme. Le condizioni iniziali possono essere di varia tipologia; alcuni esempi sono: scelte concettuali, limiti di prestazione, superfici funzionali, grandezze o caratteristiche della forma (curvature, estensioni, ripetizioni, ecc.). Inoltre possono essere espresse in operazioni o valori numerici in modo esplicito o intuitivo (ad esempio, quando si scelgono i risultati). L'importante è che le condizioni iniziali debbano essere definite tramite un sistema di relazioni geometriche.

Il secondo punto del processo è la preparazione del processo parametrico. Questa solitamente è un'azione matematica sequenziale a partire da dati numerici che genera un modello geometrico configurato come un flusso di dati e operazioni (un algoritmo). La preparazione del processo parametrico è essenziale e può essere definita il pilastro centrale di tutto il processo in quanto costruisce nella pratica ciò che conduce le condizioni iniziali e l'idea creativa di partenza ai risultati finali. A volte viene preparato appositamente e in altri casi si utilizzano procedure già esistenti che producono un repertorio di risultati o serie progressive (evolutive). Per fare questo passaggio è necessario servirsi di opportuni strumenti per la generazione di algoritmi o di scripting visuale. Alcuni di questi strumenti sono Grasshoppers, Generative Components o Digital Project, i quali però verranno approfonditi nel capitolo prossimo in maniera più dettagliata.

Pertanto la preparazione consiste spesso nella programmazione o nella ricerca di alcuni componenti o programmi completi, definendo lo sfondo e le azioni ed eseguendo alcune prove.

L'implementazione della procedura può essere chiaramente identificata come una fase operativa distintiva rispetto alla fase preparatoria, in quanto si traduce nell'esecuzione dell'algoritmo sui dati raccolti, generando risultati formali. Tale procedura è suscettibile di ripetizione, consentendo la manipolazione dei dati e la produzione di una varietà di risultati. In questo contesto, è possibile considerare questa fase, seppur sottolineando la sua sfumata distinzione, comprensiva della sua adattabilità in risposta ai dati e ai risultati generati. Tale flessibilità permette l'ottimizzazione della procedura stessa in base all'andamento delle iterazioni e dei risultati ottenuti, contribuendo così alla raffinazione continua del processo.

Infine durante l'ultimo passaggio di selezione e interpretazione dei risultati le procedure parametriche producono una soluzione o un insieme di soluzioni da integrare nel progetto architettonico. Difficilmente la prima forma che si deduce è la soluzione ottimale e quella definitiva, mentre corrisponde piuttosto ad un punto di partenza per tornare poi indietro nel processo, modificare l'algoritmo nella fase della preparazione del design parametrico e ottenere così un modello o una forma nuova più efficace: è proprio questa caratteristica, come detto precedentemente, la forza e la potenzialità di tale strumento. Questo processo richiede una valutazione critica delle opzioni disponibili, tenendo conto di parametri specifici come l'estetica, la funzionalità, il costo o altri riferimenti rilevanti. L'interpretazione dei risultati coinvolge la comprensione approfondita delle relazioni tra i parametri e le caratteristiche del design anche perché gli esiti del processo parametrico spesso generano una vasta gamma di soluzioni possibili, e interpretare correttamente queste soluzioni è fondamentale per ottenere un design ottimizzato. Diverse procedure generano sequenze in cui l'architetto può scegliere alcune forme applicando condizioni non previste all'inizio, potendo anche scartare le possibilità generate,

modificarle significativamente o realizzarne di diverse. La selezione e l'interpretazione dei risultati non sono processi statici; piuttosto, sono iterativi e possono richiedere regolazioni continue dei parametri in base alle esigenze emergenti durante il processo di progettazione. Per questo motivo la produzione della soluzione finale dovrebbe essere considerata come una effettiva attività di progettazione, che consiste nella selezione della forma e nel suo adattamento al progetto. Di solito questa soluzione, che può essere sotto forma di dati, di forme o di geometrie, deve essere trasferita a un sistema di progettazione per un'ulteriore elaborazione: per esempio, spesso si passa la soluzione, per la sua elaborazione, in una modellazione costruttiva (BIM) o in un sistema di progettazione tridimensionale (CAD 3D) e le figure possono essere incorporate nel modello completo (G. Alvarado, et al., 2013).

#### 4.2. Storia e caratteristiche del design parametrico in architettura

Il design parametrico in architettura può essere definito anche parametricismo ed esso si pone in contrapposizione alla figura del modernismo, movimento che si sviluppò a cavallo delle due guerre mondiali del secolo scorso. È possibile definire il parametricismo, infatti, come un ordine complesso e variegato, ispirato ai processi della natura piuttosto che a mettere insieme rigide ed ermetiche figure geometriche, come tutti i precedenti stili architettonici. In questo modo si può dire che il parametricismo avvicina componenti modellabili a proprio piacimento in un gioco dinamico di connessioni e di adattabilità al contesto. La principale fonte d'ispirazione nella realizzazione di figure e forme definite da parametri è individuabile nella natura. Difatti, il concetto di parametricismo si avvicina notevolmente all'architettura organica, ma si distingue per l'innovativa integrazione di software avanzati consentendo la creazione di forme notevolmente più complesse rispetto al passato. Questo approccio trova spazio in diverse



discipline, tra cui la progettazione urbana, l'architettura, il design degli interni, la moda e l'arte.

Come già scritto all'inizio di questo capitolo, nel 1940 il matematico Robert Stiles individuò come padre del design parametrico Luigi Moretti, i cui studi lo portano a dire che l'architettura parametrica ha come obiettivo quello di "definire le relazioni tra le dimensioni dipendenti dai vari parametri". Nel 1960 durante la dodicesima Triennale di Milano Moretti allestì una mostra che descrive in maniere pragmatica le funzionalità dell'architettura parametrica e dei valori del parametro tramite un progetto esemplare, che rimarrà tale anche poi. Il progetto era uno stadio dotato di ben diciannove parametri diversi i quali ottimizzavano la geometria per un uso e una forma efficaci in numerosissimi aspetti (si guardi la figura numero 8). Proprio in quegli anni Moretti ha annunciato l'arrivo di una nuova architettura che avrebbe dovuto esplorare ed espandere le considerazioni sul rapporto tra forma e struttura negli anni futuri. Tra il 1962 e il 1971 vennero costruiti gli edifici del Watergate, complesso che si può definire come uno dei primi esempi al mondo di edificio progettato tramite processi parametrici.

Il parametricismo rappresenta quindi uno spostamento nella prospettiva dell'architettura e del design, portando il concetto di invenzione fuori dai parametri tipici. Se il modernismo si orientava verso la funzionalità e l'essenzialità nello sviluppo di geometrie lineari e nella logica di una ragionevolezza visiva, allora il parametricismo si concentra sulla complessità e sull'energia della forma, con una libertà ben più sfrenata. Questa libertà è possibile grazie all'uso di algoritmi sofisticati e software di modellazione che consentono la manipolazione in tempo reale e in modo interattivo di qualsiasi forma complessa. Gli edifici possono essere progettati, simulando e ottimizzando una vasta gamma di condizioni ambientali e strutturali e influenze, in modo che non solo soddisfino i requisiti funzionali, ma si relazionino perfettamente sia al contesto naturale che urbano. In questa prospettiva, il parametricismo non è solo uno stile architettonico, ma un approccio filosofico e metodologico alla ricerca di nuove possibilità per l'interazione tra l'uomo, la tecnologia e l'ambiente, preparando uno scenario

in cui l'architettura sarà sempre più in grado di rispondere alla complessità del mondo.

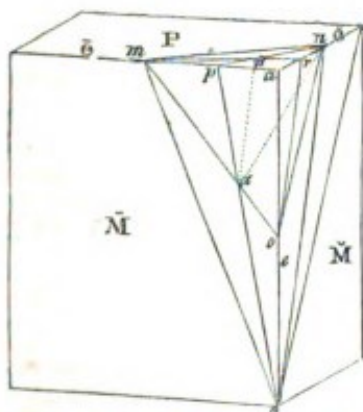
Bisogna sottolineare il fatto che, se l'architetto italiano ha introdotto e studiato per primo il parametricismo, già un secolo prima il termine "parametrico" era stato utilizzato, anche se in contesti diversi. Tale fase può essere definita come il parametrico prima della digitalizzazione avvenuta tramite i primi calcolatori. Infatti il termine parametrico si può incontrare per la prima volta nel 1837 in *The system of mineralogy* di James Dwight Dana, il testo che è considerato come codice del mineralogista. James Dwight Dana è stato un geologo e mineralogista e nel suo testo descrive la classificazione dei materiali basata sia sulle proprietà chimiche da essi possedute che sulla struttura cristallina. Per classificare i vari cristalli Dana introduce per la prima volta il termine "parametro" in quanto spiega come disegnare i cristalli tramite rapporti che sottostanno a variabili e parametri. In questo passo del testo di Dana compare la parola ed il concetto ad essa riferito: "Se il piano da introdurre fosse  $4P2$  il cui rapporto parametrico è  $4:2:1$ , dovremmo individuare allo stesso modo 4 parti di  $e$ , 2 di  $\bar{e}$  e 1 di  $e'$ ". Nelle pagine seguenti vengono poi descritte le equazioni che regolano il disegno dei cristalli seguendo sempre la variazione di alcuni parametri. È da sottolineare quanto sia interessante il fatto che dopo quasi due secoli di storia delle geometrie di oggi si basano su regole simili.

in the diclinic and triclinic systems. This is advisable because in these forms the lateral edges are equal and parallel to the vertical axis, and the basal edges, to the horizontal axes; and consequently in laying off the different planes, these edges may be substituted for the axes.

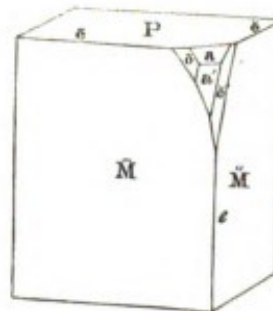
Suppose, for example, the right rectangular prism has been projected, (f. 161), and it is required to place on its angles the plane 2, whose parametric ratio is 2 : 1 : 1. Since 2 refers to the vertical axis, we lay off on the lateral edge ( $e$ ) twice as many parts of this edge as of each of the terminal edges, ( $\bar{e}$  and  $\check{e}$ ). Consequently, by taking a point in the edge  $e$  distant from  $a$ ,  $\frac{1}{2}$  the length of  $e$ , and a point in each  $\bar{e}$  and  $\check{e}$ ,  $\frac{1}{2}$  their respective lengths, and then joining these points, the conditions will be complied with, and the plane 2P will be constructed. If the plane to be introduced were  $4\bar{2}$ , the parametric ratio of which is 4 : 2 : 1, (in which 4 refers to the vertical axis and 2 to the longer horizontal) we should in the same manner mark off 4 parts of  $e$ , 2 of  $\bar{e}$ , and 1 of  $\check{e}$ ; if the plane were  $4\check{2}$  (in which 2 refers to the shorter horizontal axis), 2 parts of  $\bar{e}$  should be laid off, and 1 of  $\check{e}$ . By connecting the points thus determined, the plane  $4\bar{2}$  or  $4\check{2}$  would be delineated. If the plane were  $2\infty$  ( $2 : \infty : 1$ ), which represents a plane on the longer terminal edge, 2 parts of  $e$  should be laid off, and 1 of  $\bar{e}$ ; from the determined points in  $e$  and  $\bar{e}$ , lines should be drawn to the opposite edges parallel with the edge  $\bar{e}$ , and by connecting the extremities of the lines thus drawn, the desired representation of a plane  $2\infty$  would be completed. The same should be repeated on all the similar edges. This will suffice to illustrate the manner of substituting the edges for the axes, and also the method of delineating single planes.

The manner of determining the intersection of planes, we may illustrate by an example. Suppose it were required to place the planes 1, 2,  $4\bar{2}$ , and  $2\bar{2}$ , on a right rectangular prism. Two rectangular prisms should first be accurately projected by the method which has been explained. One, of a size which may be considered convenient for a representation of the crystal, drawn with light pencil marks; the other of larger dimensions, for the purpose of determining the direction of the intersections; these intersections, when determined, are to be transferred to the smaller figure. On f. 161 we may first lay down the plane 1, by drawing lines connecting the centres of the three faces about the angle. These lines are necessarily parallel to the diagonals of the three faces; the triangle  $mno$  is therefore the plane 1. By connecting the points  $m, b, n$ , the plane 2 is constructed; for the plane  $m\bar{b}n$  cuts off 2 parts of  $e$  to 1 of each  $\bar{e}$  and  $\check{e}$ , as the expression 2 requires. To lay off  $4\bar{2}$ , ( $4 : 1 : 2$ ), let the whole edge  $ab$  represent 4; then  $an$  ( $\frac{1}{2}$  of  $\bar{e}$ ) will represent 2 parts on the edge  $\bar{e}$ , and  $ap$  ( $\frac{1}{4}$  of  $e$ ) will equal 1 part on  $\check{e}$ , agreeably to the expression  $4\bar{2}$ ;  $npb$  is therefore the plane  $4\bar{2}$ . The perimeters of the planes  $npb$  ( $4\bar{2}$ )

161



162



and  $nmo$  (1) intersect one another in the points  $n$  and  $a$ ; consequently the line of intersection between these two planes must be situated between these points, and therefore the direction of the intersection of 1 and  $4\bar{2}$  is  $na$ .

The planes  $nmb$  (2) and  $npb$  ( $4\bar{2}$ ) intersect in the line  $nb$ , and therefore the intersection of 2 and  $4\bar{2}$  is in the direction of  $nb$ .

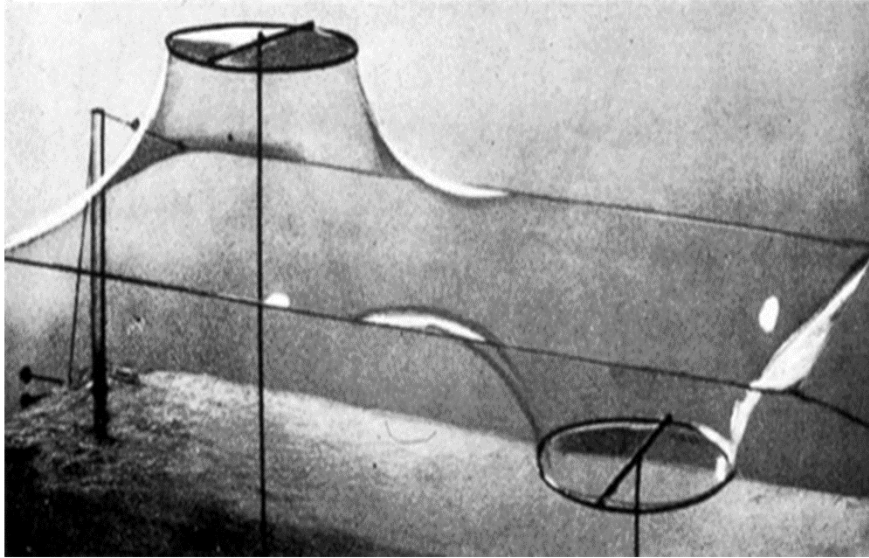
Figura 10: Estratto del libro "The system of mineralogy", circa a metà pagina si incontra per la prima volta il termine "parametrico" (J. D., Dana, The system of mineralogy of James Dwight

Dana, 1837-1868: Descriptive mineralogy, John Wiley & Sons, 1895).

Sempre prima dell'era digitale un grandissimo matematico e architetto, Antonio Gaudì, impiegò in gran parte dei suoi progetti una curva molto particolare: la catenaria. Non si sa se Gaudì era a conoscenza dei lavori precedenti sul parametricismo, ma è indubbio che i suoi studi siano ispirati da equazioni parametriche che lo portarono alla realizzazione quasi esclusivamente di superfici rigate matematiche. La catenaria è definita come la curva formata da una catena, una fune, la cui massa è uniformemente distribuita, tenuta alle sue due estremità e sottoposta solo alla forza di gravità che pende verso il basso. Tale curva particolare ha tutte le caratteristiche di una curva parametrica e Gaudì scoprì il modo per studiarla al meglio tramite il modello della catena sospesa, per distribuire in modo efficiente i carichi e ottenere forme organiche ispirate alla natura: egli modificava i parametri indipendenti generando delle versioni differenti ma sempre in perfetta compressione. Gaudì progettava pensando alla stabilità delle strutture, ma non come calcolo successivo, ma come parte del progetto stesso, e per questo utilizzò così tanto l'arco catenario nelle sue costruzioni. Questo è ciò che Gaudí disse della catenaria: "Dona eleganza e spiritualità all'arco, eleganza e spiritualità all'intera costruzione, evita contrafforti, l'edificio pesa meno e acquista una grazia vaporosa che può essere sorretta senza rari accessori ortopedici". Le catenarie di Gaudí rappresentano un connubio unico tra arte, architettura e scienza, evidenziando la sua genialità nel combinare estetica e funzionalità nelle sue opere.

Questa metodologia di visualizzazione della geometria per la sua realizzazione e non inizialmente nella soluzione di un calcolo matematico antecedente fu sperimentata anche da Frei Otto (1925 – 2015), architetto e ingegnere tedesco, noto per le sue innovazioni nell'architettura leggera e per il suo approccio ispirato alla natura. Una di queste innovazioni fu la sperimentazione che utilizzava i suoi studi sulle bolle di sapone. Il calcolo analogico è stato applicato da Frei Otto per comprendere la struttura e le proprietà delle bolle di sapone. Egli ha osservato come le bolle si auto-organizzano per formare la superficie minima tra i materiali disponibili, creando strutture leggere e resistenti, e ha analizzato i percorsi minimi trovati

attraverso la lana immersa nel liquido. L'architetto chiamò la progettazione con questi modelli form-finding.



*Figura 11: Frei Otto e gli studi sulle superfici minime e le lamine saponate (L. M. García Ballesteros, La parametrización del espacio procesos de diseño paramétrico, Escuela superior técnica de arquitectura de Madrid, 2017).*

In seguito, con l'avvento della digitalizzazione, i calcoli per effettuare la metodologia parametrica associata all'architettura furono di molto facilitati così da sostituire il percorso che fecero Gaudì e Otto, per esempio, con il calcolo delle espressioni parametriche tramite computer. Il primo che sperimentò e realizzò un supporto di progettazione fu l'architetto Ivan Sutherland il quale nel 1963 ideò Sketchpad, un software con l'obiettivo di creare "un sistema che permettesse ad un uomo e un computer di conversare", definito "A Machine Graphical Communication System". Questo software era capace, tramite una penna digitale, di disegnare quelle che oggi sono le banali funzionalità di un programma CAD come, per esempio, linee, punti e curve: ovviamente per l'epoca risultò rivoluzionario. Oltre a questi semplici comandi il software era in grado di connettere e generare relazioni

tra gli oggetti; per questo motivo si riuscì ad oltrepassare i limiti della rappresentazione additiva del disegno tradizionale arrivando a una più avanzata logica associativa. La potenzialità che ebbe Sketchpad, oltre l'ovvia innovazione e la rapidità con cui si poteva lavorare rispetto al passato, fu quella propria del parametrico, cioè il fatto che i progettisti erano liberi di modificare le connessioni create per avere un ricalcolo e una riconformazione della geometria molto veloce, cose che ora sono immediate. Questo mezzo rivoluzionò chiaramente il mondo dell'architettura, ma non quello commerciale. Infatti ci vollero quasi due decenni prima che un programma CAD riuscisse in modo definitivo a modificare il lavoro del progettista e fu nel 1982 che si brevettò AutoCAD. Nel 1997 ci fu un ulteriore importante passo in avanti: la Parametric Technology Corporation, un'associazione che fu fondata qualche anno prima, introdusse nel mercato il software Pro/ENGINEER sviluppato da Samuel Geisberg. Il programma permetteva l'associazione di oggetti tridimensionali parametrici che erano dettati da regole create dal progettista: si passò così da un disegno bidimensionale, quello tramite AutoCAD, ad uno tridimensionale di Pro/ENGINEER. Con il nuovo software si riuscì ad arrivare ad una riduzione del costo, quasi nullo, dei cambiamenti di progetto. Successivamente ci furono molti altri programmi che si basarono su modelli parametrici o li implementarono. Un esempio è Revit, sviluppato sempre dagli sviluppatori della Parametric Technology Corporation, anche se le equazioni matematiche parametriche vengono nascoste dietro l'interfaccia stessa. Dei vari software che è possibile utilizzare per la modellazione parametrica se ne discuterà più nello specifico all'inizio del capitolo successivo.

Dalle limitazioni del disegno tradizionale, però, si è passati a quelle dei software più moderni e a questo punto i progettisti si sono fatti strada nelle interfacce di scripting diventando anche programmatori, per ridefinire i codici dei programmi dall'interno. Nell'ultimo decennio è stata implementata una nuova versione di scripting, o programmazione, che viene definita visiva, la quale non è più sotto forma di stringhe a codici, ma ha una organizzazione a diagrammi di flusso. Questa tecnica è ormai usata ampiamente nei software

parametrici perché risulta più facile e immediata. Si tratterà sia delle tipologie diverse di software che si possono utilizzare sia della programmazione visiva, o nodale, nel prossimo capitolo.

In Italia alcuni degli esempi più rappresentativi delle prime sperimentazioni in architettura parametrica sono i lavori dell'architetto Sergio Musmeci e dell'ingegnere Pier Luigi Nervi. Nel decennio 1967-76 Musmeci realizza il ponte sul Basento, a Potenza, in cui concretizza le sue teorie sul minimo strutturale. A questo proposito egli poi disse: "bisognava arrivare a delineare con una formula matematica una sola soluzione di natura statica per avere la certezza del migliore impiego, cioè il minimo peso, di una data struttura" (dall'intervista a Zanaide Zanini di L. Garofalo sul Nuovo cantiere, 1998). Il ponte è una superficie continua che riduce al minimo la sua area e massimizza la sua funzione strutturale. Sergio Musmeci è riuscito a creare una magnifica superficie in calcestruzzo la cui estetica deriva direttamente dalla sua efficienza, studiata utilizzando modelli analogici e tecniche di ricerca della forma.

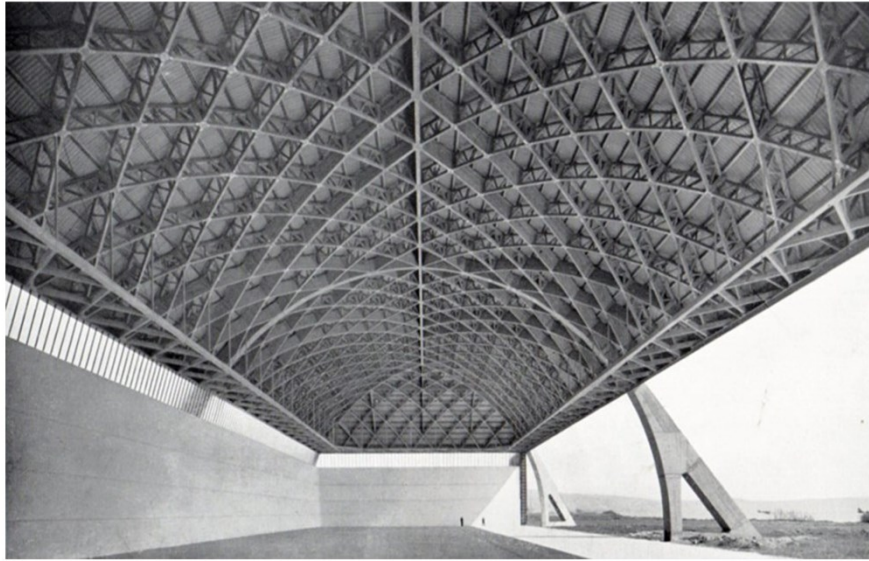


*Figura 12: Il viadotto dell'industria sul fiume Basento, conosciuto anche come ponte sul Basento o ponte Musmeci, 1976. (F. Giovanardi, Strutture fuori dal coro, Giovannardierontini, 2010).*

L'altra grandissima personalità che contribuì alla sperimentazione e alla realizzazione di opere di lettura parametrica fu Pier Luigi Nervi (1891-1979) che rappresenta uno dei maggiori artefici dell'ingegneria civile nel panorama internazionale del Novecento.

Pier Luigi Nervi operò nello stesso periodo di altri rappresentanti del Movimento Moderno e delle Avanguardie, ma risulta erroneo considerarlo con uno di questi movimenti. Tuttavia, può essere considerato altrettanto "moderno" per la sua fiducia nell'uso delle nuove tecnologie. Grazie alle sue ricerche e studi, Nervi rispose efficacemente alle esigenze del suo tempo, affermando il ruolo centrale del cemento armato nell'industria delle costruzioni, fu un pioniere nell'uso di questo nuovo materiale da costruzione, il quale egli riuscì a portare a livelli di raffinatezza estetica e funzionale ineguagliati. Le sue opere si distinguono per l'innovativa combinazione di forme e strutture, che spesso sfidano le convenzioni architettoniche dell'epoca. Nervi contribuì alla rottura dei paradigmi formali del razionalismo, le sue costruzioni, basate su ardite soluzioni tecnico-strutturali, raggiunsero risultati di straordinaria eleganza e diventarono icone di un nuovo modo di fare architettura, ammirate a livello mondiale. La sua maestria nel controllo delle forze e nel disegno delle superfici curve ha permesso la creazione di spazi di grande impatto visivo dove mostra come è possibile che l'ingegneria e l'estetica si fondono in un'armonia perfetta. Non a caso, le opere di Nervi sono state spesso paragonate a sculture monumentali, capaci di comunicare una potente bellezza formale attraverso la loro essenzialità strutturale. A lui si attribuiscono alcune delle costruzioni meglio realizzate e più belle del secolo che riuscivano a coniugare in maniera perfetta arte e la scienza del costruire; tra le opere più riconosciute troviamo: Aviorimesse dell'idroscalo di Orbetello, 1942, Palazzetto dello sport a Roma (1956-57), Aeroporto di Roma-Fiumicino (1957) e il Palazzo delle Nazioni del 1961.



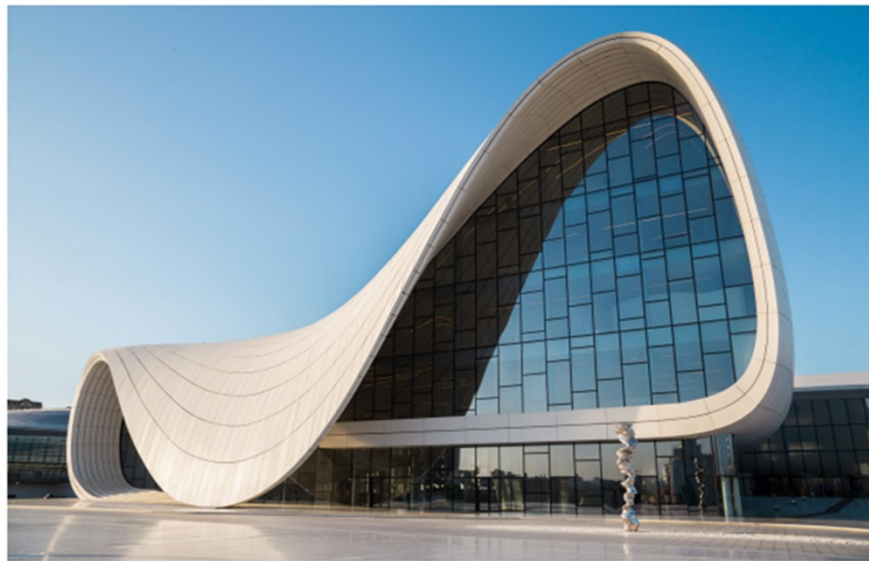


*Figura 13: Aviorimesse dell'idroscalo di Orbetello (GR), 1942 (abbattute durante la seconda guerra mondiale) (T. Trombetti, A. Trentin, La Lezione di Pier Luigi Nervi, Mondadori, Milano, 2010).*



*Figura 14: Palazzo dello Sport di Roma (progetto architettonico di Marcello Piacentini), 1957-59 (T. Trombetti, A. Trentin, La Lezione di Pier Luigi Nervi, Mondadori, Milano, 2010).*

Anche fuori dall'Italia non mancano architetti che parlano con un linguaggio parametrico in alcune delle loro opere; per citarne alcuni, Richard Buckminster Fuller, Fernando Romero, Felix Candela, Shigeru Ban e Jean de Gastines, Yurgen Mayer e Zaha Hadi. Quest'ultima ha realizzato l'edificio che più è presente nell'immaginario collettivo se si parla di design e architettura parametrica: il celebre Heydar Aliyev Centre a Baku (2012).



*Figura 15: Centro culturale Heydər Əliyev (2007–2012), Baku, Azerbaigian (J. Sanchez-alvarez, Practical aspects determining the modelling of the space structure for the free-form envelope enclosing Baku's Heydar Aliyev Cultural Centre, Editorial Universitat Politècnica de València, 2010).*

I softwares parametrici impiegati in architettura sono stati utilizzati con successo anche nelle progettazioni in altri settori tra cui il calzaturiero e la gioielleria, aprendo così nuovi scenari per quanto riguarda le architetture indossabili. Questo permette di capire quanto questa metodologia sia ampia e quale vasto spazio il design parametrico conceda alla creatività e alla immaginazione.



Figura 16: Le scarpe parametriche NU:S (2012), disegnate da Maurizio Degni, Alessio Spinelli e Arturo Tedeschi, sono state realizzate con un processo SLS utilizzando polvere di nylon (A. Tedeschi, S. Andreani, F. Wirz, AAD\_Algorithms-aided design: Parametric strategies using grasshopper, Le Penseur Publishe, 2016, pag. 250).

Dopo questa analisi sulla storia del design parametrico in architettura, si ritiene opportuno citare alcune righe prese dal cosiddetto “Manifesto del parametricismo” enunciato durante la Mostra Internazionale di Architettura della Biennale di Venezia, nel 2008, da Patrick Schumacher:

*[...] Parametricism is a mature style. That the parametric paradigm is becoming pervasive in contemporary architecture and design is evident for quite some time. There has been talk about versioning, iteration and mass customization etc. for quite a while within the architectural avant-garde discourse. The fundamental desire that has come to the fore in this tendency had already been formulated at the beginning of the 1990s [...]. This development was facilitated by the attendant development of parametric design tools and scripts that allow the precise formulation and execution of intricate correlations between elements and subsystems. The shared concepts, computational techniques, formal repertoires, and tectonic logics*

*that characterize this work are crystallizing into a solid new hegemonic paradigm for architecture. [...]*

### 4.3. Benefici e limitazioni della progettazione parametrica

I benefici della progettazione con metodologie parametriche sono molteplici e in questo paragrafo si cercherà di esporle tutte.

Grazie alla grande quantità di informazioni che i più nuovi e aggiornati software possono calcolare e tener in considerazione, uno dei vantaggi maggiore del design parametrico è quello che il processo di progettazione oggi può essere fatto continuare ad evolversi in maniera immediata e anche molto velocemente. Infatti, tenendo conto che il processo di progettazione architettonica tradizionale inizia con il primo contatto con il cliente e continua poi con una serie di passaggi laboriosi come lo sviluppo del progetto e la verifica dello stesso attraverso molteplici programmi informatici, l'invio poi, dopo un certo punto, agli ingegneri strutturali, agli ingegneri elettrici, al controllo qualità, ai costruttori, ecc., è evidente che si tratta di un processo lento e dispendioso in termini di tempo e, se è necessario apportare una qualche modifica ad una qualsiasi parte del progetto, si deve tornare con tutta probabilità al punto di partenza, ricominciando dagli schizzi iniziali per correggere eventuali dubbi o errori e poi ripercorrere l'intero processo. Questo meccanismo richiede di interrogare un numero molto elevato di professionisti, il che significa una forma di comunicazione alquanto inefficiente. Di conseguenza oggi numerosissimi studi di ingegneria e di architettura utilizzano programmi BIM<sup>7</sup> o CAD che si basano su parametri: questi, infatti,

---

<sup>7</sup> “Il BIM (Building Information Modeling) è il processo olistico di creazione e gestione delle informazioni di un asset costruito. Basato su un modello intelligente e alimentato da una piattaforma cloud, il BIM integra dati strutturati e multidisciplinari per generare una rappresentazione digitale di un asset durante tutto il suo ciclo di vita, dalla pianificazione e progettazione alla costruzione e alle operazioni” da <https://www.autodesk.es/solutions/bim>.

possono rendere il processo molto più veloce e semplice quando si tratta di correggere errori o modificare diversi aspetti in cui devono intervenire più aziende.

#### 4.3.1. I benefici del design parametrico

I benefici del design parametrico, come si è capito, sono numerosissimi, essendo quest'ultima una progettazione di tipo digitale che è assistita da computer e utilizza algoritmi per creare le possibili soluzioni di un problema. Il più intuitivo vantaggio è senza dubbio il fatto che esso può rapidamente modificare con un clic modelli tridimensionali senza nessuna fatica e senza rifare calcoli o ridisegnare daccapo il modello. Come già anticipato ampiamente nei paragrafi precedenti, questo permette di automatizzare il processo di progettazione risparmiando tantissimo tempo e semplificando notevolmente il flusso di lavoro e le tappe di un comune processo di tutta l'attività, eliminando azioni manuali dei comuni software che si basano su modelli CAD tradizionali come per esempio tracciare linee o ridimensionare angoli; nei modelli parametrici invece, proprio perché sono parametrici e i componenti sono connessi fra loro, ciò avviene spontaneamente. Un ulteriore beneficio del design parametrico, che fa sì che tale metodo sia estremamente interessante per i progettisti, è il fatto che il parametricismo offre la possibilità di creare forme, geometrie, modelli, aspetti, disegni, progetti, ecc., molto più complessi e innovativi mantenendo sempre un grado di precisione elevato e con un impegno di tempo basso (cose impossibili da realizzare con metodi di disegno tradizionale per qualità e tempo previsto). Ciò porta anche al fatto che l'architetto o l'ingegnere è più portato alla sperimentazione di soluzioni che un tempo erano limitate. L' utilizzo di software che si basano su una tecnologia parametrica come in BIM o in Grasshopper permette di avere un unico modello tra tutti gli agenti che lavorano al progetto riducendo drasticamente il tempo dedicato allo sviluppo di esso e di prendere al

momento le decisioni in merito alle modifiche del modello. L'utilizzo, inoltre, di tali software permette di processare analisi multiple nei progetti tramite appositi plug-ins. Progettare in questo modo diventa molto più efficiente in tutti gli aspetti del processo, crea un mezzo di comunicazione molto più efficiente tra le parti interessate, consente di eseguire un maggior numero di analisi prima che il processo di progettazione sia concluso, riduce notevolmente il tempo dedicato alle modifiche e il progetto può essere analizzato per verificarne la prosperità molto prima che possa essere costruito. A causa delle caratteristiche del parametro il modello è ben definito, cioè è controllabile in tutti i suoi aspetti dal progettista, e ciò permette non solo il controllo della forma e delle soluzioni ottimali (come già discusso nel capitolo precedente "Fondamenti dell'ottimizzazione", non bisogna dimenticare che questo processo fa sì che si raggiunga uno stato di ottimo desiderabile tra una vasta gamma di soluzioni possibili), ma anche del lato economico, del quale alle volte ci si dimentica. Finanziariamente gli architetti sono possono essere consapevoli del costo dell'intero progetto e ciò porta benefici ulteriori anche dal lato del cliente che può monitorare il costo del progetto minimizzando i possibili rischi.

#### 4.3.2. Le limitazioni del design parametrico

Nonostante, come si è appena visto, la progettazione del design parametrico offra molteplici benefici e vantaggi, bisogna anche citare quali svantaggi il suo utilizzo può portare. Uno dei problemi principali è quello della complessità legata all'utilizzo di questi software se comparata ad altri come i modelli CAD tradizionali, dove non servono particolari conoscenze e competenze nel mondo del digitale. Inoltre, essendo il modello parametrico basato anche su algoritmi ed equazioni matematiche, è possibile che eventuali piccoli errori, se non corretti fin da subito, portino alla compromissione del progetto intero. La conoscenza matematica e geometrica e soprattutto una certa esperienza nelle cosiddette digital skills

devono essere consolidate. Pertanto un problema per questo tipo di design parametrico potrebbe essere proprio l'apprendimento di tali materie che potrebbero causare ritardi durante lo sviluppo dei progetti.

Lylian Kubiak e Luca Sgambi in un recente articolo (L. Kubiak, 2020) discutono quale posto possa avere la programmazione nella nostra vita quotidiana e come mai gli architetti dovrebbero imparare a programmare. Dopo aver riportato interessantissimi spunti di riflessione e aver ricercato quanto oggi noi guadagniamo o perdiamo tramite l'utilizzo tecnologico in architettura, concludono dicendo come gli strumenti informatici hanno trasformato ormai il nostro modo di fare le cose e con esse anche il nostro modo di pensare l'architettura, la quale sta cambiando modalità di rappresentazione, analisi e progettazione. Gli strumenti digitali hanno trasformato e stanno trasformando l'etica tradizionale del disegno a mano per concentrarsi sulla progettazione computazionale nella attività futura. Il futuro dell'architettura e della progettazione si sta muovendo verso un approccio multidisciplinare dove design, programmazione, architettura e ingegneria si combinano di modo che le figure professionali siano sempre meno distinguibili:

*“The human plus the machine is not the future, it is our present. We have to face our fears if we want to make the most of our technology, and we have to overcome those fears if we want to get the best out of what humanity can give. By combining our strengths, human intuition and the computing capacity of the machine, human strategy, machine tactics, human experience, machine memory, would this be the most perfect project ever?”*





# 05 • L'utilizzo di nuove tecnologie per la progettazione parametrica

Come si è potuto leggere nelle pagine precedenti, la tecnologia al giorno d'oggi non solo può essere ritenuta fondamentale, ma addirittura necessaria. Da quando gli sviluppi nel campo del digitale sono entrati a far parte anche delle professioni degli architetti, disegnatori, ingegneri, progettisti e via dicendo, c'è stato un vero e proprio "maremoto elettronico" (L. Kubiak, L. Sgambi, 2020). Non si può più pensare nei termini se il digitale sia un bene o un male per la progettazione, ma si tratta di capire in che modo la programmazione influenzerà l'architettura. Di fronte alle numerosissime innovazioni tecnologiche che sono avvenute negli ultimi decenni ciò che è certo è il fatto di avere un cambiamento sempre più radicato nel nostro modo di pensare la progettazione. Guardando indietro, rivoluzioni tecnologiche in questo campo sono già avvenute, basti pensare, per esempio, all'epoca rinascimentale quando ci fu l'adozione di nuovi strumenti e procedure: come queste per l'epoca furono sconvolgenti, rilevanti e durature, così è possibile che oggi lo siano quelle digitali per l'elaborazione di progetti. Anzi, ormai siamo in grado di capire come già lo siano di fatto. Perché la rivoluzione digitale sta diventando sempre più importante oggi?

Per rispondere a tale quesito potrebbe essere utile nuovamente guardare indietro nel tempo. In principio le prime macchine tecnologiche, come le macchine calcolatrici meccaniche, le calcolatrici a nastro perforato e le prime calcolatrici elettroniche, richiedevano uno sforzo di calcolo e una potenza vastissima per rispondere a semplici calcoli banali operando al massimo un processo al secondo, mentre oggi i nostri smartphone sono in grado di elaborare

9,01x10<sup>15</sup> operazioni al secondo: il motivo è la legge di Moore<sup>8</sup>. Questa formidabile capacità dell'uomo di amplificare sempre più la potenza tecnologica è molto affascinante, ma spesso può essere anche spaventosa in quanto ci mostra la manchevolezza umana. Una vicenda molto interessante è quella del famoso match di scacchi tra Garry Kasparov, uno dei più grandi giocatori di scacchi della storia, e il supercomputer IBM Deep Blue che ha avuto luogo nel 1996 e poi nuovamente nel 1997. Nel primo match del 1996, Kasparov riuscì a vincere con una vittoria, tre patte e una sconfitta. Tuttavia, nel rematch del 1997, Deep Blue sorprendentemente ebbe la meglio su Kasparov, vincendo due partite e pareggiandone tre. Questi match hanno evidenziato la crescente potenza e capacità degli algoritmi e dell'intelligenza artificiale nel campo degli scacchi e hanno aperto la strada a ulteriori sviluppi nella relazione tra l'uomo e la macchina nei giochi e nella risoluzione di problemi complessi. Questo per dire come ormai non è più una corsa *contro* la macchina, ma *con* la macchina. Dobbiamo collaborare in questa corsa per programmare le possibilità future delle macchine, risolvere problemi, reagire alle prestazioni e all'economia del progetto. Attraverso queste ottimizzazioni possiamo risparmiare tempo di calcolo sui progetti mantenendo importanti vantaggi nella costruzione. In futuro, questi progressi consentiranno di stabilire collegamenti tangibili con lo sviluppo dei processi digitali per garantire il successo della costruzione automatizzata dall'uomo.

È possibile riscontrare questa rivoluzione tecnologica anche nel mondo dell'architettura e del costruire, a partire dagli strumenti divenuti sempre più digitali, dall'apprendimento automatico, dall'intelligenza artificiale ai Big Data; fa ormai parte integrante della quotidianità di un professionista ed è onnipresente. Il crescente interesse per questa materia in progettazione ha fatto sì che si creasse una nuova metodologia già ampiamente discussa nel capitolo

---

<sup>8</sup> La legge di Moore è una previsione formulata nel 1965 da Gordon Moore, co-fondatore di Intel. La sua osservazione iniziale affermava che il numero di transistor all'interno di un circuito integrato raddoppierà approssimativamente ogni due anni, portando a un costante aumento della potenza di elaborazione dei computer. Questa previsione si è rivelata sorprendentemente accurata nel corso degli anni, anche se alcuni ritengono che potrebbe rallentare in futuro a causa delle sfide tecnologiche e fisiche.

precedente, il metodo parametrico. Questa nuova forma di progettazione ha causato un enorme impatto all'interno del processo di progettazione tanto che al giorno d'oggi esistono numerosi programmi basati sul parametro e sugli algoritmi di calcolo. Grazie a numerosi vantaggi quali una capacità di controllo delle informazioni relazionate ai modelli tridimensionali e l'offerta di una possibilità di manipolazione non solo del modello ma anche delle idee, concetti e processi, ottimizzandoli, il design parametrico (chiamato anche design computazionale) è sempre più in crescita ed è considerato uno strumento impareggiabile. La progettazione parametrica può essere anche pensata come l'unione di due discipline quali la progettazione o il design e la programmazione o la computazione; è la sintesi tra le capacità di fattori creativi, funzionali, simbolici e produttivi, che sono la base di un processo di progettazione, con gli algoritmi e gli elementi del linguaggio digitale alla base della computazione. Il grande vantaggio del design parametrico sta nel poter fare delle connessioni logiche tra una serie variabile di dati e rendere automatizzate le possibili modifiche che altrimenti richiederebbero, oltre a tantissimo tempo, anche uno sforzo economico eccessivo, essendo l'architettura una disciplina che richiede di rispondere a più soluzioni anche contemporaneamente come, per esempio, cercare per un edificio la migliore esposizione solare possibile, un'estetica piacevole, una pianta funzionale, confort, adattabilità, ecc., e il tutto ad un costo ragionevole. Il design computazionale è lo strumento migliore con cui esprimere il proprio potenziale. L'utilizzo di nuove tecnologie per la progettazione parametrica riguarda tutti quei software (che si studieranno nel paragrafo 5.2) che si basano su una tecnologia parametrica tramite algoritmi.

Prima di discutere del significato di questo termine, è bene introdurre il concetto di flusso di lavoro. Un flusso è come una ricetta per il lavoro, è una sequenza di passaggi che guida, attraverso un processo, dalla prima idea all'azione concreta. Può essere usato in vari contesti, come nella gestione di progetti lavorativi oppure nella vita quotidiana per organizzare compiti. È un modo di dare struttura e ordine alle cose, come seguire una mappa per raggiungere una destinazione. I flussi di lavoro sono estremamente utili anche per quanto riguarda la progettazione parametrica in quanto permettono di ridurre al minimo i tempi necessari per

attuare soluzione. Tuttavia questo nuovo approccio necessita di una nuova mentalità del progettista per costruire la propria idea, come già accennato nel capitolo precedente, perché diventa necessario eseguire dei passaggi sequenziali che verranno analizzati nel paragrafo successivo. Questo approccio parametrico tramite flussi di lavoro consente una più efficace generazione di soluzioni che possono essere poi modificate a seconda del pensiero del progettista e meglio rispondere anche alle necessità e alle variazioni del progetto lungo tutto lo sviluppo del lavoro. Quindi, fin tanto che la logica e le regole che stanno dietro al flusso di lavoro, o algoritmo, rimangono incondizionate, anche questi ultimi possono supportare anche nel tempo il progetto.

In sintesi, queste nuove tecnologie, insieme alle sempre più numerose richieste e bisogni di edifici e strutture più efficienti anche da un punto di vista ambientale e della sostenibilità, hanno portato a un miglioramento e al superamento dei limiti che gli ingegneri e i progettisti si auto-prefissavano prima dell'inizio di un nuovo progetto, aprendo il campo a nuove possibilità. Di pari passo alle nuove esigenze è aumentata anche la complessità dei flussi di lavoro per cercare le soluzioni più all'avanguardia così da porre l'accento allo sviluppo della potenza di calcolo di software competenti e alla formazione di figure professionali pronte. In questa era la tecnologia continuerà ad avanzare a un ritmo sempre crescente e con essa la tecnologia stessa potrà essere utilizzata nelle pratiche di progettazione quotidiane; tuttavia, ciò che conta davvero sono ingegneri e designer di valore che apprendono, si adattano e abbiano capacità di applicazione nel processo di progettazione.

## 5.1. Modellazione algoritmica

Nel vasto panorama della scienza informatica, la modellazione algoritmica emerge come un campo cruciale e affascinante. La sua centralità nell'era digitale è innegabile, poiché le società si affidano sempre più ad algoritmi sofisticati per risolvere una gamma diversificata di problemi complessi.

Iniziamo definendo il concetto di modellazione algoritmica come un processo creativo che mira a tradurre problemi del mondo reale in rappresentazioni formali, manipolabili attraverso algoritmi con l'obiettivo primario di sviluppare modelli che catturino in modo accurato ed efficiente la complessità dei fenomeni in esame. La scelta o la creazione di algoritmi appropriati è un passo critico in questo processo, poiché influisce direttamente sulla qualità e sulla velocità delle soluzioni proposte. Una prospettiva fondamentale e interessante della modellazione algoritmica è la sua applicazione in settori come l'intelligenza artificiale, l'ottimizzazione, la simulazione e l'apprendimento automatico. Queste aree rappresentano un terreno fertile per l'innovazione, poiché algoritmi sempre più sofisticati stanno emergendo, consentendo alle macchine di apprendere da dati, ottimizzare processi e prendere decisioni autonome. Tuttavia il percorso verso algoritmi avanzati è disseminato di sfide, in quanto la complessità computazionale, la gestione delle risorse e le questioni etiche connesse sono solo alcune delle problematiche che richiedono un'attenzione accurata. In conclusione, la modellazione algoritmica rappresenta una pietra miliare nel progresso della scienza informatica.

Attraverso una attenta analisi delle sue metodologie e delle sfide connesse, la mia tesi mira a contribuire alla comprensione critica di questa disciplina in rapida evoluzione, delineando un percorso verso un utilizzo consapevole e responsabile degli algoritmi nell'era digitale. L'approccio alla creazione di un algoritmo per la soluzione di un qualsivoglia problema è in gran parte condizionato da quello studiato nel paragrafo 4.1.1 "Lo sviluppo del parametrico". In particolare, lo studio algoritmico della modellazione è indicato al secondo "passo" dello sviluppo parametrico (Preparazione del processo parametrico).

È possibile suddividere il lavoro in tre grandi step. In primo luogo è essenziale definire in principio quali sono le condizioni al contorno, definite come le condizioni iniziali del processo. Queste potranno essere modificate durante tutto il percorso progettuale e controllano la soluzione del modello finale. Dopodiché è importante definire su quali regole (matematiche,

geometriche, ecc.) l'algoritmo si basi, quali connessioni deve possedere e come le variabili del problema influenzeranno il lavoro. Si imposta così il flusso di lavoro. Infine, bisogna "descrivere" questo flusso logico al computer tramite un software apposito per poi farlo partire. L'utilizzo dello strumento dell'algoritmo è una notevole risorsa positiva per il mondo ingegneristico e quello architettonico perché aiuta il lavoro del progettista rendendo semplici alcune azioni ripetitive e permettendo di saltarle automatizzando il processo: infatti un algoritmo, una volta che è stato ben fatto e ottimizzato al meglio, può essere usato infinite volte modificando alcuni parametri e ottenendo infinite soluzioni con lo stesso flusso logico per poter scegliere quella più desiderabile senza mai ritornare al punto di partenza. Molte professioni, compresa quella dell'architetto e del progettista, nel ventunesimo secolo devono approcciarsi alla materia in modo differente: il loro lavoro non può più essere basato solo sull'intuizione e sull'esperienza per ottenere un risultato, ma piuttosto, senza rinunciare a queste competenze, si deve aggiungere un linguaggio nuovo, cioè il design computazionale.

### 5.1.1. Definizione e concetti di base

Ci si propone ora di esplorare il concetto di algoritmo, un pilastro fondamentale nel campo dell'informatica e della scienza dei computer: l'algoritmo rappresenta un insieme preciso e finito di passi organizzati per risolvere un determinato problema o eseguire una specifica operazione. La sua onnipresenza nell'era digitale è incontestabile, permeando ogni aspetto della nostra vita quotidiana. Pertanto si cerca di fornire una panoramica completa e approfondita dell'algoritmo, illuminando la sua importanza nella risoluzione di problemi computazionali e nella trasformazione della nostra realtà digitale.

Il termine algoritmo, che letteralmente significa procedimento di calcolo, ha un'antichissima origine dal termine latino medievale *algorismus*, il quale

a sua volta deriva dal matematico uzbeko Mohammed ibn-Musa al-Khowarismi, vissuto circa nel IX secolo d.C., il quale divenne celebre per aver scritto un libro sull'algebra. La nozione che noi oggi definiamo algoritmo fu utilizzata ben prima di quello che pensiamo, basti pensare che in maniera meno formale era utilizzata dai matematici greci come una sequenza finita di formule oppure ancora che si riesce a ritrovare la nozione di algoritmo già in documenti risalenti al XVII secolo a.C. (alcuni papiri trovati in Egitto contengono problemi con le relative soluzioni trovate tramite procedimenti algoritmici). Oggi col termine algoritmo si intende: “la descrizione precisa di una sequenza di azioni che devono essere eseguite per giungere alla risoluzione di un problema computazionale” (A. A. Bertossi, 1995). L'algoritmo può essere ulteriormente definito come quella procedura ben definita che prende un certo numero di valori definito input e ne produce altri i quali vengono definiti output. L'algoritmo è quindi una sequenza di passi computazionali che trasformano delle informazioni in altre (T. H. Cormen et al., 2010). Un algoritmo è una procedura utilizzata per ottenere una soluzione a un quesito richiesto oppure anche una specifica azione tramite un compito dato. L'importanza dell'algoritmo consiste nel fatto che queste risposte sono ottenibili solo attraverso “un elenco finito di istruzioni di base e ben definite” (A. Tedeschi, 2014). È interessante osservare come l'algoritmo, studiato anche nel capitolo quarto, segua in qualche modo l'intelligenza umana per raggiungere la soluzione dividendo il problema in più piccoli step, che possono essere risolti o calcolati in maniera molto semplice rispetto che cercare la soluzione al problema nella sua integrità.

Algoritmi e dati sono le due parti essenziali di qualsiasi soluzione di progettazione parametrica. Ma scrivere algoritmi non è banale. Il processo di progettazione algoritmica è altamente logico e richiede l'esplicitazione delle intenzioni progettuali e dei passi per raggiungerle. Tutti gli algoritmi comportano la manipolazione di dati e quindi algoritmi e dati sono strettamente collegati. Ciò, se ci si pensa, avviene esattamente anche tramite il linguaggio del computer. Per esprimere propriamente un

algoritmo, è importante che ci siano quattro caratteristiche principali (A. Tedeschi, 2014):

- I. Un algoritmo è un insieme non ambiguo di istruzioni correttamente definite.
- II. Un algoritmo si aspetta un insieme definito di input.
- III. Un algoritmo genera un output ben definito.
- IV. Un algoritmo può produrre messaggi di errore e avvertimenti all'interno dell'editor specifico.

È evidente quindi come gli algoritmi dipendono essenzialmente dalle istruzioni o regole che vengono loro imposte: se l'algoritmo non è definito chiaramente tramite le istruzioni, allora anche il risultato con tutta probabilità sarà scorretto. Inoltre gli algoritmi necessitano di una serie di input i quali possono essere di diverso tipo e quantità. Per esempio, allo step zero esso necessiterà di determinati numeri e informazioni, mentre in uno step successivo ne serviranno altri. Infine, un algoritmo, come visto, serve per generare degli output i quali anch'essi possono essere di diversa tipologia. Bisogna sottolineare il fatto che, se all'interno dell'algoritmo ci sono degli errori anche piccoli, esso non funziona e produrrà messaggi d'errore.

Nonostante la teoria matematica degli algoritmi si sia iniziata a studiare già nella prima metà del ventesimo secolo, solamente nella seconda metà le tecniche per progettare tali algoritmi e per analizzare la giusta correttezza ed efficienza si sono evolute in quanto erano necessari i calcolatori elettronici. Per scrivere e costruire degli algoritmi si sfrutta il potenziale del computer che al giorno d'oggi è una risorsa tanto fondamentale quanto accessibile. Esistono diversi programmi linguistici che permettono l'editing di specifici algoritmi e istruzioni (quelli più famosi



e più utilizzati al giorno d'oggi sono ad esempio C#, Python, ecc.), mentre gli editor integrati, cioè quelli editor di script forniti da programmi che consentono agli utenti di scrivere istruzioni per automatizzare le attività, sono Rhinoceros e Autocad e altri che verranno descritti in seguito.

È possibile definire alcune proprietà degli algoritmi come segue:

- I. Non ambiguità: per tutto il procedimento che definisce l'algoritmo non ci deve essere mai ambiguità e deve essere tutto chiaramente comprensibile.
- II. Generalità: gli step che vengono eseguiti dall'algoritmo dipendono solamente dal problema che si pone alla base e non dai dati, i quali possono essere di volta in volta mutati.
- III. Correttezza: l'algoritmo può essere considerato corretto se produce il risultato anche se le variabili in ingresso cambiano; per valutarne la correttezza si può usare una dimostrazione matematica o un'ispezione informale.
- IV. Efficienza: per misurare se un algoritmo è efficiente e quindi non inutile si guarda il tempo di esecuzione e la memoria impiegata. L'algoritmo deve essere facile da capire, facile da modificare e facile da mantenere.

Oltre che a considerarne le principali proprietà, è possibile dividere gli algoritmi in sottocategorie o classi: un algoritmo che porta a un numero è chiamato procedura di calcolo, mentre un algoritmo che genera un sì o un no è chiamato procedura di decisione. Una particolare tipologia di algoritmo può anche portare alla creazione di forme geometriche o di modelli tridimensionali grazie all'utilizzo di editor di scripting all'interno di software di modellazione quali CAD o altri. La geometria che si viene a creare si può manipolare e plasmare attraverso la sequenza di istruzioni e di parametri che viene insegnata all'algoritmo con alla base le cosiddette primitive come il punto, la linea, la curva, ecc. Gli oggetti non vengono più manipolati con il mouse, ma sono definiti da procedure espresse in uno

specifico linguaggio di programma digitale. Questo approccio innovativo muta, come si è visto, il legame tra il progettista e l'idea che esso vuole creare. La soluzione che si ottiene (output) alla fine del processo algoritmico non è semplicemente una traccia come quella che può creare una matita su un foglio di carta con l'unica differenza che questa è digitalizzata, ma il risultato può essere considerato come un modello digitale il quale si può modificare a proprio piacimento e che risponde a tutte quelle variazioni parametriche del sistema (input). Facendo un esempio, si può costruire un segmento partendo dalla sua definizione di retta compresa fra due punti e successivamente modificare tale segmento cambiandone la lunghezza le coordinate di partenza o le coordinate di arrivo, ecc., ma la connessione logica non cambia, perché il segmento, seppur diverso, è sempre un segmento. Questo può avvenire anche con geometrie molto più complesse le quali sono definite da una sequenza ben più difficile di istruzioni poiché la progettazione algoritmica è basata su relazioni associative tra entità, in questo caso geometrie. Come affermato più volte, l'algoritmo non crea un oggetto o una forma, bensì il processo che serve per arrivare a tale soluzione.

Altro concetto da definire è quello dei diagrammi di flusso, che sono estremamente importanti nell'illustrare il processo algoritmico di costruzione di una geometria. Un diagramma di flusso è una visualizzazione delle istruzioni, decisioni e trasferimenti logici applicati in un algoritmo, organizza efficacemente le relazioni associative tra le forme e le trasmette in un formato chiaro. Ciò è possibile grazie ai simboli convenzionali, tra cui rettangoli per le operazioni, rombi per le decisioni e frecce che indicano il flusso di controllo. È quindi facile capire e modificare il processo, in quanto qualsiasi cambiamento che si apporti in uno qualsiasi dei luoghi del diagramma verrà riflesso automaticamente nella forma finale. Nella pratica, il design algoritmico in diagrammi di flusso è utile nella progettazione, comprensione e comunicazione dei flussi di processo complessi sottostanti alla creazione della geometria in modo che ogni variazione possa essere tracciata e misurata rispetto all'intero.

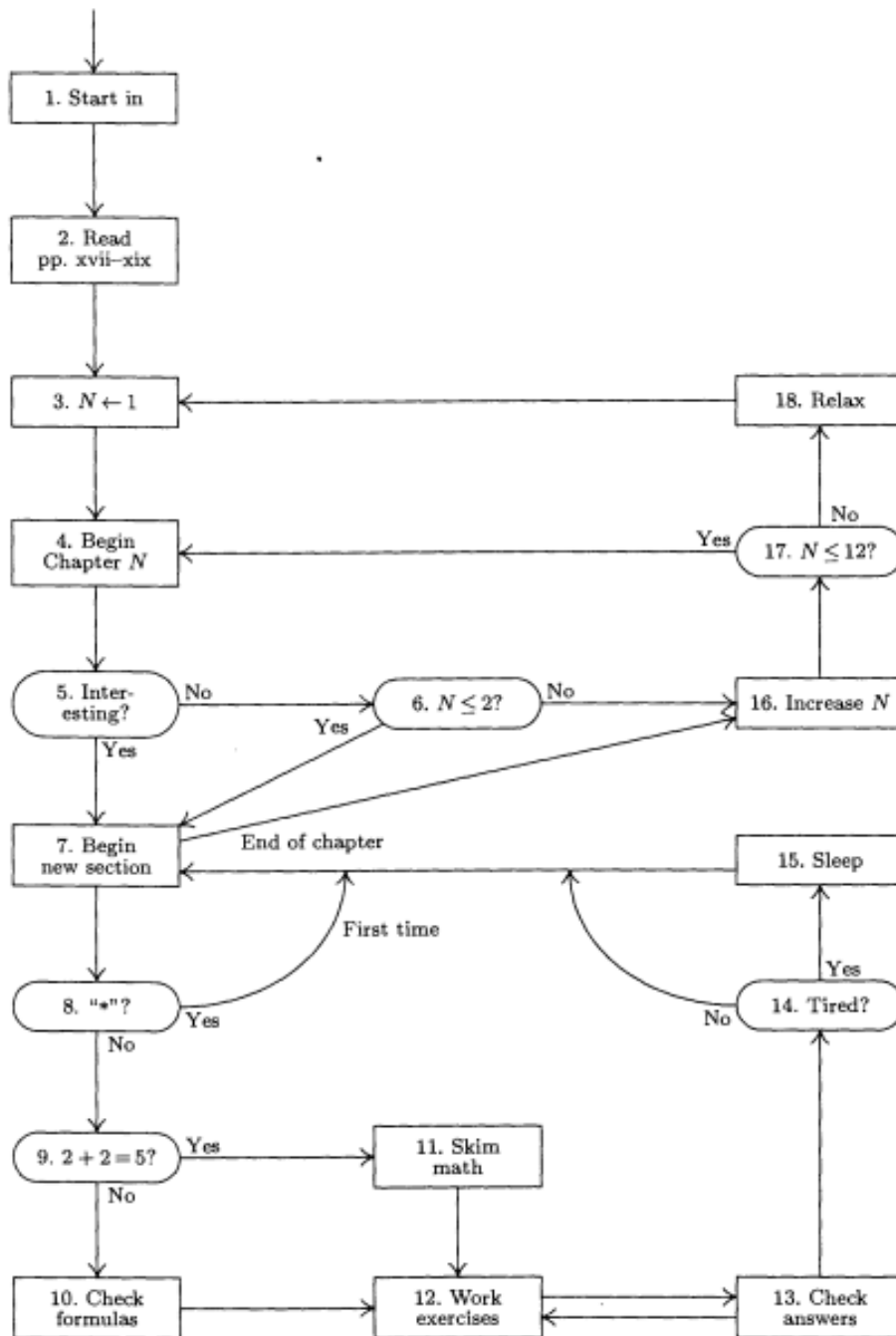


Figura 17: Diagramma di flusso creato da D. E. Knuth per leggere il suo libro "The art of computer programming" (D. Knuth, The art of computer programming, volume 1, Fundamental algorithms, Addison-Wesley Professional, Massachusetts, 1968, pag. 12).

Come è possibile vedere dall'immagine in alto, la creazione di un algoritmo si basa su istruzioni e regole in passi successivi tra loro, i quali possono

essere ben raffigurati come in questo caso da un diagramma di flusso. L'intelligenza di un algoritmo risiede nella sua capacità di dedurre nuova conoscenza ed estendere alcuni dei limiti dell'intelligenza umana. Un esempio più chiaro ancora rispetto al diagramma di flusso di D. E. Kuth è quello di paragonare un algoritmo alle fasi di una ricetta culinaria: raccogliere gli ingredienti, prepararli, combinarli, cucinarli e servirli sono fasi algoritmiche della preparazione di un cibo. In gioco ci sono numerose variabili che potenzialmente cambiano il risultato finale: il numero, la dimensione e la qualità degli ingredienti, la sequenza e la tempistica degli eventi, nonché il servizio e la presentazione del prodotto finale sono fattori chiave di una ricetta; esattamente come in un algoritmo per un calcolo matematico. Un algoritmo è l'astrazione di un processo e serve come schema sequenziale tramite istruzioni che porta alla realizzazione di un compito desiderato. Ad esempio, l'algoritmo (banalizzando) per la cottura delle patate può essere composto dai seguenti passaggi:

- I. Sbucciare
- II. Bollire
- III. Tagliare
- IV. Servire

Se si invertono le fasi o si aggiunge o si elimina una fase, è possibile creare ricette alternative che producono risultati diversi. Questi risultati possono essere migliori, uguali o peggiori rispetto all'intenzione originale. Come è normale, alterazioni, casualità o incidenti di percorso possono portare a soluzioni inaspettate e diverse, nessuna delle quali è prevista in anticipo. In questi casi, l'algoritmo serve come modello di pensiero, aiuta a comprendere il problema, affronta le sue possibili soluzioni ed è un veicolo per la definizione di nuovi problemi (K. Terzidis, 2006). Il pensiero algoritmico implica l'assunzione di un ruolo interpretativo per comprendere i risultati del codice generato, saper modificare il codice per esplorare nuove opzioni e ipotizzare ulteriori potenziali di progettazione. Stiamo passando da un'epoca in cui gli architetti usano il software a un'epoca in cui creano software (B. Peters, 2013).

### 5.1.2. Algoritmi in sistemi nodali

Gli algoritmi nei sistemi di nodi rappresentano elementi chiave nei campi delle reti complesse e della teoria dei grafi. In questo caso, un sistema di nodi è costituito da un insieme di nodi interconnessi, ciascun nodo rappresenta un'entità o un elemento e i collegamenti tra i nodi rappresentano relazioni o interazioni tra loro. Indipendentemente dalla loro complessità, tutte le soluzioni algoritmiche hanno tre elementi costitutivi: input (parametri), processo chiave (istruzioni) e output (soluzione). Si noti che il processo chiave può richiedere input e processi aggiuntivi. Comprendere gli algoritmi nei sistemi nodali richiede solide basi matematiche e la capacità di applicare concetti come la teoria dei grafi diretti e non orientati, le matrici di adiacenza e l'analisi della complessità degli algoritmi. L'obiettivo è sviluppare metodi efficienti per risolvere problemi specifici in reti complesse, ottimizzare le risorse di calcolo e garantire prestazioni ottimali. L'evoluzione della ricerca in questo campo è continuamente orientata verso lo sviluppo di algoritmi più sofisticati in risposta alle sfide emergenti nei sistemi nodali. Negli ultimi anni molti software basati su algoritmi parametrici hanno implementato questo nuovo modo di scrivere gli algoritmi: dall'editor classico di scripting si è passati ad uno più semplice, quello visuale. Questa nuova metodologia non solo è più semplice e permette di approcciarsi alla materia anche ad utenti che non hanno conoscenze di programmazione, ma è molto più intuitiva, permettendo un utilizzo migliore e più rapido. Il motivo della maggiore intuitività di questo metodo è dovuto al fatto che anche il nostro cervello ragiona per associazioni (vedi l'esempio della cottura delle patate nel paragrafo precedente), quindi anche lo scripting visivo, con regole di associazione di un metodo grafico basato su diagrammi di nodi, risulta più accessibile. Già lo Sketchpad dell'architetto Ivan Sutherland nel 1963, già citato nel capitolo precedente, era ideato a diagrammi e nodi; attraverso il diagramma e le relative connessioni mostrate a schermo l'utente poteva

manipolare il grafico con effetti istantanei sul disegno. In linea teorica tramite la rete di connessioni del diagramma è possibile generare qualsiasi elemento. Simile allo scripting, il visual scripting si basa su due processi di lavoro principali:

- il diagramma dei nodi, chiamato anche diagramma parametrico o algoritmo visivo;
- l'output del diagramma parametrico costituito da geometrie parametriche 3D o 2D.

Il diagramma parametrico ha il potenziale per creare modelli associativi che esplorano molteplici configurazioni attraverso il controllo dei parametri di ingresso. Si andrà ora a mostrare nella pratica come lo scripting visuale di alcuni software sia più facile e intuitivo anche da un punto di vista di visualizzazione e lettura. Si pone a confronto la costruzione di una figura geometrica primitiva, in questo caso un cerchio, in due linguaggi informatici differenti. Il primo linguaggio utilizzato è Java, linguaggio di programmazione ampiamente utilizzato e uno dei più popolari.

```
import java.awt.*;
import javax.swing.*;
public class DrawCircle extends JFrame {
    public DrawCircle() {
        setTitle("Drawing a Circle");
        setSize(400, 400);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
    @Override
    public void paint(Graphics g) {
        Graphics2D g2d = (Graphics2D) g;
        g2d.drawOval(150, 150, 100, 100);
    }
    public static void main(String[] args) {
        new DrawCircle();
    }
}
```

Come secondo esempio si è utilizzato Grasshopper, plug-in di Rhinocheros, costruendo sempre un cerchio di raggio variabile in un piano XY.

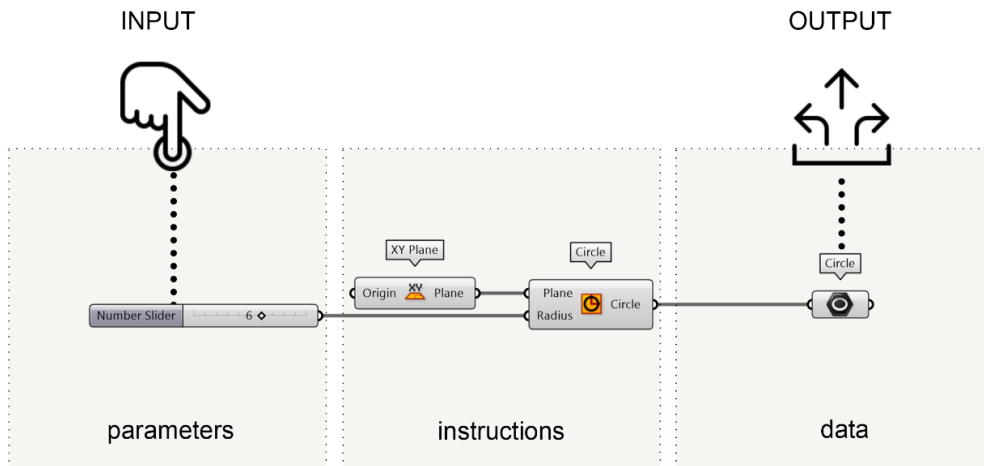


Figura 18: Creazione di un cerchio tramite algoritmo nel software Grasshopper.

È subito visibile come, nonostante i due esempi utilizzino un linguaggio della programmazione digitale, il secondo esempio che utilizza uno scripting basato su algoritmi nodali sia più intuitivo anche per coloro che non possiedono gli strumenti per capirlo.

## 5.2. Algoritmi genetici

Gli algoritmi genetici possono essere considerati come un sottogruppo o anche una particolare famiglia dei più vasti algoritmi definiti evolutivi. La loro storia inizia con l'idea di usare selezione e mutazione casuale per un compito di ottimizzazione con il lavoro dello statistico George E. P. Box che tuttavia non fece uso dell'elaboratore elettronico in quanto ancora troppo poco potente. Box giunse a formulare una metodologia statistica molto in voga negli anni successivi di nome *evolutionary operation* (G. E. P. Box, 1969). Negli stessi anni circa, altri studiosi concepirono l'idea di simulare l'evoluzione

tramite un computer; alcuni esempi sono Nils A. Barricelli e James Fraser, i quali utilizzarono simulazioni al calcolatore per studiare i meccanismi dell'evoluzione naturale; il biomatematico Hans J. Bremermann, cui va dato il credito di avere per primo riconosciuto nell'evoluzione biologica un processo di ottimizzazione (H., J. Bremermann, 1962); John Holland, un professore di psicologia e informatica, con i suoi studi negli anni Sessanta a seguito delle ricerche sull'adattamento dei sistemi naturali e artificiali. Holland era affascinato dalla teoria dell'evoluzione di Charles Darwin e si chiedeva se i principi evolutivi potessero essere applicati per risolvere problemi complessi. Questi algoritmi sono ispirati al processo di selezione naturale e riproduzione degli organismi viventi. L'idea di base è quella di creare una popolazione di soluzioni candidate a un problema, valutarle in base alla loro idoneità (o "fitness") e successivamente far evolvere le soluzioni attraverso operatori genetici come crossover e mutazione. Negli anni successivi altri ricercatori hanno contribuito a sviluppare e perfezionare gli algoritmi genetici. In particolare, David E. Goldberg ha scritto un libro nel 1989 intitolato *Genetic Algorithms in Search, Optimization, and Machine Learning*, che ha contribuito a diffondere la conoscenza di questa tecnica. Gli algoritmi genetici sono stati applicati con successo in una vasta gamma di campi, dalla ricerca operativa all'ottimizzazione dei processi industriali e creazione di componenti o forme ottimizzate nel soddisfare una moltitudine di richieste strutturali, funzionali, estetiche ecc. Come si vedrà in seguito, per andare ad eseguire un algoritmo genetico va prima descritto il problema in maniera analoga di modo che si adotti uno schema di rappresentazione opportuno individuando la funzione fitness (definita anche obiettivo) e i dati quali parametri e variabili. La loro forza risiede nella capacità di esplorare efficacemente lo spazio delle soluzioni e trovare soluzioni migliori nel corso del tempo in modo simile all'evoluzione biologica. Con il parallelismo creato tra la natura e gli algoritmi genetici anche molti termini usati nel linguaggio della prima materia vengono riutilizzati e fatti propri per esprimere concetti della seconda. Di seguito vengono mostrati i principali termini che verranno utilizzati più volte nel presente capitolo e anche in quelli successivi:



Biologic/Natural	Genetic Algorithm
Chromosome	String
Gene	Feature, character, or detector
Allele	Feature value
Locus	String position
Genotype	Structure
Phenotype	Parameter set, alternative solution
Epistasis	Nonlinearity

*Figura 19: Comparazione tra la terminologia della biologia e quella degli algoritmi genetici (D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co. Inc., Boston, 1989, p. 22).*

Andando ad esplorare più in profondità la caratteristica degli algoritmi genetici, si può evincere, considerando anche il nome che viene loro attribuito, come si può seguire un parallelismo tra gli algoritmi e la biologia. È infatti esperienza comune che gli esseri umani guardino alla natura per risolvere i problemi, a volte proprio prendendo ispirazione dalla natura, anche solo per l'immaginazione. Un esempio è il fatto che l'esistenza degli uccelli ci fa pensare che sia possibile volare e la tecnologia effettivamente ci ha portato al volo, anche se non nello stesso modo degli uccelli. Queste intuizioni furono chiamate algoritmi genetici, che rientrano nel campo della computazione evolutiva, che a sua volta è una branca dello studio dell'intelligenza artificiale. Gli algoritmi evolutivi, quindi, possono essere definiti come tecniche informatiche ispirate alla biologia e basate su una metafora: come avviene in natura che un individuo deve cercare di adattarsi all'ambiente circostante per sopravvivere e riprodursi, così le soluzioni che si possono trovare tramite questa particolare tipologia di algoritmo devono essere adatte a risolvere un problema. In linea generale ciò che succede in un algoritmo genetico è il fatto che una popolazione di possibili soluzioni per un problema viene fatta evolvere nell'ottica di trovare col passare delle generazioni soluzioni sempre

migliori al problema originale. Ogni individuo rappresentato generalmente come una stringa binaria<sup>9</sup> avrà una serie di caratteristiche che sono del tutto analoghe a dei cromosomi o a un corredo genetico detto genotipo. Perché questa evoluzione avvenga, l'algoritmo genetico necessita di una rappresentazione genetica del proprio campo delle soluzioni. Si tratta di una sorta di descrizione generica della popolazione e di quella che si chiama una funzione di fitness ovvero una funzione che valuti quanto i vari membri della popolazione siano adatti a sopravvivere, cioè a risolvere il problema. In biologia la fitness è una misura del successo riproduttivo di un individuo o di un genotipo: è qualcosa che ci dà un'idea di quanto quell'individuo o quei geni si siano adattati all'ambiente e soprattutto siano in grado di riprodursi e di passare alla generazione successiva. Di fatto la fitness ci dà un'idea di come cambiano le generazioni nel tempo e di quali geni della generazione precedente è più probabile che verranno passati alla generazione successiva. Un altro ruolo fondamentale della fitness è che ci dice quanto una soluzione di quelle che abbiamo è vicina a essere la soluzione del problema che stiamo risolvendo e di conseguenza quanto è probabile trovare delle soluzioni figlie nella generazione successiva. In altri termini si può dire che le soluzioni che vanno più vicine a quella che effettivamente sarà la soluzione del problema hanno più probabilità di riprodursi e di passare i propri geni alla generazione successiva. Il meccanismo procederà fin quando l'algoritmo non è terminato, ovvero l'algoritmo va fatto procedere fin quando o si trova una soluzione perfetta o rientrante negli standard minimi di ottimizzazione oppure non troviamo una soluzione o ci accorgiamo che il problema è risolvibile, ma che richiede troppo tempo (M. Gestal, 2013).

---

<sup>9</sup> La stringa può avere una lunghezza variabile a seconda della quantità di dati di parametri e di variabili all'interno del problema (solitamente nella formulazione del problema di algoritmi genetici la codifica della stringa avviene in sistemi binari, cioè in successione di zero e uno, ma è possibile codificare tale stringa anche tramite altre formulazioni quali per esempio sequenze del tipo "true" e "false").

L'algoritmo genetico si sviluppa in diverse fasi. Inizialmente la prima generazione viene generata randomicamente dall'algoritmo genetico, dopodiché viene eseguita la valutazione della popolazione che si crea; quindi si verifica se il criterio dettato dalla funzione fitness viene soddisfatto oppure no o in alternativa si verifica se è stato raggiunto il numero massimo di generazioni definite all'inizio del processo. Se durante la valutazione ci si accorge che la qualità delle soluzioni trovate e la nuova generazione creata non soddisfa le proprie "necessità", si utilizzano nuovi strumenti quali la selezione degli individui e l'applicazione di due operatori genetici esistenti quali l'incrocio (*crossover*) e la mutazione (*mutation*).

Le generazioni successive alla prima vengono composte selezionando, per mezzo di una funzione che imita la selezione naturale che sfrutta la fitness, le soluzioni nate da incrocio che possiedono le caratteristiche più adatte a risolvere il problema. La selezione degli individui è un'ottima strategia grazie alla quale si riesce ad ottenere una generazione migliore rispetto a quella generata in modo casuale all'inizio del problema perché, come in natura riescono a sopravvivere solo gli organismi che si adattano all'ambiente circostante, anche all'interno di un algoritmo genetico vengono scelti gli individui che più si avvicinano alla soluzione. La fase di valutazione è condizionata dalla cosiddetta funzione fitness che può essere anche definita funzione di adattamento. Essa può essere di duplice natura: infatti può essere sia singola sia multipla a seconda che la soluzione finale sia analizzata su uno o diversi aspetti prestazionali. Seguendo la metafora descritta in precedenza, come in natura il processo di selezione ed evoluzione delle specie è terminato dal fatto che solo gli individui migliori sopravvivano nell'ambiente circostante, anche negli algoritmi genetici la selezione avviene per prendere e conservare all'interno della popolazione solamente il corredo genetico (i dati, i parametri, le variabili) migliore di modo che si possano anche trasmettere alle generazioni future. All'interno degli algoritmi genetici si possono riscontrare diverse tipologie di selezione, le quali in modo differente esprimono il concetto appena descritto (A. Tettamanzi, 2005):

- I. Torneo di selezione: con il torneo di selezione vengono scelti a caso due individui all'interno della popolazione e viene poi scelto per la trasmissione del genoma solo quello con fitness migliore.
- II. Roulette: nel metodo della roulette esistono due diverse varianti. La prima, la cosiddetta roulette tradizionale, è esposta come la roulette di un casinò, con probabilità di selezione identiche per ogni individuo, mentre nella seconda tipologia le probabilità cambiano a seconda della qualità dell'individuo.
- III. Elitismo: nell'elitismo c'è essenzialmente la copia di uno o più individui che meglio performano all'interno di una generazione. Questo metodo è utilizzato per dare sempre gli stessi individui finché l'algoritmo non ne trova altri migliori e per non perdere qualità genetiche che le operazioni quali l'incrocio e la mutazione rischiano di far perdere.

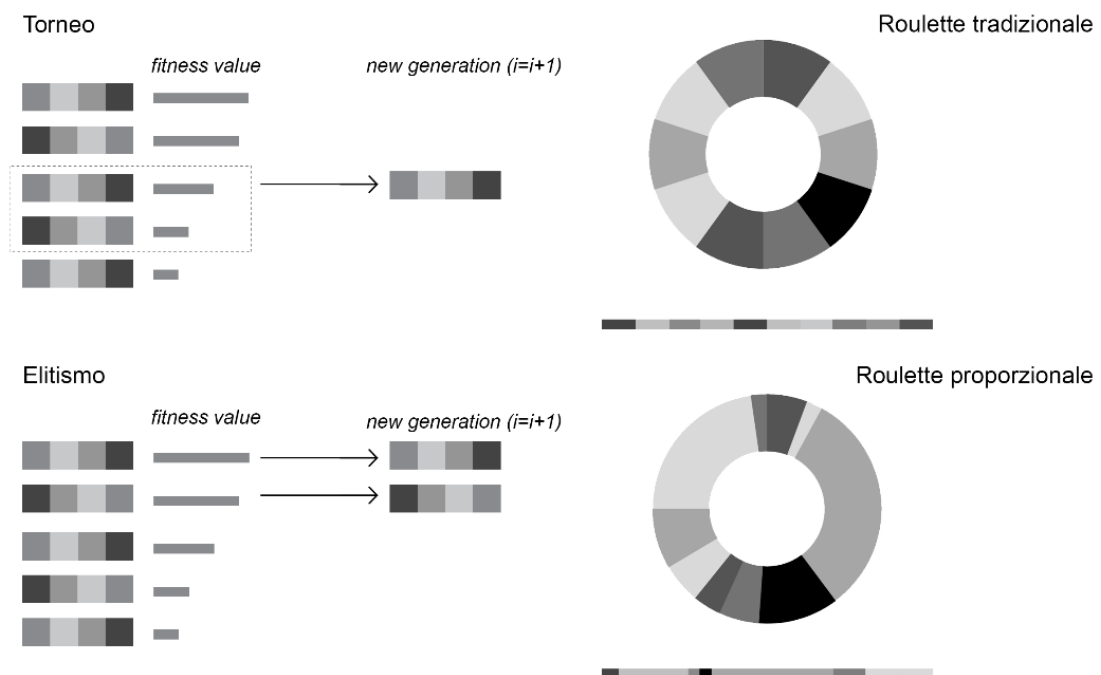


Figura 20: diverse tipologie di selezione all'interno di un algoritmo genetico (torneo, elitismo e roulette) (D. Madeddu, Architetture genetiche. Gli algoritmi genetici come strumento di ottimizzazione in architettura, Ricerche di Architettura, Gangemi, 2011).

Una delle ultime fasi dello sviluppo di un algoritmo genetico consiste nell'utilizzare opportuni operatori, l'incrocio e la mutazione, i quali servono per manipolare l'informazione genetica degli individui che sono stati selezionati nella fase precedente. Nella fase di incrocio avviene la ricombinazione che è il processo di riarrangiamento delle sequenze di DNA che avviene tra cromosomi omologhi provenienti da due diversi organismi. Questa avviene attraverso la rottura di pezzi di un cromosoma e il reinserimento nell'altro cromosoma. Come detto prima, ogni individuo della popolazione delle soluzioni può essere paragonata a un cromosoma: quindi ciò che fa la funzione di ricombinazione dell'algoritmo genetico è di fatto prendere due individui, spezzare in parti le stringhe che li rappresentano e inserire parti di una stringa nell'altra creando così nuove stringhe, le soluzioni "figlie". Ricapitolando, nel caso dell'incrocio, se accade ciò che avviene similmente nella riproduzione naturale ovvero che il codice genetico di due individui differenti viene rimescolato e ricombinato nell'espressione del figlio, questa operazione può avvenire con un certo numero di tagli in uno o anche in più punti, scambiando informazioni fra le stringhe di una popolazione.

Questo processo aumenta la variabilità della popolazione, ma non è l'unico a poterlo fare. Esiste infatti la possibilità di inserire un'ulteriore funzione all'interno dell'algoritmo genetico: la mutazione. Questa, in maniera simile alla sua analogia biologica, introduce una modifica alla sequenza. La fitness della sequenza così modificata viene poi calcolata e tenuta in considerazione dalla funzione di selezione naturale per formare la nuova generazione. Si può dire che la mutazione fa sì che avvenga una modifica all'interno del patrimonio genetico di un individuo introducendo materiale genetico differente. All'interno di un algoritmo genetico, allo stesso modo, si può dire che vengono aggiunte nuove informazioni o dati all'interno di una popolazione composta da stringhe. Tale operazione è molto significativa in quanto va ad esplorare nuove risorse e soluzioni più vaste, evitando di contare sempre sulle stesse informazioni. Un'ottima strategia è quella di applicare entrambe le operazioni genetiche, incrocio e mutazione, per poter tenere i geni migliori delle popolazioni

“genitori” e avere contemporaneamente possibilità di variazione per fornire altri geni ottimali. Una volta completata la nuova popolazione creata dai vari tipi di selezione e dalle operazioni genetiche quali incrocio e mutazione è possibile poter fare nuovamente la valutazione degli individui. Nuovamente, se si raggiungerà la soluzione ottima o il numero massimo di generazioni o ancora l'indicatore di convergenza, si potrà terminare l'algoritmo genetico. Se così non fosse, in modo iterativo si ritornerà a fasi precedenti dell'algoritmo genetico (A. Tettamanzi, 2005).

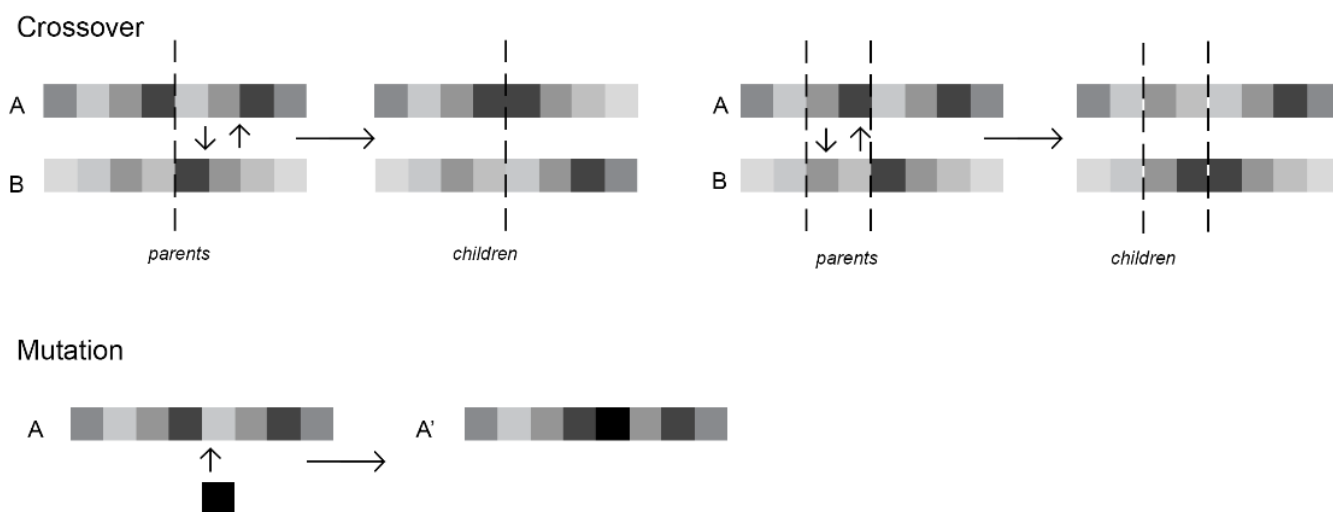


Figura 21: Rappresentazione schematica di come funzionano gli operatori genetici in un algoritmo (crossover e mutazione) (D. Madeddu, Architetture genetiche. Gli algoritmi genetici come strumento di ottimizzazione in architettura, Ricerche di Architettura, Gangemi, 2011).

Questi algoritmi hanno un grosso vantaggio dato dalla funzione di selezione naturale: elimina una delle maggiori difficoltà che i software designer si trovano ad affrontare, cioè il dover specificare in anticipo tutte le caratteristiche del problema e le azioni che il programma dovrebbe eseguire quando si trova ad avere a che fare con esso. Ovviamente, però, questi

utilissimi algoritmi hanno alcuni limiti che non li rendono sempre ideali per ogni situazione: è vero che trovano la miglior soluzione, ma di fatto questa lo è solo in quanto paragonata alle altre prodotte dall'algoritmo stesso e pertanto non è detto che sia la migliore in assoluto. Non è sempre facile stabilire quale sia il criterio più adatto per stoppare l'algoritmo. In effetti sembra che gli algoritmi genetici non siano molto efficienti quando si ha a che fare con problemi ad alta complessità che in genere hanno un elevato numero di individui nella popolazione in studio per i quali si preferisce spesso usare altri algoritmi evolutivi (A. Tettamanzi, 2005).

Inoltre, un altro aspetto da considerare riguardo agli algoritmi genetici è il loro comportamento nei confronti della diversità genetica: durante il processo evolutivo, è possibile che la popolazione converga prematuramente verso soluzioni subottimali (cioè soluzioni che si avvicinano a quella di ottimo o che in qualche aspetto ci assomiglia) a causa della perdita di diversità genetica. Questo fenomeno, noto come *premature convergence*, può portare a una stagnazione della ricerca di soluzioni migliori, limitando l'efficacia dell'algoritmo. Per mitigare questo problema, si possono adottare tecniche come la mutazione adattativa o l'introduzione di meccanismi di mantenimento della diversità, come il *niching* o il *crowding*.

In conclusione, va sottolineato che questo argomento è vasto, perché abbraccia molteplici sfaccettature e approfondimenti. Al fine di mantenere la chiarezza e la focalizzazione, la presente trattazione si è concentrata solo sui concetti essenziali. Ciò non esaurisce, tuttavia, la ricchezza e la complessità dell'argomento.

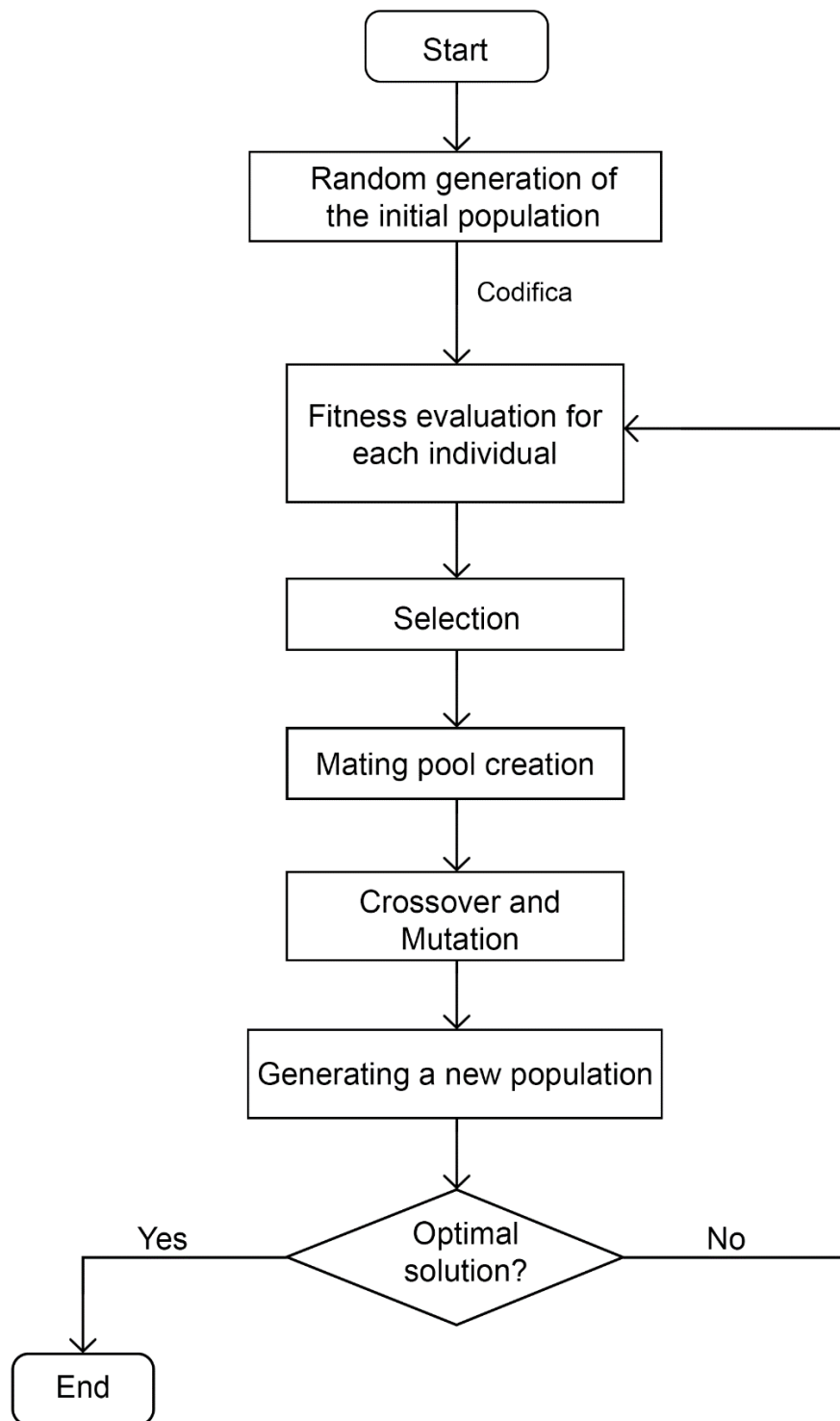


Figura 22: Diagramma di flusso di un algoritmo genetico (S.N.Sivanandam, S.N.Deepa, Introduction to Genetic Algorithms, Springer Berlin Heidelberg New York, 2008, pp. 20-30, 33-35).



### 5.3. Panoramica dei software parametrici principali

Analizzati nei dettagli tutti concetti che stanno alla base del parametrico e della creazione di algoritmi, è ora possibile esporre una panoramica dei software parametrici più conosciuti. Nell'ambito della progettazione contemporanea, l'utilizzo di software parametrici si è affermato come un pilastro fondamentale, ridefinendo radicalmente il modo in cui concepiamo e sviluppiamo design complessi. Al centro di questa trasformazione troviamo strumenti avanzati che consentono una progettazione visuale e algoritmica che va oltre i limiti della creatività convenzionale. Questi strumenti sono diventati indispensabili per gli architetti e i designer, permettendo la creazione di forme dinamiche e geometrie innovative. In sintesi, l'evoluzione della progettazione parametrica ha aperto le porte a un nuovo paradigma creativo. L'integrazione di algoritmi, l'automazione dei processi e la flessibilità dei parametri stanno ridefinendo il nostro approccio al design, anticipando un futuro dove la progettazione è intrinsecamente legata a una dinamica fluida e in continua evoluzione. I programmi che si basano su strutture parametriche hanno un'enorme utilità del campo della progettazione architettonica, non solo in quanto possono essere definiti come dei modellatori di forme e geometrie volute dall'utente molto complesse, ma anche in quanto sono in grado di "comandare" il software a proprio piacimento grazie allo scripting, la scrittura di codici, cosa che precedentemente non era possibile fare utilizzando i tradizionali software CAD per la modellazione tridimensionale. Come ampiamente discusso nei paragrafi precedenti, è possibile scrivere codici anche tramite algoritmi visualizzati su nodi e quasi tutti i software parametrici di ultima generazione adottano questa metodologia. Inoltre è possibile vedere in tempo reale la forma creata sullo schermo. È in questo modo che lavorare tramite questi software è molto semplice e ha reso questo lavoro aperto anche a coloro i quali non sono esperti programmatori. L'utente può essere considerato a metà tra un architetto, progettista e un programmatore in grado di agire sulla propria creazione in modo diretto

tramite i parametri e l'algoritmo stesso. L'utilizzo di questi software ha aperto le frontiere alla sperimentazione di nuovi studi sull'architettura ampliando lo sguardo ad una più vasta gamma di opzioni compositive architettoniche. Va sottolineato che, durante l'utilizzo di questi strumenti, ciò che diviene fondamentale è la scrittura del codice rispetto alla rappresentazione dell'oggetto. Diventa più importante che le connessioni dell'algoritmo siano corrette e ben definite, mentre è posto in secondo piano il riportare banalmente la forma. Infatti la forma visualizzata può cambiare totalmente tramite pochi clic e soprattutto in tempo reale o quasi, perché la velocità di calcolo dei più attuali computer è elevatissima. Anche in base alle abilità del designer di utilizzare l'opportuno programma e nella maniera più corretta possibile la scrittura dei codici, si riuscirà a raggiungere gli obiettivi prefissati inizialmente. Oggigiorno tramite internet è possibile usufruire molto semplicemente di moltissimi programmi parametrici. Tra i software più comuni utilizzati dai progettisti, che vogliono approcciarsi alla progettazione parametrica, si possono riscontrare i seguenti (bisogna però ricordare che questi sono solo una piccolissima parte e in rete la scelta di un software parametrico è ampissima):

- Grasshopper, associato a Rhinoceros;
- Dynamo, associato a Revit (Autodesk);
- Digital Project, di Gehry Technologies;
- Generative Components, distribuito da Bentley System
- Solidworks
- Unreal Engine

La scelta di utilizzare il software Grasshopper come strumento principale nella realizzazione di questo lavoro di tesi e il lavoro del caso studio nel capitolo sesto è stata motivata da diverse considerazioni. Come evidenziato nella tabella sottostante (fig. 23), Grasshopper emerge come il software più

ampiamente adottato all'interno della comunità di progettazione e analisi parametrica. La sua popolarità non è solamente riflessa nel numero di utenti, ma anche nella sua reputazione di efficacia e intuitività. La flessibilità offerta da Grasshopper nel manipolare parametri e generare geometrie complesse, unita alla sua integrazione con altri ambienti di modellazione e simulazione, ha reso questo software la scelta ideale per affrontare le sfide specifiche poste dalla ricerca. La sua interfaccia visuale e la vasta gamma di plug-in disponibili hanno ulteriormente facilitato il processo di sviluppo, consentendo una rapida iterazione e ottimizzazione dei modelli. In sintesi, l'adozione di Grasshopper si è dimostrata cruciale per il successo e la robustezza delle analisi condotte in questo lavoro di tesi. I benefici che si possono riscontrare in Grasshopper come piattaforma di lavoro per la modellazione parametrica possono essere così elencati (A., Tedeschi, et al., 2016):

- Comunità ampia, dinamica e in crescita; Grasshopper non è solamente un software indipendente, ma permette tramite un network di utenti di condividere i lavori, il sapere, domande, discussioni, sfide e problemi.
- Aggiornamenti costanti; correzione dei bug e nuove funzionalità si basano sul feedback degli utenti.
- Ecosistema; tramite i numerosi plugins indipendenti di Grasshopper possono estendere infinitamente il proprio potenziale.
- Interazione tra software; è possibile creare interazioni fra software diversi con Grasshopper, per esempio Excel, Photoshop, Revit, etc.
- Interazione tra hardware.

Producer - Software	Visual Programming/Paratric Plug-In	% users
Gehry Technologies: Digital Project	n/a	3%
Bentley Systems: Mictostation	Generative Components	0%
Autodesk: Revit Architecture	Dynamo (2011)	7%
McNeel and Associates: Rhinoceros3D	Grasshopper (2008)	90%
Nemetschek North America: Vectorworks	Marionette (2015)	<1%

*Figura 23: Percentuale di utenti che dichiarano di utilizzare software parametrici (Cichocka, W. N. Browne, R. R. Edgar, Optimization in the Architectural Practice - An International Survey ,caadria, 2017, p.387).*

### 5.3.1. Interfaccia e funzionalità principali di Grasshopper

Grasshopper, un plug-in per Rhinoceros®, è un editor basato su nodi sviluppato da David Rutten presso Robert McNeel & Associates. Creato nel 2007 per Rhino 4.0 - originariamente chiamato Explicit History - è stato ribattezzato Grasshopper nel 2008. Nel giro di pochi anni il plug-in si è guadagnato una vasta comunità di utenti e sviluppatori, tra cui studenti, accademici e professionisti. Grasshopper è disponibile gratuitamente. L'interfaccia principale di Grasshopper è composta da una finestra la quale lavora in contemporanea con l'interfaccia principale di Rhinoceros. Tramite i componenti, che corrispondono ai nodi dell'algoritmo, si costruisce la propria geometria 3D che verrà visualizzata in modo istantaneo, o quasi, per i processi di calcolo più complessi, in Rhinoceros.

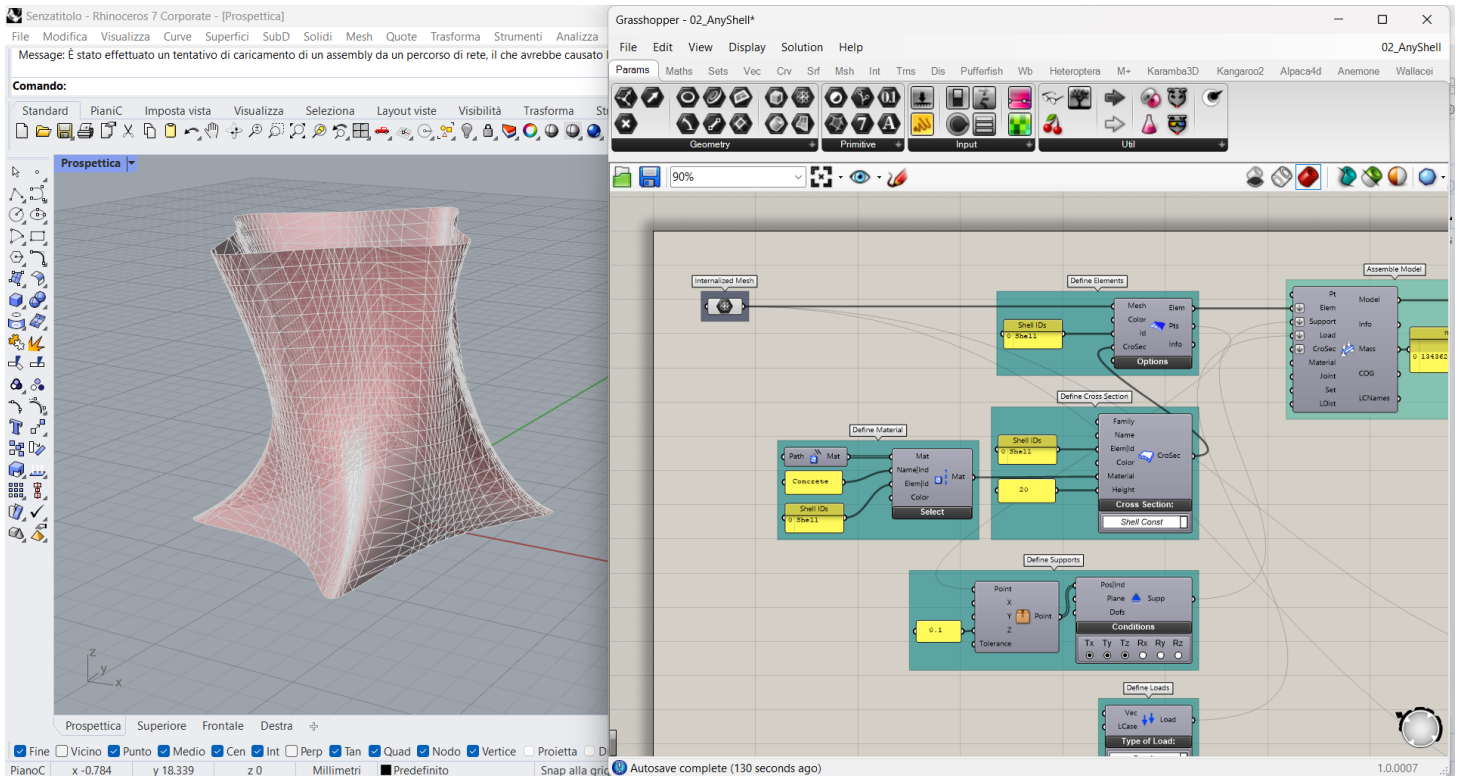


Figura 24: L'immagine rappresenta l'interfaccia di Rhino per la modellazione e la finestra di Grasshopper per il relativo spazio di lavoro.

Nello specifico, nella finestra di lavoro di Grasshopper sono presenti quattro zone principali:

- La barra del menù: la barra del menù possiede le caratteristiche classiche del layout di Windows. Tramite essa è possibile fare le operazioni più elementari e comuni come in ogni classico software, per esempio, salvare file, aprire file, modificare le impostazioni, etc.
- La tabella dei componenti: tramite la tabella dei componenti è possibile scegliere i propri elementi per creare l'algoritmo visivo. I componenti possono rappresentare diverse entità, come, per esempio, primitive, geometrie, operazioni etc. e vengono suddivisi per tabelle e successivamente per pannelli. All'interno di ogni categoria vi ci saranno numerosi componenti che svolgono le

funzioni appartenenti alla categoria corrispondente (all'interno della categoria Params ci saranno i pannelli Geometry, Primitive, etc., i quali conterranno i propri componenti come Circle, Curve, Plane, etc. e nello stesso modo nella categoria Surface ci saranno i pannelli Analysis, Freeform, etc. e così via).

- La barra degli strumenti della tela: tramite essa è possibile modificare le impostazioni di visualizzazione e altri comandi minori.
- L'area di lavoro: è lo spazio dove l'utente trascina i componenti e collegarli fra loro per creare l'algoritmo parametrico nodale.

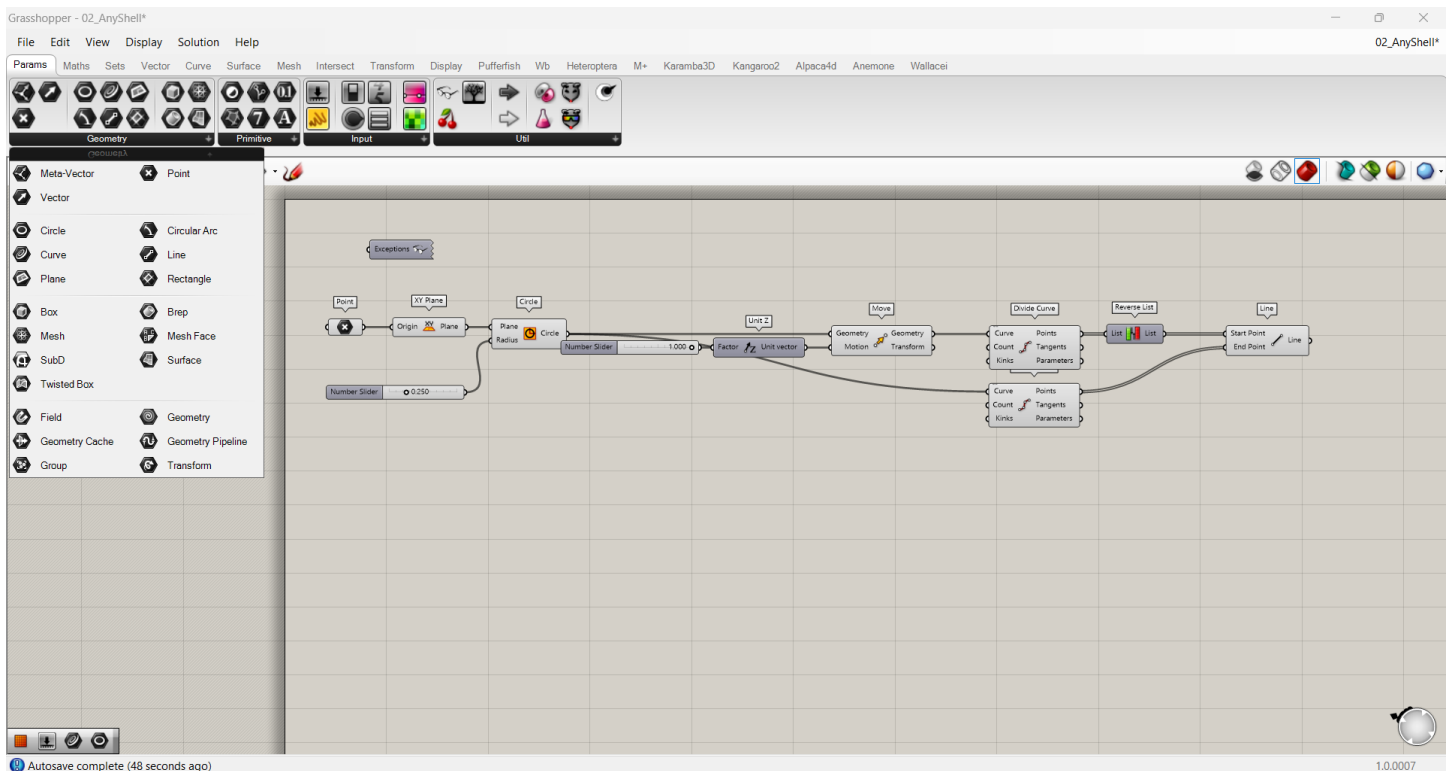


Figura 25: L'immagine mostra l'interfaccia principale di Grasshopper con barra del menù, tabella dei componenti, barra degli strumenti e l'area di lavoro.

Come già discusso ampiamente nei capitoli e paragrafi precedenti, si è dimostrato come la costruzione di un algoritmo nodale possa diventare semplice tramite uno software visivo. Tale costruzione avviene grazie a delle istruzioni fornite dall'utente e messe in relazione fra loro, le istruzioni sono i componenti. Esistono tre differenti tipologie di componenti:

- **Componenti standard:** questi sono i più comuni e sono costituiti da tre parti: l'input, cioè i dati in entrata, il nome del componente e l'output, cioè i dati in uscita dopo l'operazione processata. Per esempio il componente Construct Point (vedi immagine di seguito) costituisce lo strumento per costruire un punto nello spazio, necessita di tre dati, coordinata x, coordinata y e coordinata z (input) e fornisce per l'appunto il punto desiderato (output).
- **Input components:** sono quei componenti che forniscono i dati (numeri, colori, parametri booleani, etc.) per i componenti standard ma che non gli rielaborano.
- **Container components:** questa tipologia di componenti contengono i dati dell'algoritmo.

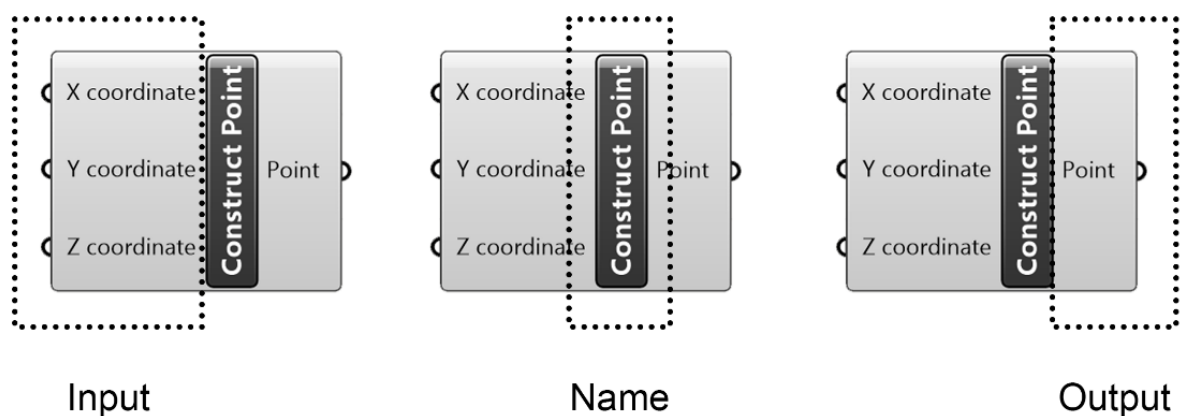


Figura 26: Il componente Construct Point in Grasshopper.

*"Grasshopper seems to be winning out in the competitive struggle for domination as the preferred tool for scripting, at least in the avant-garde segment of the discipline"*

*Patrik Schumacher*

### 5.3.2. Strumenti e plug-ins per l'ottimizzazione mono-obiettivo

All'interno del software Grasshopper associato a Rhinoceros è possibile trovare soluzioni ottimali per i propri problemi tramite plug-ins di ottimizzazione, ovvero, il raggiungimento di una posizione di ottimo, ossia del massimo risultato possibile con i termini dati o in relazione a un determinato fine. Infatti Grasshopper, come anche altri principali programmi parametrici, ha la possibilità di installare al proprio interno software aggiuntivi che forniscono funzionalità aggiuntive al programma principale. In sostanza, è come un accessorio che si può aggiungere al software per estenderne le capacità o personalizzarne le funzioni. Questi "software aggiuntivi" vengono chiamati plug-ins.

Esistono due diverse famiglie di plug-ins per l'ottimizzazione parametrica: i plug-ins mono-obiettivi e i plug-ins multi-obiettivo. La differenza maggiore che si riscontra è ovviamente, come dice il nome stesso, il numero di obiettivi, o per meglio dire fitness, che vogliamo dare al nostro programma secondario. Come descritto nel capitolo terzo, infatti, è possibile costruire gli algoritmi per l'ottimizzazione o avendo come ottimo un unico obiettivo tale da essere soddisfacente ed efficace per il nostro problema o assumendo, anche più comunemente, come traguardo una moltitudine di possibilità decisionali e di possibili criteri fra cui scegliere la soluzione "giusta" (come visto più volte gli algoritmi genetici trovano spesso la soluzione che si avvicina di più, ma che non è detto sia l'unica). Entriamo innanzitutto in un processo decisionale caratterizzato da



molteplici obiettivi, tanto da poter dire che una decisione è più o meno preferibile ad un'altra, a seconda dei compromessi tra tutte le variabili del problema in esame. In questo paragrafo si andranno a vedere due plug-ins mono-obiettivi principali per poi nel paragrafo seguente esplorare quelli multi-obiettivo.

In Grasshopper esistono numerosissimi plug-ins di qualsiasi tipologia; tra i più scaricati, utilizzati e valutati meglio ci sono Galapagos e Silvereye. Galapagos ha due input: il Genoma e il Fitness. Il valore di fitness può essere descritto, come si è visto più volte, con il valore che si desidera ottimizzare, il genoma, invece, è un insieme di parametri che influenzano il fitness (i dati del problema).



*Figura 27: Il componente Galapagos in Grasshopper*

Facendo doppio-click sul componente di Galapagos possiamo ora entrare nella sua interfaccia e trovare la soluzione, non prima di aver predisposto impostazioni.

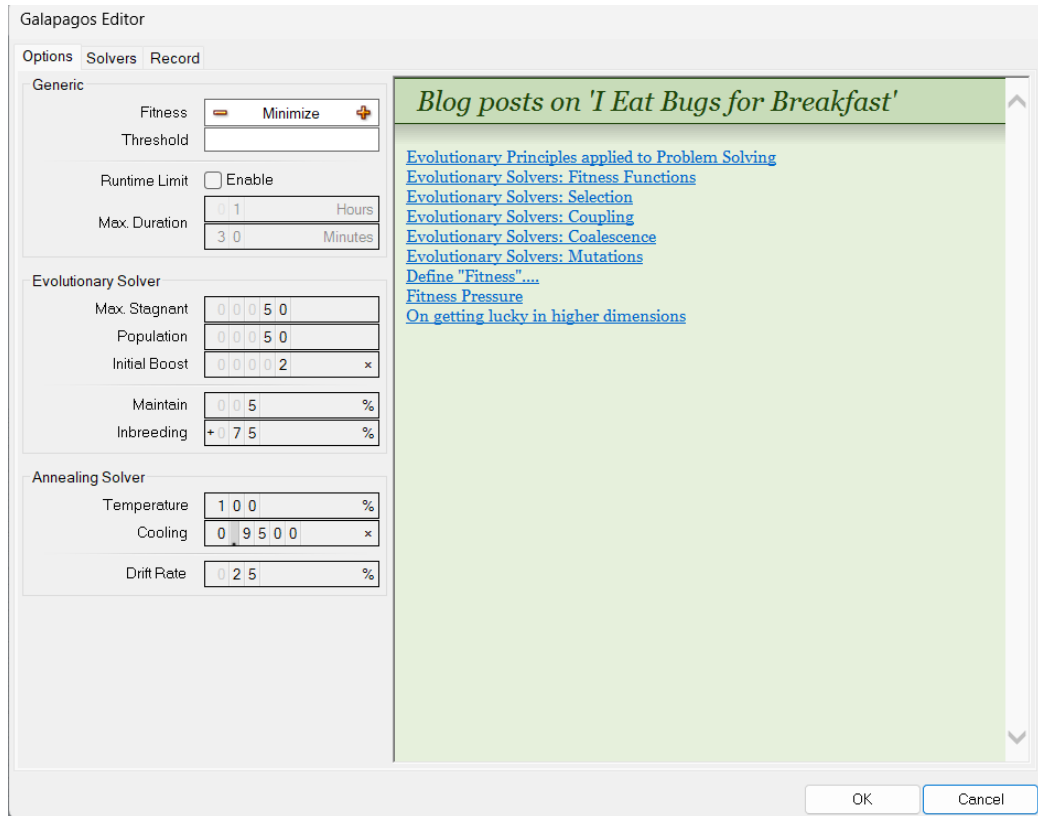


Figura 28: L'interfaccia "Options" del componente Galapagos.

Nell'interfaccia delle opzioni all'interno di Galapagos ci sono tutte quelle impostazioni che ci permettono di configurare al meglio l'algoritmo e trovare la soluzione più adatta alle nostre esigenze.

**Fitness/threshold:** Definisce se si vuole massimizzare o minimizzare il valore; come spiegato nella teoria nel capitolo terzo, è possibile farlo cambiando semplicemente il segno della "domanda". È possibile inoltre impostare una soglia: se l'algoritmo trova una soluzione con questa soglia, si ferma, se si lascia la soglia vuota, l'algoritmo continuerà all'infinito, finché non avrà provato tutte le opzioni o non avrà raggiunto un limite di tempo.

**Runtime Limit:** Impostando un limite di tempo, è possibile definire il tempo massimo di esecuzione del modello. Galapagos dispone di due tipi di algoritmi di ottimizzazione: "evolutionary" e di "annealing". Il solutore

evolutivo cerca un buon risultato e poi lo ottimizza apportando piccole modifiche ai parametri. Tuttavia, in alcuni casi, uno script è costruito con un grande grado di libertà che può portare a risultati inaspettati e utilizzare il risolutore “annealing” diventa più consono. Senza entrare troppo nello specifico con la spiegazione di ogni singola impostazione che si può configurare in questa sezione, si può determinare quanti calcoli l’algoritmo deve calcolare, quanti calcoli deve predisporre prima di progredire al passaggio successivo, come interfacciarsi con le combinazioni, ecc.

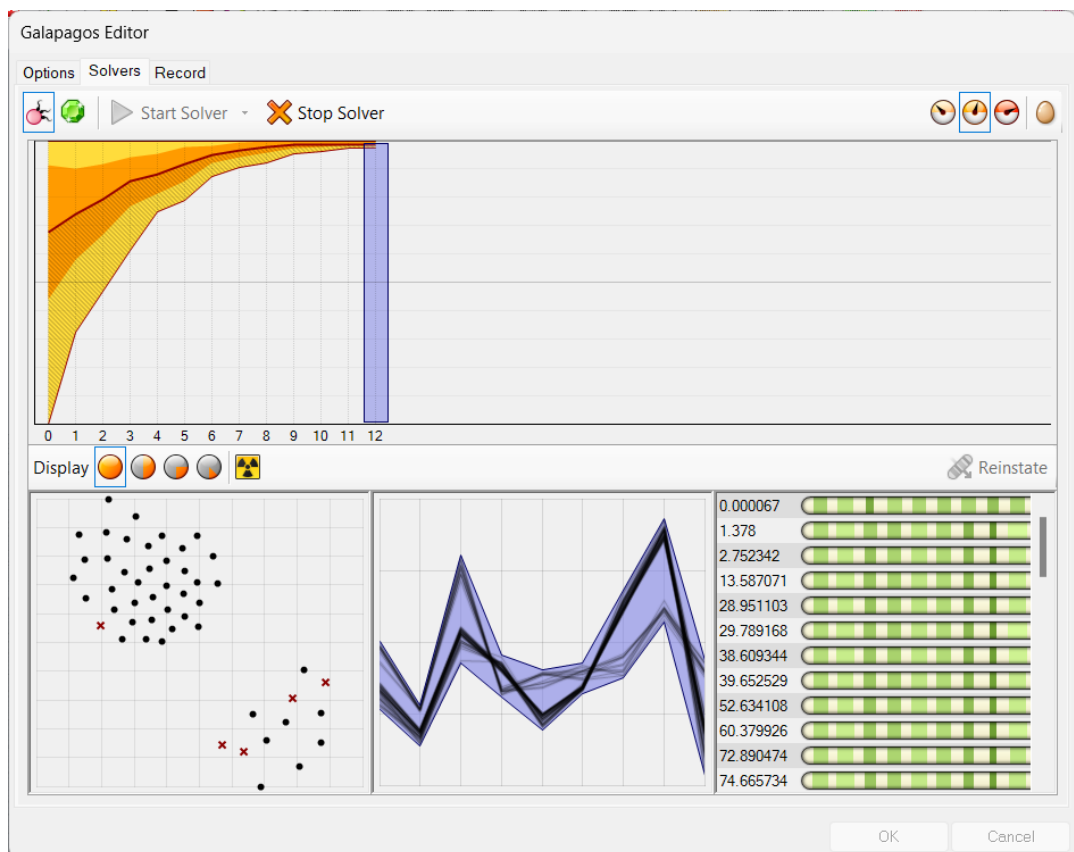
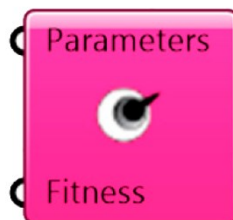


Figura 29: L'interfaccia “Solvers” del componente Galapagos

Dopo aver eseguito l'algoritmo per un certo tempo, è possibile arrestare il risolutore e visualizzare le soluzioni. Il diagramma in basso a destra della figura 29 mostra alcune delle soluzioni migliori in ordine dalla migliore alla

peggiore, mentre negli altri pannelli è possibile visualizzare le iterazioni avvenute, il miglioramento dell'algoritmo, la distanza dei punti scelti dall'algoritmo, quali punti sono stati scelti e quali no e altre informazioni secondarie.

Un altro esempio di plug-in mono-obiettivo è Silvereye, dedicato anch'esso per l'ottimizzazione come Galapagos o Octopus. Tuttavia, a differenza di questi programmi che si basano su algoritmi genetici, Silvereye esegue la routine di ottimizzazione basandosi sull'intelligenza degli sciame, la così detta "Swarm Intelligence". È facile da usare anche per i non addetti ai lavori ed è adattato alle esigenze di architetti, designer e ingegneri nelle prime fasi di progettazione.



*Figura 30: Il componente Silvereye in Grasshopper*

Per l'implementazione è stata scelta la tecnica dell'ottimizzazione a sciame, che "cerca di rendere le cose più semplici piuttosto che più complesse" (Kennedy & Eberhart, 1995). L'algoritmo implementato in Silvereye soddisfa tutti e cinque i principi fondamentali della Swarm Intelligence descritti da Millonas (Millonas, 1994). L'algoritmo PSO, "Particle Swarm Optimization", il quale è implementato nella sua forma nativa per il funzionamento del software con l'aggiunta di fattori di peso interni, introdotti per evitare che le velocità diventino troppo grandi. Come si evince dalle figure che mostrano l'editor di Silvereye non c'è molta differenza con l'interfaccia usata da Galapagos. I due plug-ins per l'ottimizzazione mono-obiettivo non differiscono nell'uso ma solamente dai metodi per calcolarlo. Va però detto che in molti studi condotti con le

stesse condizioni Silvereye abbia una velocità per raggiungere la soluzione poco più superiore con meno iterazioni (J. Cichocka, 2016).

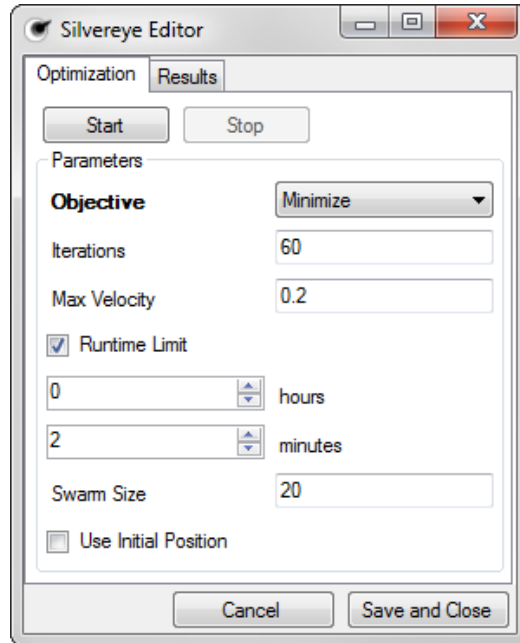


Figura 31: L'interfaccia "Optimization" del componente Silvereye

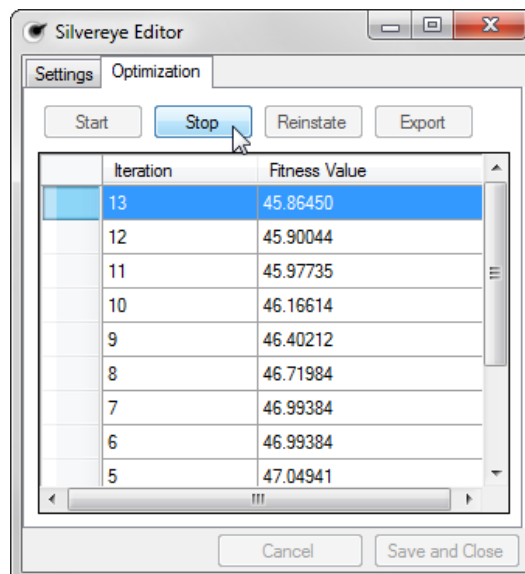


Figura 32: L'interfaccia "Results" del componente Silvereye

Si precisa che questo lavoro di tesi si concentra sull'applicazione pratica degli strumenti di ottimizzazione piuttosto che sull'analisi dettagliata dei loro complessi algoritmi sottostanti. Esplorare in profondità la complessità di tali algoritmi potrebbe risultare non solo eccessivamente complicato, ma anche fuori dal contesto dello scopo specifico della ricerca di studio. Pertanto, invece di addentrarmi nelle meccaniche interne di programmazione informatica e teoria matematica, ho scelto di citare questi software come strumenti chiave per il mio lavoro, focalizzandomi sull'implementazione e sull'ottimizzazione dei risultati ottenuti.

### 5.3.3. Strumenti e plug-ins per l'ottimizzazione multi-obiettivo

Il limite dell'approccio a obiettivo singolo è che si può far calcolare all'algoritmo una sola "richiesta" in quanto viene fissata solo una fitness. Per un problema di ottimizzazione multi-obiettivo, non esiste una soluzione unica che ottimizza l'obiettivo, ma molteplici soluzioni che soddisfano il calcolo contemporaneamente in quanto, avendo numerose fitness, è possibile che molteplici parametri e variabili soddisfino alcune condizioni, mentre diversi parametri e variabili soddisfino altre. Con questa metodologia più complicata si riesce meglio a rispecchiare un problema nella realtà pratica. In particolare, lo scopo è quello di cercare soluzioni non dominate (migliori rispetto a quelle dominate). Si dice soluzione non dominata se non c'è un'altra soluzione che abbia prestazioni migliori sotto ogni aspetto. Se si analizzano le prestazioni di un gran numero di opzioni progettuali nello spazio delle funzioni obiettivo per la valutazione, il limite esterno di questa collezione di punti definisce il limite oltre il quale il progetto non può essere ulteriormente migliorato (J. I. Zaratiegui Fernandez, 2014). Se confrontata con qualsiasi altra soluzione, una soluzione non dominata è superiore per almeno un criterio. Questa

soluzione è soluzione ottimale di Pareto<sup>10</sup>, il gruppo di soluzioni ottimali di Pareto è chiamato frontiera di Pareto o fronte di Pareto. In assenza di informazioni aggiuntive sulle preferenze soggettive, tutte le soluzioni ottimali di Pareto sono considerate ugualmente buone (R. Vierlinger, 2013).

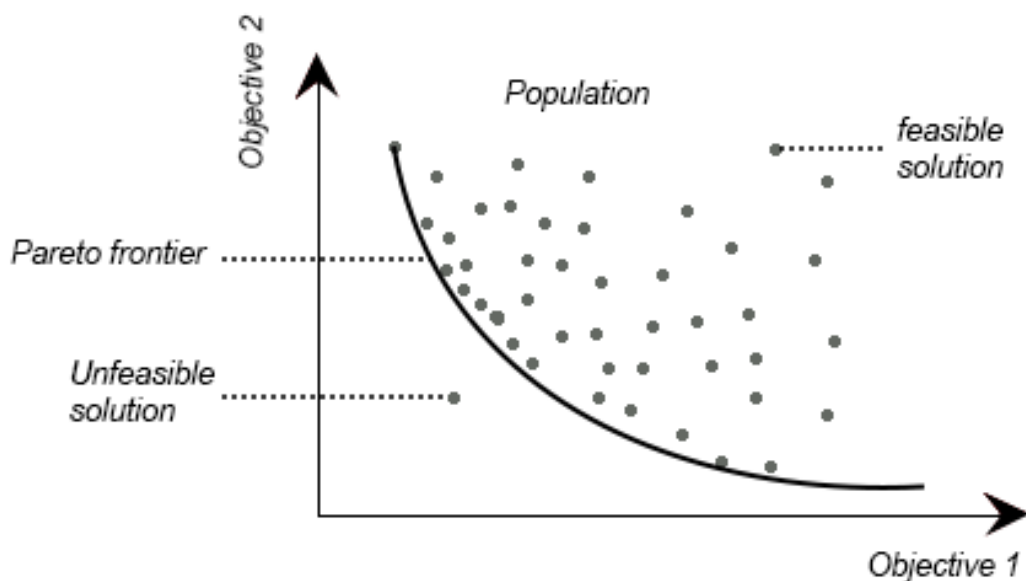


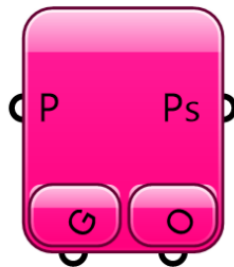
Figura 33: Schema della soluzione Pareto-ottima (J. I. Zarategui, Intelligent design objects applied to the spatial allocation problem, Master Degree, Middle East Technical University, 2014).

Come nel caso dell'ottimizzazione tramite plug-ins mono-obiettivi in Grasshoper, anche in questo paragrafo si andranno ad esplorare i principali plug-ins per l'ottimizzazione multi-obiettivo.

---

<sup>10</sup> Il concetto trae il suo nome dall'economista italiano Vilfredo Pareto (1848 – 1923). Nel caso dell'ottimizzazione multi-obiettivo, una soluzione Pareto-ottima rappresenta uno stato in cui è impossibile migliorare un criterio senza deteriorare almeno un altro. In altre parole, nessuna soluzione è migliore sotto tutti gli aspetti considerati. Questa condizione riflette il concetto di efficienza, per cui è impossibile ottenere miglioramenti in un aspetto senza sacrificarne un altro.

Octopus è stato originariamente creato per l'ottimizzazione evolutiva multi-obiettivo. Questo strumento, come gli altri che si mostreranno in questo paragrafo, consente la ricerca e il calcolo di molti obiettivi contemporaneamente, producendo una serie di soluzioni che sono prese facendo compromessi fra tutte le possibili scelte ottimizzate tra gli estremi di ciascun obiettivo scelto in precedenza. Viene utilizzato e funziona in modo simile a Galapagos e Silvereye, ma introduce il principio di Pareto per gli obiettivi multipli.



*Figura 34: Il componente Octopus in Grasshopper*

I geni, cioè le variabili del nostro sistema, devono essere collegati al campo G, mentre il campo O deve essere collegato agli obiettivi della simulazione, cioè le fitness, le quali devono essere minimo due. È possibile ma non obbligatorio collegare al plug-in anche una mesh 3D, che definisce fenotipo (P) e ciò permette di visualizzare le soluzioni nello scenario. Le variazioni del fenotipo e le soluzioni sono raccolte nell'output del componente. Una volta che il plug-in ha raccolto tutti i dati richiesti, può iniziare l'ottimizzazione. Facendo doppio clic nel mezzo del componente, si apre la finestra principale di Octopus, che fornisce un'interfaccia grafica che si può configurare. Gli individui calcolati di ogni generazione appaiono come cubi di differente colore nella vista al centro della finestra, in un spazio cartesiano definito dagli assi x-y-z; il colore rosso indica che quella soluzione è potenzialmente ottimale, mentre il colore verde dimostra la lontananza dalla frontiera di Pareto.



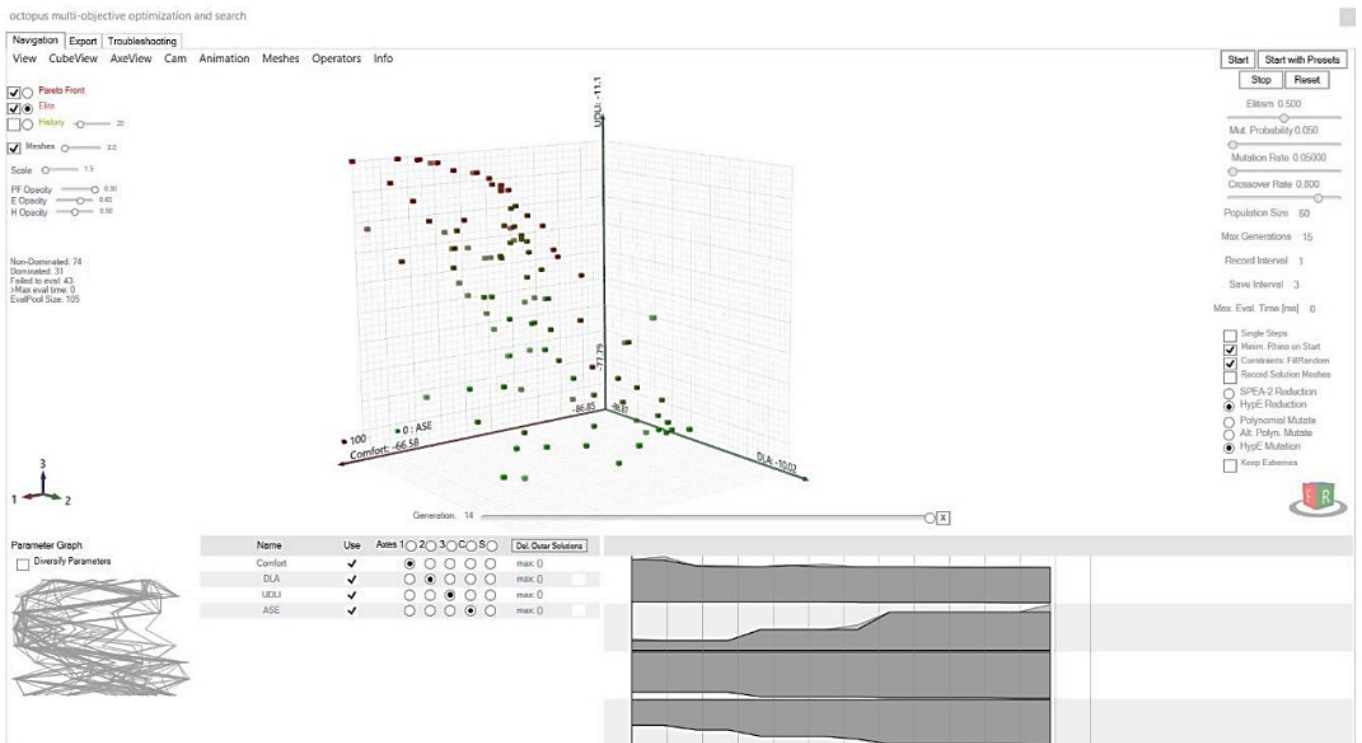
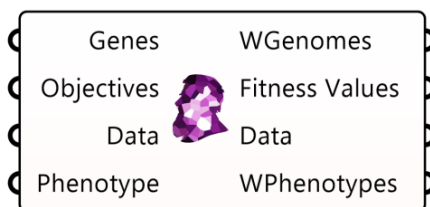


Figura 35: L'interfaccia "Navigation" del componente Octopus

Wallacei, altro esempio di plug-in per l'ottimizzazione in Grasshopper, pubblicato per la prima volta nel 2018, si basa sulla ricerca condotta da Mohammed Makki. La sua ricerca si è concentrata non tanto sul calcolo evolutivo come strumento per arrivare alla soluzione, ma maggiormente sulla ricerca di ciò che avviene ai lati dell'algorithm evolutivo; in questo caso, quindi, l'attenzione si è rivolta soprattutto sulla formulazione del problema di progettazione, sull'analisi dei risultati e sulla selezione delle soluzioni ottimizzate. Wallacei X è il componente principale della piattaforma Wallacei, grazie alla quale l'utente è in grado di eseguire tutta una serie di azioni dell'algorithm evolutivo come analizzare i risultati, eseguire metodi di selezione, esportare fenotipi, accedere al forum online, il tutto all'interno di un'unica interfaccia utente (M. Makki et al., 2019).

Wallacei X impiega l'algoritmo NSGA-2<sup>11</sup> come algoritmo evolutivo principale.



*Figura 36: Il componente Wallacei in Grasshopper*

Il Wallacei X si distingue per le sue capacità analitiche, ma anche per la sua attenzione all'usabilità e all'accessibilità da parte dell'utente. Contiene strumenti di visualizzazione e interpretazione dei dati generati dal processo di ottimizzazione che rende possibile una comprensione approfondita delle dinamiche evolutive in gioco. Gli utenti possono, quindi, visualizzare e interagire con i dati attraverso vari metodi quali grafici di dispersione, diagrammi paralleli e altri strumenti visivi che li aiutano a identificare le soluzioni più promettenti e a valutare il rapporto tra i parametri disponibili. Questa interattività crea maggiore accessibilità e favorisce scelte progettuali più basate su prove informate. Il plug-in è altamente versatile e può essere utilizzato in design architettonici, design di prodotto e innumerevoli altre discipline e campi di ricerca.

Per aprire l'interfaccia utente di Wallacei X basta inserire i dati minimi per far funzionare il plug-in, cioè collegare i geni e gli obiettivi e poi fare doppio clic sul componente (l'interfaccia utente non si apre se i geni e gli obiettivi non vengono inseriti nel componente).

---

<sup>11</sup> L'algoritmo segue lo schema generale di un algoritmo genetico con una selezione modificata dell'accoppiamento e della sopravvivenza. In NSGA-II, in primo luogo, gli individui vengono selezionati frontalmente. In questo modo, si verifica una situazione in cui un fronte deve essere diviso perché non tutti gli individui possono sopravvivere. In questo fronte di divisione, le soluzioni vengono selezionate in base alla distanza di affollamento (K. Deb et al., 2002).

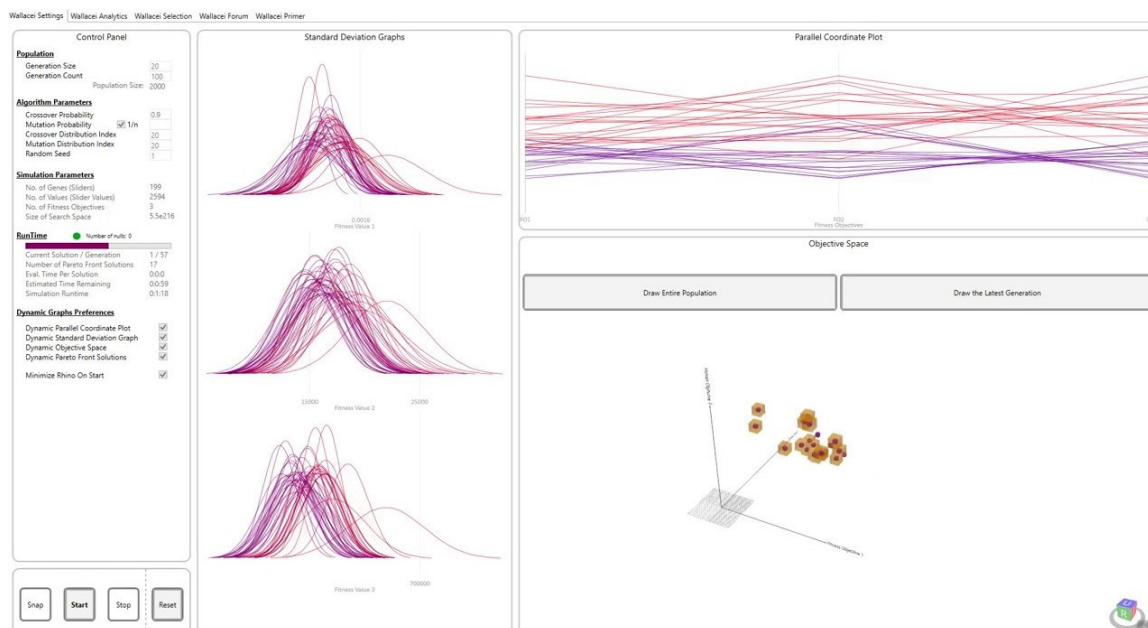


Figura 37: L'interfaccia "Settings" del componente Wallacei

Nell'interfaccia principale mostrata nella figura 37 è possibile configurare le opzioni dell'algoritmo per la ricerca della soluzione desiderata, procedendo ugualmente come negli altri plug-ins visti in precedenza. Sul pannello di sinistra (Control Panel) è possibile controllare l'avvio e lo stop del programma, calibrare a proprio piacimento i dati del problema come la popolazione e i parametri e, inoltre, monitorare l'avanzamento, una volta avviato, del processo. Negli altri tre pannelli (Standard deviation graphs, Parallelo ordinate plot e Objective space) è possibile monitorare "live" altre principali informazioni quali i grafici per ogni obiettivo durante la simulazione, le coordinate parallele di tutte le generazioni create e lo spazio obiettivo, cioè vedere disegnato tutte le possibili soluzioni in uno spazio cartesiano (con la possibilità di filtrare le soluzioni ultime).

L'ultimo strumento considerato per l'ottimizzazione multi-obiettivo è Opossum, il quale include due dei migliori algoritmi di ottimizzazione monobiettivo di Grasshopper: RBFOpt basato su modelli e CMA-ES evolutivo e, inoltre, l'algoritmo multi-obiettivo RBFMOpt e gli algoritmi multi-obiettivo MACO (Ant Colony), MOEA/D, NSGA-II e NSPSO (Particle Swarm) della libreria Pygmo 2 (T. Wortmann, 2017). Opossum è uno strumento di ottimizzazione basato su modelli, finalizzato all'ottimizzazione della progettazione architettonica e applicabile in

particolare a problemi che comportano simulazioni di prestazioni che richiedono molto tempo (T. Wortmann, 2017).



*Figura 38: Il componente Opossum in Grasshopper*

Per rendere questa complessità accessibile anche ai non esperti, l'interfaccia grafica di Opossum è composta da tre schede che consentono livelli crescenti di controllo: la prima scheda consente di scegliere tra minimizzazione e massimizzazione, di selezionare uno dei tre pre-insiemi di parametri e di avviare e interrompere l'ottimizzazione. Le preimpostazioni (Fast, Extensive e Alternativa) si basano su test intensivi con funzioni di prova matematiche. "Fast" esegue MSRSM con un algoritmo genetico. "Estensivo" è identico, ma dedica più tempo alla ricerca del modello. "Alternativa" esegue Guttman, che funziona bene in certi casi. Nella prima scheda viene visualizzato anche un grafico di convergenza animato per informare gli utenti sull'avanzamento dell'ottimizzazione. La seconda scheda consente di definire le condizioni di arresto in base al numero di iterazioni o al tempo trascorso e di eseguire e registrare più ottimizzazioni. La terza scheda accetta i parametri della riga di comando per RBFOpt. Se lo si desidera, questa finestra "esperta" permette all'utente di avere il pieno controllo, con i parametri inseriti qui che sovrascrivono i parametri impostati dalle prime due schede. In Grasshopper, Opossum segue l'aspetto e il comportamento degli strumenti di ottimizzazione esistenti, comprese le convenzioni relative ai colori dei componenti di ottimizzazione e alle loro connessioni con le variabili.

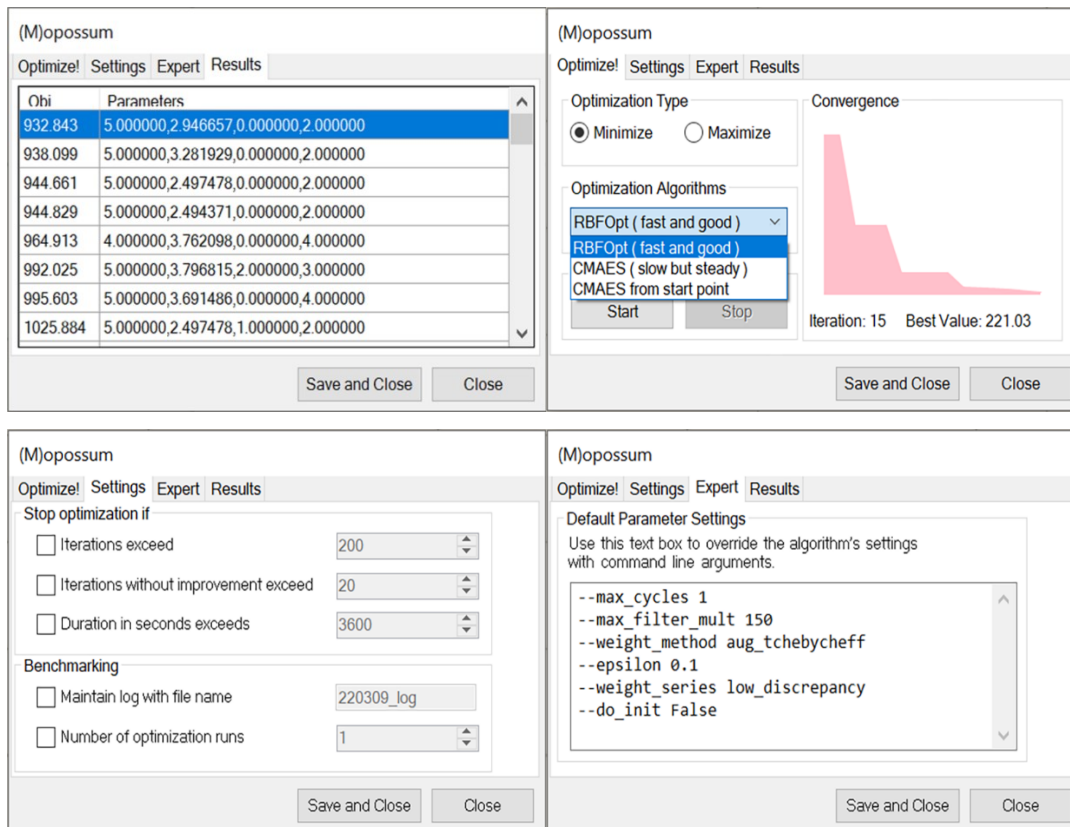


Figura 39: L'interfaccia del componente Opossum in Grasshopper

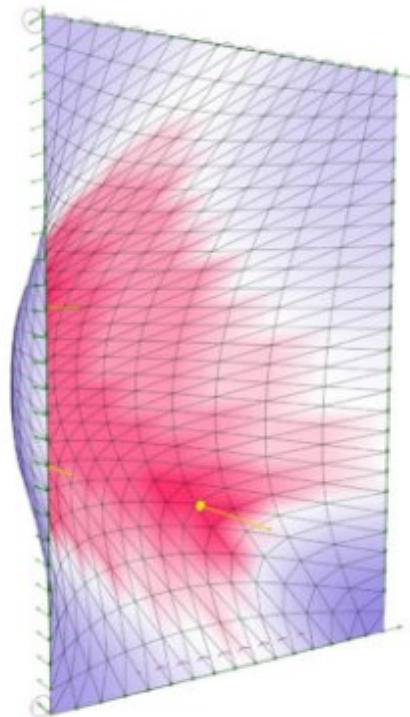
### 5.3.4 Integrazione di Karamba3D nei processi di progettazione parametrica

Il plug-in Karamba è stato utilizzato durante lo sviluppo del caso studio (capitolo sesto) e ricopre un'importante parte dello studio svolto fino ad ora. Pertanto è necessario descriverne brevemente il suo utilizzo in campo parametrico. Il plug-in Karamba3D, lanciato nel 2011 da Clemens Preisinger e Andreas Bastian, rappresenta uno strumento innovativo per l'analisi strutturale all'interno dell'ambiente di modellazione parametrica di Grasshopper per Rhinoceros. Karamba3D si distingue per la sua capacità

di integrare l'analisi strutturale direttamente nel flusso di lavoro parametrico, consentendo agli utenti di esplorare e ottimizzare progetti architettonici e ingegneristici con una precisione senza precedenti. Questo plug-in è particolarmente utile per il design di strutture complesse e non convenzionali, offrendo un'interfaccia che permette di effettuare analisi strutturali avanzate in modo intuitivo e interattivo. Karamba è un programma agli elementi finiti (FE) per la previsione del comportamento delle strutture sottoposte a carichi esterni. Sebbene sia stato sviluppato in uno studio di ingegneria strutturale, il suo obiettivo principale non è questo. È orientato all'uso in un ambiente di progettazione architettonica interattivo e parametrico. Uno degli obiettivi era quello di creare uno strumento veloce e leggero che facilitasse il flusso di dati tra modelli strutturali e geometrici. Inizialmente, la necessaria codifica delle parametrizzazioni geometriche come script si è rivelata macchinosa e ha ridotto la gamma dei possibili utenti a una ristretta cerchia di esperti di scripting. Per risolvere questo problema, il programma è stato inserito in un software esistente che offriva un ambiente di modellazione interattivo e generativo. Grasshopper consente l'integrazione di plug-in personalizzati che possono utilizzare gli elementi dell'interfaccia grafica del programma e quindi interagire con altri plug-in in un modo ben definito. Questo ha portato alla prima versione pubblica del plug-in Karamba. La maggior parte degli aspetti di un modello Karamba può essere resa dipendente da input parametrici. Il vero valore aggiunto di Karamba3D risiede nella sua integrazione fluida con Grasshopper, il che consente una interazione dinamica tra modellazione e analisi. Gli utenti possono modificare le geometrie e i parametri di progetto e osservare immediatamente come queste variazioni influenzano le performance strutturali. Questa interattività facilita un processo iterativo di progettazione e verifica, ottimizzando le soluzioni strutturali in tempo reale. Un modello di questo tipo consiste in componenti visivi che hanno l'aspetto dei blocchi del software nativo. Il fatto che Karamba reagisca immediatamente a qualsiasi cambiamento dei parametri di ingresso aiuta a comprendere i meccanismi

strutturali del progetto. Utilizzare Karamba è come guardare un film rispetto a un'immagine fissa; si può creare facilmente una serie di immagini in tempo reale manipolando il widget dell'interfaccia utente corrispondente invece dell'approccio più tradizionale che prevede una risposta del sistema solo a un particolare stato (C. Preisinger, (2013)).

In sintesi, Karamba3D rappresenta un avanzato strumento di analisi strutturale che, integrandosi con Grasshopper, offre nuove possibilità nel campo della progettazione parametrica. La sua capacità di combinare modellazione, analisi e ottimizzazione all'interno di un'unica piattaforma consente agli utenti di affrontare le sfide progettuali con maggiore efficacia e precisione. L'adozione di Karamba3D può portare a una significativa innovazione nel design strutturale, facilitando la creazione di soluzioni più efficienti e creative.



*Figura 40: Esempio di utilizzo del plug-in Karamba3D in Grasshopper. I punti arbitrari di un guscio vengono utilizzati per definire i carichi perpendicolari al guscio. Al guscio viene assegnata anche un'unica proprietà del materiale ([//karamba3d.com/](http://karamba3d.com/)).*





# 06 • L'ottimizzazione del Toshiba IHI Pavilion: il caso studio

Nel vasto panorama dell'architettura contemporanea, il Toshiba IHI Pavilion a Expo '70 emerge come un monumento molto interessante dalla visione futuristica. Questo capitolo si propone di delineare con precisione il contesto storico in cui questo edificio architettonico ha preso forma, immergendosi nelle problematiche del periodo in cui è stato concepito. Il periodo è elemento essenziale perché avrebbe plasmato il destino del padiglione. L'analisi approfondita della vita e della filosofia del suo progettista, Kisho Kurokawa, e del Movimento Metabolico, con il quale Kurokawa è intimamente legato, sarà il filo conduttore che guiderà l'esplorazione. Si affronterà poi la complessità della struttura e della geometria dell'IHI Pavilion. Utilizzando come strumento il software Grasshopper di Rhinoceros, si costruisce il processo di ricreazione virtuale del padiglione. L'obiettivo non è solo riprodurre digitalmente l'architettura in forma parametrica tramite algoritmi, ma anche ottimizzarla, sottolineando l'intersezione tra creatività umana e potenzialità computazionali. Infine, si condurrà uno studio approfondito delle soluzioni ottenute attraverso l'applicazione di algoritmi genetici, mettendo in luce le scelte e le sfide affrontate nel processo di ottimizzazione.



## 6.1. Il contesto storico dell'Expo del '70 a Osaka, Giappone

Per studiare in maniera approfondita il contesto storico in cui è nato il Toshiba IHI Pavilion di Kisho Kurokawa bisogna iniziare ben prima della sua costruzione per poi giungere all'Expo del 1970.

Dopo la Seconda Guerra Mondiale, le principali città giapponesi furono distrutte e gran parte delle aree residenziali e delle fabbriche industriali dovettero essere ricostruite. Fu allora che fornire rifugi e strutture urbane alle persone divenne una priorità. Con l'espansione dell'economia, il Giappone iniziò a occidentalizzare gran parte della propria produzione per competere con gli altri Paesi sviluppati. Nel 1948 ebbe così inizio un nuovo periodo per il Giappone il quale avviò un nuovo metodo di ricostruzione. È vero infatti che dopo qualche anno dalla Seconda Guerra Mondiale il Giappone ebbe un grande cambiamento nella politica economica: il paese divenne il principale fornitore di materiali agli Stati Uniti, oltre a diventare l'importante base strategica degli americani in Estremo Oriente (come baluardo contro il blocco comunista che stava cercando di espandersi in Asia). Il Giappone iniziò un'imponente opera di modernizzazione della base industriale e di assimilazione delle tecnologie provenienti dall'estero. I principali eventi e progetti significativi che hanno avuto un ruolo effettivo nell'architettura del Giappone si sono verificati negli anni successivi alla seconda guerra mondiale e si possono riassumere con il seminario di Konrad Wachsmann (1955), che ha riunito numerosi architetti giapponesi ed è stato il primo passo verso l'occidentalizzazione dell'architettura giapponese, la partecipazione di Kenzo Tange alla riunione del CIAM in Olanda (1959) e la World Design Conference che si è svolta nel 1960 a Tokyo, che fu una grande occasione per gli architetti e i designer giapponesi di promuovere i loro lavori e le loro idee, nonché di incontrare e scambiare esperienze e opinioni con professionisti e colleghi stranieri (vi parteciparono 84 progettisti stranieri provenienti da 26 Paesi). Questi avvenimenti sono

stati il punto di partenza di nuovi movimenti e ideologie internazionali e nazionali come il Metabolismo (del quale si parlerà in modo approfondito nei paragrafi successivi in quanto tema fondamentale per comprendere l'architettura giapponese), che portarono gradualmente gli architetti giapponesi a iniziare ad esprimere il proprio stile grazie alle opportunità offerte dagli eventi internazionali con i Giochi olimpici giapponesi. I Giochi Olimpici di Tokyo del 1964, infatti, furono un'occasione unica per convalidare la rinascita nazionale dalla distruzione della guerra e il suo ritorno alla comunità internazionale. Inoltre furono una piattaforma per gli architetti "metabolisti" per mostrare i loro nuovi concetti in progetti reali .

Nel 1970 il Giappone è stato scelto come sede dell'esposizione mondiale a Senri Hills, a Osaka. L'esposizione, secondo le intenzioni del Giappone, si sarebbe dovuta tenere addirittura nel 1940, ma fu cancellata con l'avvento della guerra e tuttavia vendette all'epoca già numerosi biglietti (circa un milione) che furono poi utilizzati ben trenta anni dalla loro vendita durante l'Expo ad Osaka. Il tema scelto dell'edizione del 1970 era "Progresso e armonia per l'umanità". Furono invitati più di dodici architetti per la progettazione di parti fondamentali distinte. Le esposizioni mondiali sono da sempre state utilizzate come occasioni di rinnovamento materiale e sociale, attraverso il cambiamento dell'ambiente costruito e la motivazione delle persone ad adottare e apprezzare il progresso delle tecnologie. Sostenuta dal governo, l'Esposizione Universale del 1970, o Expo'70, propose e spianò la strada alle future città e allo sviluppo regionale. Nel 1958 un tratto di terreno agricolo senza pretese e una foresta di bambù selvatici nel distretto di Suita furono scelti come luogo per l'esposizione mondiale del 1970. Expo'70 fu costruita su un sito di 815 acri e durò 183 giorni, dal 15 marzo al 13 settembre. L'Expo'70 rappresentò una significativa deviazione dal corso delle esposizioni mondiali. Degna di nota perché la prima delle esposizioni mondiali a tenersi in Oriente, fu anche un evento eccezionale che attirò oltre 64 milioni di visitatori, la più grande partecipazione a un'esposizione internazionale dalla sua nascita più di un secolo e mezzo

prima. L'affluenza è stata significativa anche perché più del 90% dei visitatori erano giapponesi, che sono stati ispirati e incoraggiati a viaggiare da ogni parte del Paese per vedere ciò che il mondo aveva da offrire. Rappresentativo del pensiero dell'esposizione è sicuramente il logo<sup>12</sup> di Expo '70:



*Figura 41: Il logo dell'EXPO '70 a Osaka, Giappone (l'immagine è tratta da S. Köhn, Presentation for the Japanology seminar "Resistance in Japan", University of Cologne, 2 Nov 2021).*

---

<sup>12</sup> L'emblema dell'Expo'70 di Osaka è una rappresentazione simbolica ricca di significato, volta a incarnare il tema "Progress and Harmony for Mankind": il disegno centrale è un fiore di sakura, la fioritura di ciliegio, che è emblematico del Giappone. Ogni petalo di questo fiore rappresenta uno dei cinque continenti, simboleggiando la speranza che il mondo si unisca e partecipi a questo evento globale. Al centro del fiore si trova un cerchio, che rappresenta il Hinomaru, la bandiera nazionale del Giappone, simbolizzando l'orgoglio e l'identità nazionale. Le aree bianche circostanti suggeriscono spazio per lo sviluppo e il progresso, riflettendo una visione di un futuro in evoluzione. L'intera composizione emana un senso di stabilità e dignità, esprimendo l'ambizione di realizzare un evento di livello mondiale che promuova l'armonia e la cooperazione internazionale. Questo simbolismo visivo sottolinea l'impegno del Giappone nell'accogliere il mondo con apertura e spirito di collaborazione, aspirando a un futuro condiviso di progresso.

Expo'70 fu anche un evento in cui le nazioni non giapponesi si rivolsero alla popolazione, alle aziende e al governo giapponesi per ottenere vantaggi politici o economici. Nonostante l'intenzione dell'esposizione di promuovere il tema del progresso e dell'armonia per l'umanità, le due superpotenze internazionali dell'epoca, gli Stati Uniti e l'Unione Sovietica, usarono l'esposizione come palcoscenico per mostrare la loro influenza globale e il loro modo di vivere. L'architettura è stata uno strumento importante sia per gli organizzatori che per i produttori dell'Expo'70. Infatti la tensione tra questi due Paesi è testimoniata da una battaglia architettonica: l'Unione Sovietica creò un enorme pinnacolo bianco per trasmettere la potenza e la forza del socialismo come forza politica significativa sulla scena mondiale. A questo si contrapponeva il padiglione americano, situato sul lato opposto dell'area espositiva, creato da un'unica membrana gonfiabile che assume la forma architettonica di una superficie elegante e ondulata a basso profilo (S. Kalantari, et al., 2017).

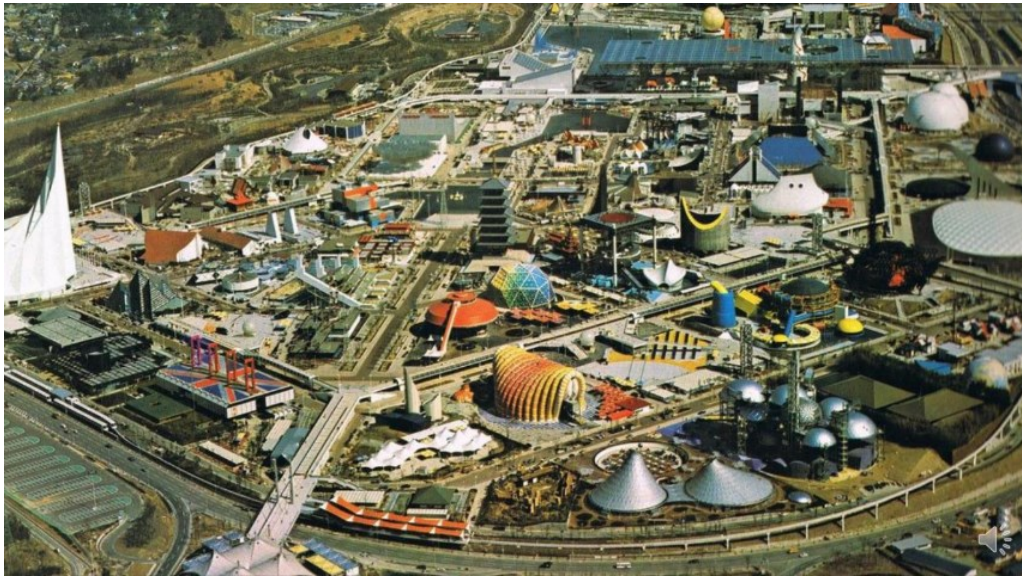
L'esposizione del 1970 prese come titolo *Progresso e armonia per l'umanità*. Esso sembra indicare una soluzione a problemi di enorme entità. Tuttavia, se da un lato il titolo dell'esposizione allude a una serie di obiettivi positivi, dall'altro tiene nascoste le sue basi socio-politiche. È vero infatti che il Giappone degli anni Sessanta non era del tutto favorevole ai cambiamenti che stavano avvenendo all'interno della società e l'Expo'70 può essere interpretata come un riorientamento della coscienza pubblica giapponese. I cambiamenti della società che avvennero durante quegli anni hanno portato alla luce diverse questioni sociali. L'opinione politica dell'epoca riconobbe che, pur essendo vantaggiosa in alcune aree, la marcia in avanti della tecnologia portava con sé anche una grande capacità di distruzione e disarmonia. Per esempio, la ritrovata forza economica del Giappone del dopoguerra era in parte dovuta al posizionamento come centro manifatturiero mondiale e questo aveva un costo per la società, poiché l'inquinamento industriale colpiva duramente l'opinione pubblica. Un ulteriore esempio di come il

Giappone fu scosso in quegli anni furono le violente rivolte studentesche che, iniziate già i primi anni Sessanta, culminarono nel 1968 con le aggressive proteste contro le politiche americane sulla guerra del Vietnam. Gli scontri degli anni Sessanta furono frequenti, violenti e, forse più significativamente, enormemente dannosi per l'armonia sociale della nazione. Il Giappone arrivò all'organizzazione dell'esposizione mondiale in un clima di forte disarmonia, contrasto e attrito. La nazione era allo stesso tempo percorsa dal desiderio di cambiare e alla disperata ricerca di una forma di solidità, di permanenza o di consolidamento. L'Expo'70 cercò di mettere in una luce positiva il progresso per unire il popolo giapponese e ricercare una pace sociale fra la popolazione (T. J. Moleta, 2010).

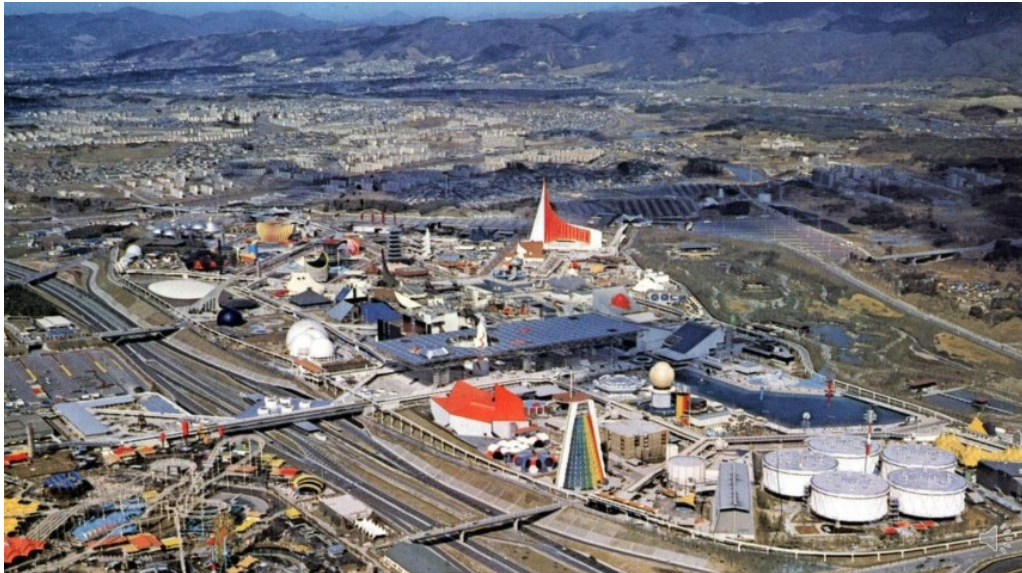
Per la creazione dell'esposizione dell'Expo '70 gli organizzatori presero spunto dalle edizioni precedenti con l'obbiettivo di caratterizzarla come un evento a tema. La mostra fu pianificata in modo tale che garantisse la realizzazione di un marchio, che potesse essere identitaria, che potesse interpretare o addirittura accelerare i cambiamenti in atto e che potesse concepire nuovamente il *cambiamento* come una nozione di *progresso* coscienzioso. Gli organi direttivi utilizzarono l'esposizione come un'opportunità per presentare il progresso come attrattivo e per educare le classi medie giapponesi a ricercarlo nei nuovi prodotti, servizi e tecnologie. In questo modo il significato di progresso poteva associarsi all'idea che potesse portare all'armonia che all'epoca mancava e i giapponesi ricercavano. In questo modo il tema scelto rispondeva al clima sociale dell'epoca. Il comitato della mostra avviò una discussione che mise in mostra il progresso, incorporando architettura, design e tecnologia. I padiglioni dell'Expo'70 si impegnarono al massimo nella rappresentazione di aziende fiorenti e avanzate. Le aziende giapponesi, inoltre, investirono ingenti somme di denaro per la creazione dei padiglioni in quanto la mostra fu usata dal governo giapponese come piattaforma per rientrare nel mercato internazionale promuovendo e sponsorizzando la partecipazione: in cambio le aziende ottennero dal

governo stesso grandi stanziamenti e sussidi. Il mondo dell'Expo'70 che i visitatori incontrarono fu un'esperienza vibrante e multisensoriale, meticolosamente realizzata utilizzando colori, proiezioni, suoni digitali, prefabbricazione e infinite quantità di plastica, nylon, vinile e acetato. Le città di plastica del futuro presentavano un mondo che mescolava abilmente svago, intrattenimento e comodità con luci al neon, trasporti veloci, videotelefonati e articoli per la casa inimmaginabili (S. Kalantari, et al., 2017). Durante Expo'70 furono riuniti per questo intento un gruppo di architetti e designer che crearono una serie di soluzioni architettoniche per affrontare i problemi urgenti della società. Questi architetti avevano il compito di progettare i padiglioni e miravano a rappresentare la forma e la direzione di un Giappone moderno attraverso il loro lavoro. Il prodotto finale era uno stile architettonico completamente nuovo in cui forma, contenuto, struttura, illuminazione, materiali e strategie erano tutti pronti per essere riconfigurati e reinterpretati. L'ambiente costruito dell'Expo'70 ha abbracciato materiali, tecniche e meccanismi precedentemente inesplorati come l'idraulica e la plastica, che sono stati utilizzati in volumi e metodi senza precedenti.





*Figura 42: Vista aerea di Expo '70 a Osaka, Giappone (l'immagine è tratta da S. Köhn, Presentation for the Japanology seminar "Resistance in Japan", University of Cologne, 2 Nov 2021).*



*Figura 43: Vista aerea di Expo '70 a Osaka, Giappone (l'immagine è tratta da S. Köhn, Presentation for the Japanology seminar "Resistance in Japan", University of Cologne, 2 Nov 2021).*

## 6.2. Kisho Kurokawa

Kisho Noriaki Kurokawa, nato a Nagoya, in Giappone, nel 1934 e venuto a mancare a Tokyo nel 2007, è stato uno dei principali architetti giapponesi del secolo scorso. Ha influenzato con il suo pensiero e la sua filosofia l'architettura giapponese i cui valori ancora oggi si possono riscontrare nell'ampia eredità sia architettonica che culturale lasciata. Dopo essersi laureato all'Università di Kyoto, ha studiato con l'architetto di fama mondiale Shin Isozaki all'Università di Tokyo, all'epoca al centro dell'architettura giapponese. Nel 1959 prende parte alla World Design Conference e fonda da giovanissimo, insieme a Kenzo Tange ed altri suoi colleghi, il Movimento Metabolista. All'età di 26 anni è stato il più giovane membro del movimento. Dopo circa due anni dalla World Design Conference apre il proprio studio di architettura "Kisho Kurokawa Architect & Associates" che è ancora ad oggi operativo, nonostante la morte di Kurokawa. È possibile visionare i loro lavori e studiare i valori principali a cui si basa lo studio<sup>13</sup>. Kisho Kurokawa, oltre che per i numerosi progetti da lui ideati, fu importante anche come scrittore tanto da vincere anche premi come il Japan Grand Prix of Literature per il libro *Philosophy of Symbiosis* del 1987.

Il pensiero di Kisho Kurokawa si basa sull'unione dei concetti che stanno alla base della tradizione giapponese con l'avanguardia tecnologica contemporanea, in riferimento a una profonda riflessione filosofica, in quanto Kurokawa ebbe un'elevata istruzione e una puntuale conoscenza del mondo buddista in cui fu educato. L'approccio di Kurokawa all'architettura si basa sull'idea che la filosofia comprende tre concetti fondamentali: metabolismo, metamorfosi e simbiosi. Kurokawa ritiene che la "simbiosi" sia un'altra modalità concettuale dell'architettura giapponese che incarna contemporaneamente idee apparentemente contrastanti come principi universali e differenze regionali, sensibilità occidentale e orientale, storia e futuro, piccola e larga scala, identità culturale e tecnologia moderna. Le idee

---

<sup>13</sup> [www.kisho.co.jp](http://www.kisho.co.jp)

apparentemente contrastanti non si annullano secondo Kurokawa, ma anzi possono coesistere e anche mantenere ognuna i propri significati e radici. I suoi progetti architettonici su larga scala, sia costruiti che non costruiti, sono simili ai suoi scritti teorici sull'architettura nell'era postmoderna e riflettono i suoi tentativi di definire le caratteristiche fondamentali dell'architettura giapponese. Viene portata avanti l'idea filosofica che esiste la coesistenza di differenti valori e tradizioni culturali (Y. H. Lee, 2015) .

Questa filosofia della simbiosi si mostra anche all'interno dell'architettura metabolista. Essa infatti, non essendo un'architettura ferma e pietrificata nel tempo una volta progettata e costruita, è un'architettura che continua a crescere e a trasformarsi nel tempo in evoluzione (K. Kurokawa, 1997).

L'edificio costruito da Kisho Kurokawa più significativo, dove si vedono maggiormente i segni del metabolismo e della sua filosofia di architettura, è la Nagakin Capsule Tower. La torre utilizza tecnologie e materiali all'avanguardia ed è costruita con unità prodotte in serie con un nucleo centrale per collegamenti ad alta tensione per affrontare i problemi delle popolazioni urbane contemporanee; allo stesso tempo, però, le singole stanze all'interno della torre fanno riferimento al tradizionale stile Sukiya, basandosi sulla disposizione della sala da tè; l'interno in plastica stampata incarna semplicità, ordine ed economia nel design di base della sala da tè così come il passaggio dalla strada all'alloggio individuale può essere descritto come una sorta di isolamento e fuga. È possibile dire in sintesi che Metabolismo e Simbiosi nell'architettura e negli scritti di Kurokawa si sono tradotti nella progettazione formale di capsule ed altri elementi integrando concetti organici, tradizionali e tecnologici creando progetti che riflettevano l'idea di edifici come organismi in crescita e cambiamento. Infatti Kurokawa utilizzava forme fluide e innovative, spesso ispirate alla natura (Y. H. Lee, 2015).

### 6.2.1. La filosofia del Movimento Metabolista: una visione organica

Il movimento metabolista, emerso nella metà del XX secolo, si configura come un'importante corrente architettonica e urbanistica che ha rivoluzionato il modo di concepire e progettare gli spazi urbani. Fondato inizialmente da un gruppo di architetti giapponesi che si stavano preparando per partecipare alla World Design Conference, tra cui Kenzo Tange e Kisho Kurokawa, il Metabolismo trae ispirazione dalla biologia e propone un approccio organico e adattabile alla crescita e al cambiamento delle città. Nel 1960 Kisho Kurokawa annunciò quello che viene definito come il manifesto metabolista: proposta per un nuovo urbanesimo. Egli scrive: "Metabolismo è il nome di un gruppo che cerca di offrire una visione concreta della società che verrà. Noi postuliamo la società umana come un processo di sviluppo cosmico dall'atomo alla nebulosa (K. Kurokawa, 1994). Usiamo il termine biologico "Metabolismo" perché per noi progettazione e tecnologia non sono nient'altro che estensioni del potere vitale dell'uomo. Per questa ragione, non accettiamo semplicemente il metabolismo della storia come naturale ma cerchiamo di svilupparlo attivamente". Il Metabolismo non era solo una teoria architettonica volta a risolvere i problemi di sovrappopolazione dell'affollato Giappone durante il periodo dell'espansione economica, ma anche una teoria critica della società analizzata da una prospettiva architettonica. Il termine Metabolismo induce, inoltre, a far riferimento alla biologica, la quale sostituisce la metafora meccanica dell'architettura moderna ortodossa. La teoria fondamentale che sta alla base di questo movimento è la concezione della città come un organismo vivente, capace di crescere, adattarsi e rigenerarsi nel corso del tempo. Il Metabolismo paragona gli edifici e le città allo sviluppo naturale della vita: i cicli di cambiamento e la costante rigenerazione e distruzione del tessuto organico. Gli architetti metabolisti si sono concentrati sulla creazione di strutture modulari e flessibili, progettate per essere aggiunte o modificate in risposta alle

esigenze evolutive delle comunità urbane (M. Masquelier, 2022). Il movimento Metabolista, inoltre, non punta tanto a creare forme e stili – i quali sono manifestazione di un periodo ben preciso e quantificabile nella storia -, ma a esprimere una filosofia e una nuova comprensione dell'architettura. Altro aspetto fondamentale della filosofia metabolista è la tecnologia, che nel mondo orientale è un'estensione dell'umanità stessa e, in ogni caso, deve essere al servizio delle nuove trasformazioni. Proprio grazie alla tecnologia, come in biologia così anche in architettura, l'uomo è in grado di modificare le parti che subiscono processi di cambiamento, ottenendo una certa flessibilità di fronte ai fenomeni di crescita, metamorfosi e morte comuni agli organismi viventi, rendendo più facile per l'uomo controllare il proprio habitat che, con aggiunte o sostituzioni, può essere adattato ai propri desideri o a circostanze specifiche (E. Martín-Gutiérrez, 1990). L'obiettivo non si limita a implementare sistemi industrializzati per la creazione di spazi economici e prodotti in serie, ma mira altresì a manifestare nuovi stili di vita e a adeguarsi a tali cambiamenti. La visione metabolista ha influenzato significativamente la pianificazione urbana, promuovendo concetti di sostenibilità, efficienza e resilienza, divenendo così un punto di riferimento nell'evoluzione del pensiero architettonico contemporaneo (M. Masquelier, 2022).

### 6.3. Toshiba IHI Pavilion: design architettonico e visualizzazione grafica tramite algoritmo in Grasshopper

Prima di addentrarci nell'analisi dettagliata di questo edificio, è interessante considerare un'interpretazione visiva che cattura i suoi caratteri unici. L'artista Kusuma Affandi ci offre uno sguardo personale su questa struttura architettonica. Nel dipinto che segue, scopriamo come il pittore ha trasceso la mera rappresentazione dell'edificio, imbattendosi in sfumature emotive e atmosfere che si intrecciano con la magnificenza dell'edificio stesso. Con il tema Progresso e Armonia per l'Umanità,

l'Expo'70 in Giappone, questa mega-fiera sostenuta da un gruppo di architetti tra i più visionari e significativi della storia, è stata per Affandi una festa per gli occhi e per l'anima, considerando la sua vita semplice in una zona rurale del Paese. Affandi riesce a catturare l'impressione del padiglione imponendo la scala e la grandezza dell'insolita struttura. Nell'interpretazione di Affandi, lo sguardo si abbassa per seguire la macchia d'erba scura sul lato destro della tela, che ci porta a trovare la struttura del padiglione, che si ancorerà al suolo. La raffigurazione di figure umane ha accentuato ulteriormente lo stupore di Affandi per la grandiosità del padiglione, in quanto esse sono state messe in ombra dalle dimensioni gigantesche della costruzione in questa composizione festosa ma intima. Dipinta spontaneamente nella semplicità delle linee, una folla di visitatori si sta dirigendo verso la cupola centrale, in attesa di ciò che incontreranno o sperimenteranno quando vi entreranno. L'Expo'70 - Padiglione Toshiba-IHI è stato probabilmente uno dei luoghi preferiti dall'artista durante la fiera. Sebbene la scena fosse limitata all'intimo ritratto del padiglione Toshiba-IHI, Affandi celebra la progressione dell'umanità attraverso la sua straordinaria visione e interpretazione.

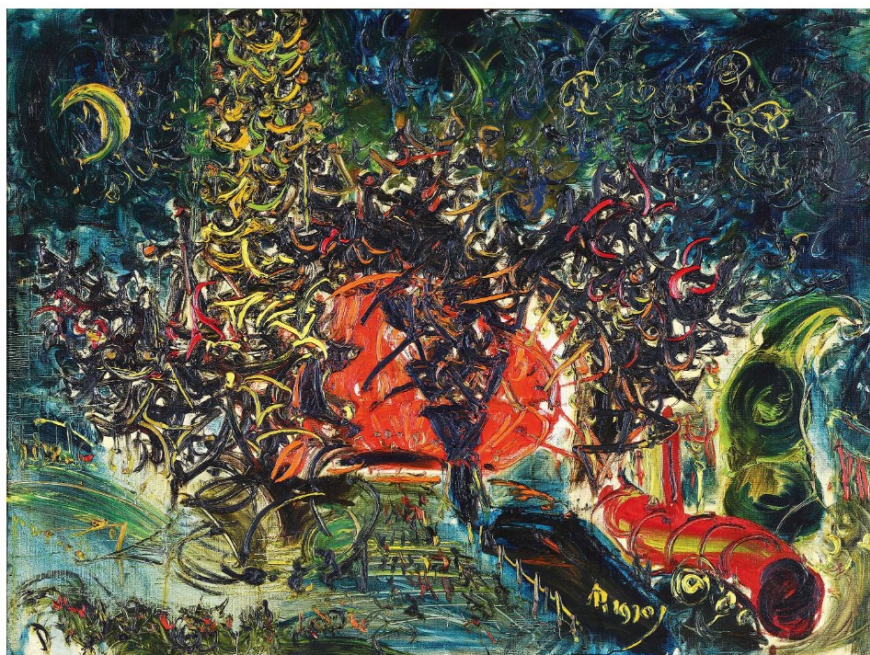


Figura 44: Kusuma Affandi, Expo 70 – Toshiba-IHI Pavilion, 1970 (Collezione privata).

Gli interessi di Kurokawa per la forma modulare e le possibilità di espansione tridimensionale sono stati esplorati alla scala del componente strutturale del padiglione Toshiba IHI all'Expo'70 di Osaka. Il teatro circolare a cupola da 500 posti ha offerto a Kurokawa l'opportunità di sperimentare una struttura spaziale metallica composta da unità tetraedriche che potevano essere espanse in 14 modi diversi. "Il primo obiettivo", ha dichiarato Kurokawa, "è stato quello di introdurre questo processo di rigenerazione nell'architettura e nella pianificazione urbana; il nome è espressivo della convinzione che un'opera architettonica non debba essere congelata una volta completata, ma debba invece essere percepita come una cosa - o come un processo - che si evolve dal passato al presente e dal presente al futuro".

In linea con le ideologie espresse nel movimento metabolita Kisho Kurokawa, per progettare il padiglione, utilizzò come unità di base la forma di un tetraedro (un particolare poliedro formato da quattro facce quattro vertici e sei spigoli: è un solido platonico avente come facce quattro triangoli equilateri perfettamente uguali fra di loro con tutti gli spigoli congruenti), il quale si poteva scomporre secondo ben 14 direzioni secondo un sistema di coordinate ben preciso.



*Figura 45: Il Toshiba IHI Pavilion di Kisho Kurokawa (l'immagine è tratta da S. Köhn, Presentation for the Japanology seminar "Resistance in Japan", University of Cologne, 2 Nov 2021).*

Proprio come alla base dell'architettura metabolica tutto può essere scomposto e diviso nelle sue unità costituenti, anche in questo caso, come venne ampiamente mostrato in altre opere dell'architetto, è possibile avere un modello di base per poi poterlo variare e crearne virtualmente un numero infinito.

Per il Toshiba IHI Pavillon furono precisamente utilizzati quattro differenti tipi di tetraedri i quali si possono definire come tetra-unità. Kurokawa definisce quattro diversi tipi di unità (una con bordi liberi diritti e 3 con bordi arcuati) per questo edificio cosicché ognuna di esse potesse avere diverse resistenze modificando le estremità delle falde che costituivano l'unità, senza però modificarne la geometria. È possibile notare infatti come le tetra-unità più basse e vicine ai punti d'appoggio siano più robuste e "piene", mentre più si sale in altezza, più tali tetra-unità si assottigliano e perdono materiale fino a che in cima le estremità delle saldature che



formano i tetraedri sono state talmente tanto tagliate da perdere quasi totalmente la superficie. Il “tetra-piece”, che è un elemento del “tetra-frame”, è costituito da sei diverse piastre di acciaio saldate ai bordi di un triangolo isoscele formato da 4 vertici e un punto di gravità. Utilizzando l'apice del tetra-pezzo come punto di contatto, gli elementi sono collegati da sfere di acciaio fuso. L'unione di tutte le tetra-unità conferiva alla struttura una tale resistenza che era in grado di resistere addirittura a venti con forze paragonabili a quelle di un tifone (K. Kurokawa, 1997).

Una volta terminata la fase di progettazione dell'intera struttura, ci si accorse che il risultato finale era molto somigliante ad una foresta fatta interamente in acciaio. Fu così che questo padiglione fu rinominato “La civiltà è nata dalla foresta”. La “foresta” d'acciaio si legava così al tema fulcro dell'esposizione, l'armonia e il progresso della civiltà. L'edificio fu progettato e costruito con l'intenzione che dovesse durare solamente per il periodo dell'esposizione, cioè, sei mesi, e così fu. La cosa sorprendente è il fatto che, una volta terminato il periodo di mostra, l'edificio, data la sua struttura e avanzata tecnologia, fu in grado di essere facilmente smontato e tutto il materiale utilizzato fu rigenerato e riciclato per altri usi futuri.

L'edificio nel suo complesso era composto da ben 1444 tetra-unità, che si disponevano sopra un ampio piedistallo accessibile tramite delle rampe. All'interno della struttura metallica, assomigliante ad una foresta, fu installato un teatro a visione globale (all'interno del teatro vi erano 9 schermi di dimensioni veramente notevoli, avendo un diametro esterno pari a circa 40m), che poteva facilmente contenere all'interno 500 persone. Un aspetto fondamentale che rende questo padiglione tecnologicamente molto avanzato fu il fatto che la struttura interna poteva muoversi sotto la cupola grazie a un pavimento girevole di 26 metri di diametro che va su e giù. La sala con 500 posti a sedere sale attraverso la “foresta” fino in cima. Il pavimento è sostenuto da un martinetto con una pressione dell'olio di 300 tonnellate. L'edificio, che si trova al centro dell'accesso principale al padiglione Toshiba IHI, si completa con una torre alta 50m che è assemblata dagli stessi tetra-pezzi.

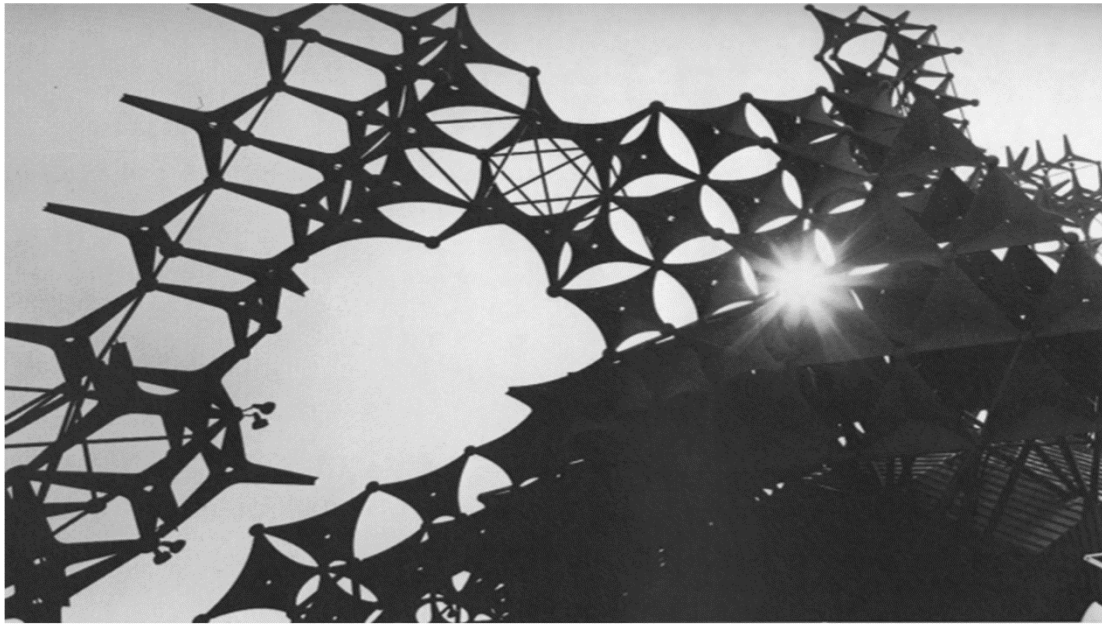
Nel contesto di questa eccezionale mostra di innovazione architettonica, è importante ricordare il ruolo del concetto di sostenibilità nel design del padiglione Toshiba-IHI. In effetti, Kurokawa concepì non solo una struttura imponente e versatile, ma anche principi di sostenibilità ambientale che erano pionieristici per l'epoca. La progettazione modulare e l'utilizzo di materiali riciclabili hanno reso la costruzione e lo smontaggio semplici e il riutilizzo delle risorse efficiente. Questo metodo ha mostrato una prospettiva lungimirante e ha anticipato decenni le pratiche di architettura sostenibile che sono oggi diffuse. La possibilità di smontare e reimpiegare i componenti strutturali ha ridotto l'impatto ambientale dell'evento e ha anche sostenuto l'idea di un'architettura che si adatta e si trasforma nel tempo. In questo modo, il padiglione non rappresentava solo una meraviglia tecnologica e artistica, ma anche un manifesto di responsabilità ecologica e di innovazione sostenibile, capace di ispirare le generazioni future di architetti e urbanisti.



*Figura 46: Il Toshiba IHI Pavilion di Kisho Kurokawa (K. Kurokawa, Metabolism in Architecture, Studio Vista, Londra, 1977, p. 129).*



*Figura 47: Il Toshiba IHI Pavilion di Kisho Kurokawa, particolare dell'interno del teatro (K. Kurokawa, Metabolism in Architecture, Studio Vista, Londra, 1977, p. 131).*



*Figura 48: Il Toshiba IHI Pavilion di Kisho Kurokawa, particolare del telaio spaziale (K. Kurokawa, Metabolism in Architecture, Studio Vista, Londra, 1977, p. 128).*



*Figura 49: Il Toshiba IHI Pavilion di Kisho Kurokawa durante le fasi costruttive (K. Kurokawa, Metabolism in Architecture, Studio Vista, Londra, 1977, p. 130).*

### 6.3.1. La modellazione parametrica del Toshiba IHI Pavilion e la sua ottimizzazione tramite algoritmi genetici

In questo paragrafo si andrà a sviluppare l'algoritmo grazie al quale si modellerà il Toshiba IHI Pavilion di Kisho Kurokawa e la sua relativa ottimizzazione tramite il software Grasshopper. È bene evidenziare che gli input di base fissi e le variabili di programmazione che entreranno in gioco durante tutto il processo di costruzione dell'algoritmo saranno descritti e definiti di volta in volta durante i vari step di lavoro per una maggior chiarezza nella lettura e per una maggior coerenza della natura logica del problema. Inoltre va aggiunto che il Toshiba IHI Pavilion di Kisho Kurokawa non è la sua opera più celebre e ciò ha portato ad una quantità di testi, documenti o disegni minore rispetto ad altri lavori e ciò ha portato spesso a scelte ipotizzate. L'operazione che ora sviluppiamo è giustificata dal fatto che l'obiettivo del caso studio non è quello della mera riproduzione esatta della struttura, ma piuttosto quello di evidenziare le potenzialità del software Grasshopper, poter migliorare la forma e la struttura del padiglione e quello ovviamente di esplorazione del pensiero che ha guidato la sua creazione attraverso l'uso di algoritmi come strumenti di indagine. Durante lo svolgersi del lavoro si sono utilizzati numerosi plug-in scaricabili da <https://www.food4rhino.com>, servizio offerto da McNeel nel quale gli utenti possono trovare gli ultimi plug-in per Rhino, per Grasshopper, materiali, texture, script e altro, possono inoltre contattare gli sviluppatori, accedere al forum e tanto altro, i quali verranno citati ma non descritti perché non essenziali ai fini della tesi. Come ultimo aspetto bisogna sottolineare che, mentre si procederà all'esposizione dell'algoritmo, si accompagnerà la spiegazione testuale, per una più immediata e completa chiarezza, con dei riferimenti di immagini, che però sono solo dei frammenti dell'algoritmo completo o degli algoritmi subordinati.

Le immagini per comodità sono raccolte nell'Appendice di questo lavoro.

### 6.3.1.1. Costruzione del teatro

La costruzione del teatro è relativamente semplice: si tratta di una geometria composta da barre metalliche come struttura con al di sopra una pelle arancione, anch'essa metallica, che la nasconde; all'interno, invece, l'edificio ospita un vasto spazio che funge da teatro-cinema. Come vedremo, per la costruzione di questa parte di opera così come per le sezioni esaminate successivamente sono in gioco alcune variabili e alcuni elementi fissi. Poiché si utilizza un algoritmo surrogato, cioè subordinato, questo "pezzo" di algoritmo verrà poi ripreso per la costruzione più complessa dell'opera finale come unico "pacchetto", ma così perdendo alcune delle variabili e i parametri modificabili. Per la logica della costruzione è stato più facile partire dall'involucro e dalla creazione della geometria del teatro per poi passare alla struttura di supporto strutturale.

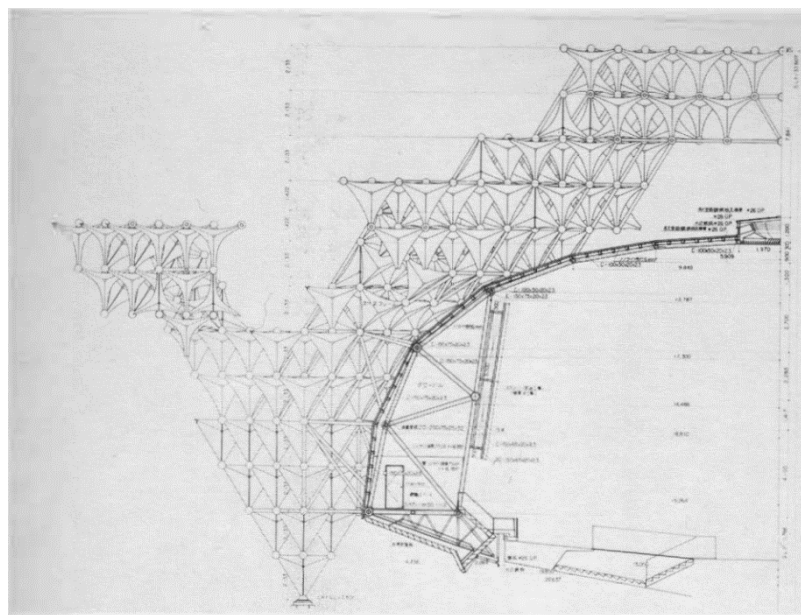


Figura 50: Sezione longitudinale del padiglione (K. Kurokawa, *Metabolism in Architecture*, Studio Vista, Londra, 1977, p. 129).

Per quanto riguarda la geometria del teatro, si tratta di una forma semplice, simile ad una forma sferica sfaccettata. In Grasshopper è stata quindi fatta grazie ad una curva principale (generatrice) che è stata poi rivoluzionata (per la comprensione della geometria si rimanda al capitolo secondo) secondo un asse centrale per 360° così da formare un oggetto tridimensionale proprio con la stessa logica con cui si costruiscono alcune figure geometriche viste nel capitolo secondo *La geometria delle superfici*. La generatrice della forma può essere considerata come input fisso, la forma allora non può variare, utilizzando i disegni del padiglione originali di Kurokawa (se non scalandolo o utilizzando particolari comandi non di nostro interesse).

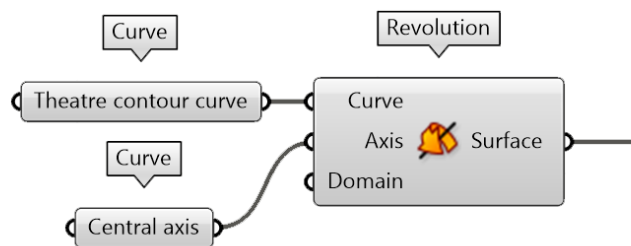


Figura 51: La creazione dell'involucro esterno secondo una rivoluzione attorno ad un asse principale utilizzando i componenti presenti in Grasshopper.

Dopo aver creato la geometria di base, si è proseguito nel suddividerla in dodici parti tramite meridiani e paralleli, secondo un parametro variabile (si è deciso di scegliere di dividere la struttura in dodici parti perché rispecchia le scelte del padiglione originale di Kurokawa), e creare quindi delle linee che, incontrando i punti delle intersezioni dei meridiani e dei paralleli, formano la struttura principale. L'immagine del teatro (fig. n. 50) corrisponde alla sezione longitudinale dello stesso, dove si può vedere bene la parte esterna ("l'involucro") e la parte interna ("il supporto"; questa parte fa da supporto non solo alla pelle esterna del teatro, ma anche ai giganteschi schermi che verranno poi

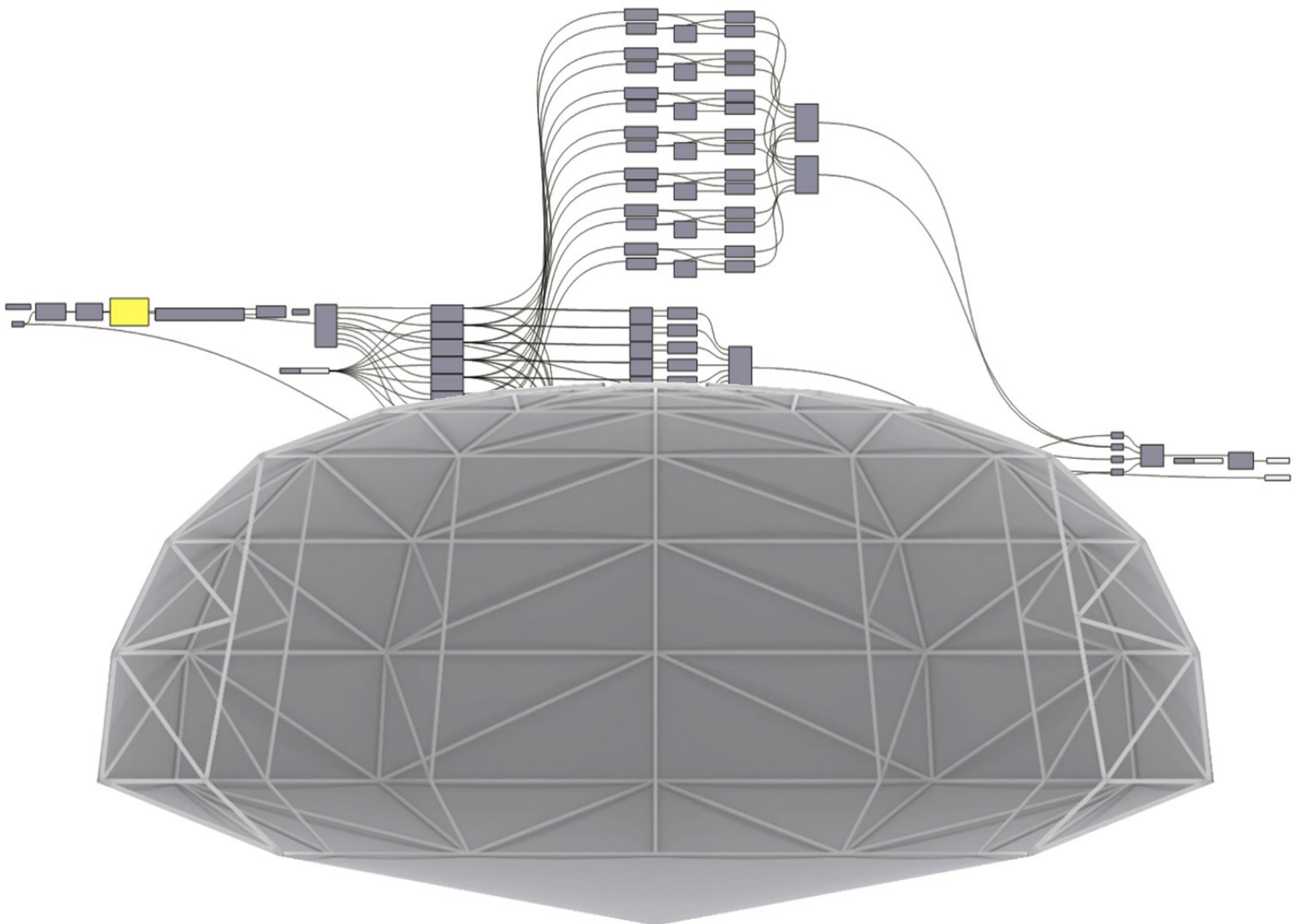
utilizzati all'interno del teatro). Questa parte di opera viene così "cucinata" ("bake": termine utilizzato dal software Rhinoceros, il quale permette di utilizzare e modificare le cose create in Grasshopper). Ora, quindi, si utilizzerà tutta questa struttura in modo fisso come input per altri svolgimenti senza poter più modificare quasi tutte le variabili. Ciò, come si mostrerà nei paragrafi successivi, verrà fatto anche in altre occasioni per rendere meno pesante l'algoritmo e più maneggevole.

Per la struttura interna, infine, si è proceduto nell'utilizzo del plug-in Karamba per verificare la stabilità della stessa. Questo utilizzo dello strumento non intende calcolare strutturalmente e definire gli sforzi che insistono fra gli elementi, ma solo essere un aiuto per un'accurata progettazione nelle fasi iniziali. Questo processo verrà attuato, infatti, per ogni paragrafo successivo per "monitorare" l'andamento della costruzione. Il dimensionamento è stato ipotizzato non conoscendo le reali misure del padiglione nella realtà. Si sottolinea nuovamente come questi sforzi e queste deformazioni vanno presi in considerazione solamente come parametri utili a capire l'andamento e il comportamento della struttura, in sostanza per rispondere alla domanda "Cosa succederebbe se  $k$  (variabili del sistema) avesse avuto questa quantità?". Nuovamente, l'intento in questo caso studio è quello di comprendere gli elementi essenziali della forma del Toshiba IHI Pavilion di Kurokawa e di ricreare forme che si originano da quegli stessi elementi utilizzando una nuova tecnologia, allora inesistente. Questi parametri in questa fase sono del tutto variabili, ma nelle fasi successive questi dati verranno "congelati" per una più facile e veloce costruzione dell'algoritmo. In ogni caso rimane la possibilità di compiere dei passi indietro per modificare l'intera procedura.

Il mancato sviluppo delle parti interne del teatro può essere attribuito alla considerazione che la modellazione di tali componenti risulta insignificante nel contesto delle attività e degli obiettivi definiti finora. Questa mia decisione è stata presa sulla base di una valutazione critica delle risorse disponibili e delle priorità del progetto. Data l'attenzione



concentrata sui segmenti esterni e sulla struttura generale del teatro, si è ritenuto che l'espansione delle parti interne non avrebbe aggiunto valore significativo alle finalità e ai risultati attesi. Pertanto, si è deciso di concentrare le risorse e gli sforzi su aree che contribuiscono in modo più sostanziale al raggiungimento degli obiettivi prefissati.



### 6.3.1.2. Costruzione del tetraedro

Per la costruzione del tetraedro anche in questo caso è stato piuttosto semplice: tramite il componente Polygon in Grasshopper si è riusciti a creare un poligono a tre lati perfettamente uguali e tre angoli uguali da  $60^\circ$  all'interno della figura (infatti per calcolare gli angoli interni di un qualsiasi poligono basta fare  $n - 2 * (\frac{180^\circ}{n})$  dove  $n$  sono i lati del poligono). In questa fase della programmazione gli input fissi sono chiaramente il numero di lati del poligono, in quanto se modificati si andrebbe ad intaccare la geometria dell'elemento di base; invece ciò che può essere modificato tramite parametro è la lunghezza dei lati. Dopodiché si è passati alla costruzione della geometria dal piano allo spazio creando la figura tridimensionale. Ciò è stato possibile estrudendo il centro del poligono di base e creando così il vero e proprio tetraedro. Il tetraedro è una figura composta da quattro facce, tutte triangoli equilateri uguali, quattro vertici e sei spigoli. Per rendere il tetraedro regolare, quindi con tutti i lati, gli angoli e le facce perfettamente identiche, si è ricorsi alla relazione che lega l'altezza tra vertice e base della figura ( $(\frac{\sqrt{6}}{3}) * x$ , dove la  $x$  rappresenta la lunghezza di un lato).

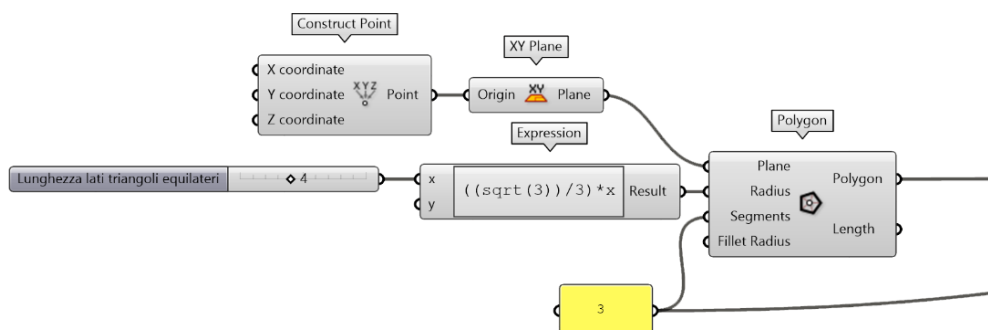
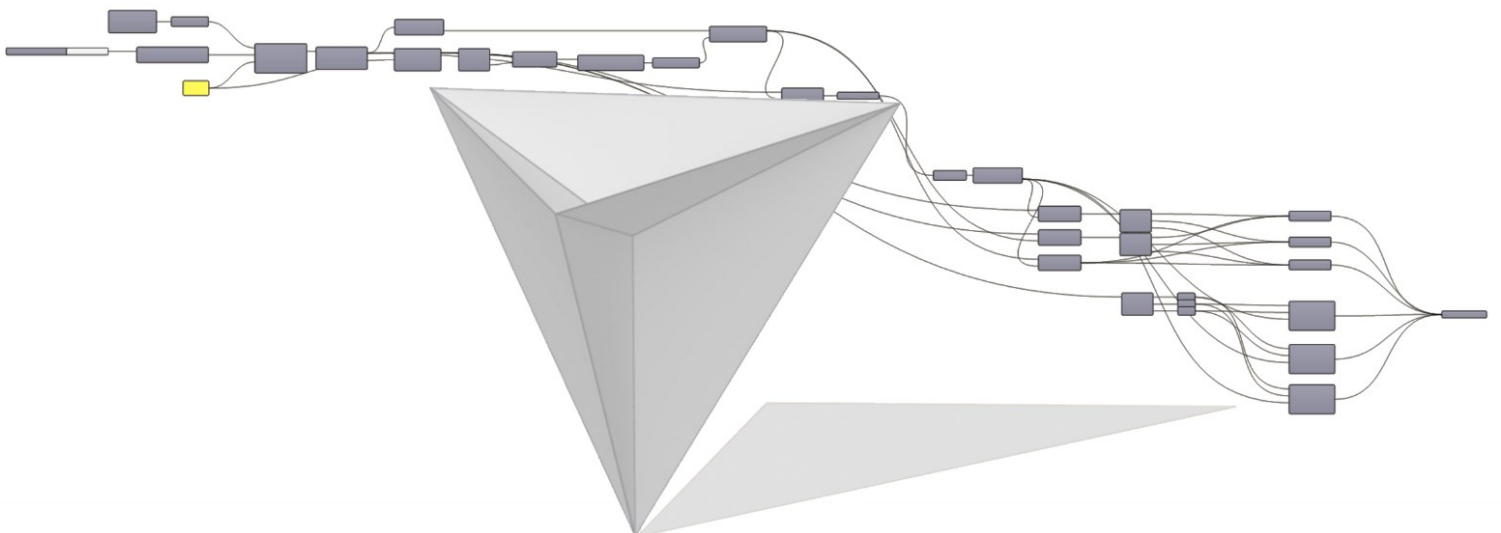


Figura 52: La creazione del tetraedro regolare utilizzando i componenti in Grasshopper.

La figura che è presente nell'IHI Pavilion di Kurokawa, però, è costituita da piastre metalliche saldate fra loro che creano sì un tetraedro come figura modulo, ma presentano una particolarità: la base è incavata all'interno della geometria così da creare una figura con le stesse dimensioni ma non più a quattro facce e quattro vertici, bensì a sei facce e cinque vertici. Per realizzare una mesh il più possibile "pulita" senza linee, punti o superfici sovrapposti, è stato utilizzato il plug-in Weaverbird's. Infine, si fa notare come nel padiglione costruito da Kisho Kurokawa esistono quattro diverse tipologie di tetraedri, i quali variano per alleggerirsi man mano che si sale in elevazione e ruotano per adattarsi meglio alle linee del teatro. Per una ragione semplificativa è stato utilizzato solamente il tetraedro principale all'interno dell'algoritmo.

Anche in questo caso, come fatto con il teatro, è stata fatta una veloce analisi agli elementi finiti con Karamba per visualizzare se l'elemento preso singolarmente fosse stabile. Risulta che su una shell, con delle forze puntiformi posizionate sui tre punti principali del tetraedro, ipotizzando un incastro perfetto al vertice di appoggio, ci sarebbero delle deformazioni tali da rendere l'elemento stabile. Anche in questa fase va ricordato che l'utilizzo del plug-in Karamba va pensato solamente come un aiuto, soprattutto visivo, del possibile andamento del tetraedro sotto determinati sforzi e per individuare le possibili deformazioni.



### 6.3.1.3. Costruzione della “foresta” di tetraedri

Per la costruzione della “foresta” di tetraedri il processo è stato più lungo e più complesso, perché risulta la parte dell’intera struttura più grande ed intricata nonché la parte più interessante e principale. Per “foresta” si intende dire, come già detto nel paragrafo 6.3 dedicato al design del padiglione, tutti quei tetraedri che posizionandosi uno sopra l’altro formano una vera e propria boscaglia molto fitta a simulare una foresta geometrica apparentemente casuale. In questo step dell’algoritmo del padiglione Toshiba IHI si è cercato di trovare un metodo corretto per posizionare i tetraedri nei punti giusti (il vertice di base che poggia sui tre vertici superiori) e trovare una giusta configurazione in modo tale che la “foresta” sia stabile.

Prima di spiegare i passaggi e il metodo eseguito per riuscire a disegnare tutto ciò, va detto che, come nei due paragrafi precedenti e per la costruzione seguente dell’algoritmo, sono stati utilizzati sia parametri variabili che fissi. Definire i parametri è essenziale perché determina ciò che si può cambiare all’interno della costruzione e cosa no. In questa fase i parametri fissi, cioè tali che vengono dati come non variabili, sono: il numero di appoggi per la “foresta” di tetraedri (infatti, come nel padiglione reale, sono stati ideati sei appoggi con un tetraedro iniziale) e il “numero di piani” dove i tetraedri si possono posizionarsi (i “numeri di piani” non sono altro che la quantità di volte che un tetraedro può alzarsi di una  $h$  pari all’altezza del tetraedro stesso; il vincolo fissato è di 10 piani, i quali sono sufficienti per ricoprire totalmente il teatro come nel Toshiba IHI Pavilion. Calcolando che un tetraedro ha una altezza massima di 2m l’intera struttura avrà un massimo di 20m di altezza dal suolo). Le variabili in gioco sono

principalmente due<sup>14</sup>: la posizione dei sei punti di appoggio (i punti, come si vedrà in seguito, non sono fissi ma nemmeno casuali e arbitrari) e il numero di tetraedri all'interno della "foresta".

All'interno dell'algoritmo di questa sezione per la prima volta si è usato il componente *cluster*, che serve per raggruppare le funzioni di un piccolo algoritmo in Grasshopper; è una tecnica efficace per ridurre la complessità visiva del file.

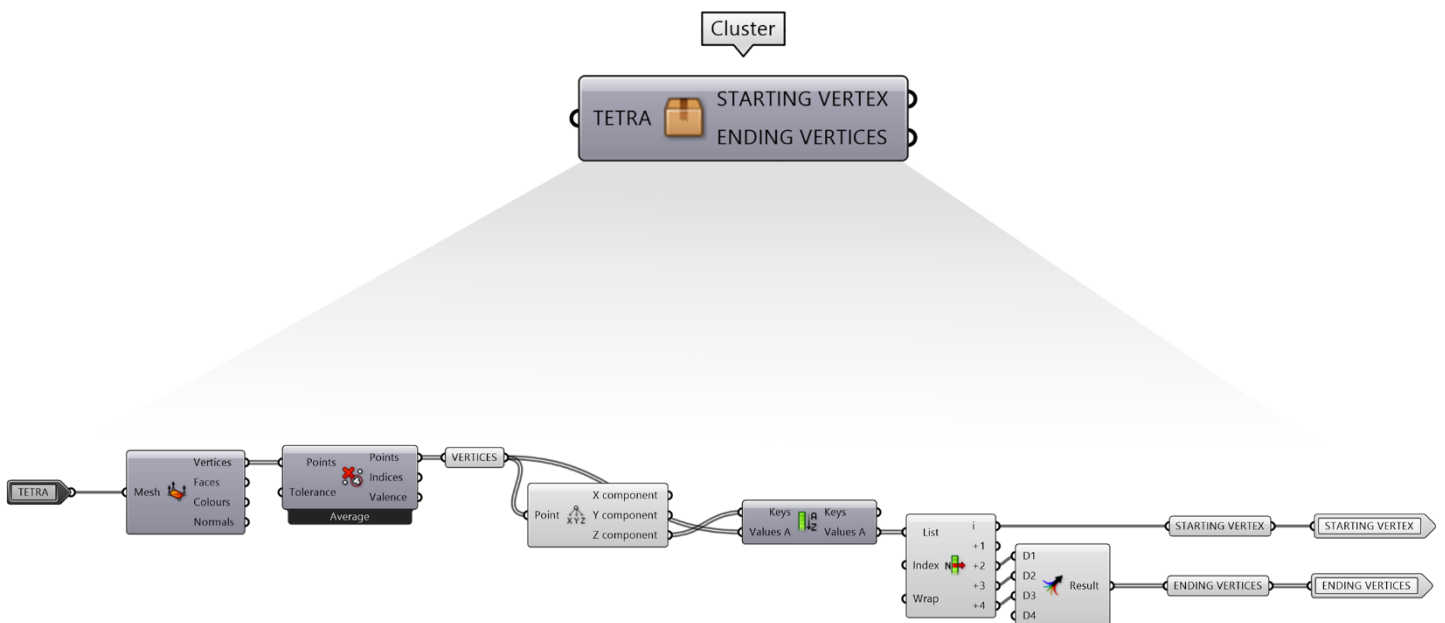


Figura 53: I componenti all'interno del primo cluster utilizzato all'interno dell'algoritmo in Grasshopper.

<sup>14</sup> Nell'algoritmo in realtà i parametri sono moltissimi non solo due, ma i parametri principali, i quali governano la struttura e ne modificano le principali proprietà, sono quelli sopra citati. Per esempio, la variabilità del numero di tetraedri all'interno dell'algoritmo è basata su quattro o cinque parametri variabili utili a veicolare nel modo più corretto possibile l'informazione e di almeno trenta comandi diversi di Grasshopper. È evidente come più l'algoritmo diventa complesso e più parametri ci sono che ne possono controllare le sue caratteristiche. Risulta normale prendere in considerazione solo quelle che possono dare informazioni tali da modificare drasticamente il lavoro svolto, quelle interessanti ai fini del lavoro di tesi e quelle davvero utili alla comprensione dell'algoritmo.

I cluster consentono di copiare e incollare facilmente un particolare algoritmo sia all'interno del file che in altri file di Grasshopper che altrimenti andrebbe copiato e incollato più volte o addirittura rifatto. Se si modifica un cluster, tutte le copie all'interno del file aperto verranno aggiornate di conseguenza. Così facendo risulta comodo non dover rifare sequenze di algoritmo all'interno del nostro campo di lavoro e poter aver un componente con specifiche funzioni (scelte da noi) utilizzabile a nostro piacimento anche in altri file di Grasshopper.

Per la costruzione della “foresta” di tetraedri si può suddividere il processo in quattro sotto procedimenti:

- I. I passaggi atti a trovare la nuvola di punti per la destinazione dei tetraedri (la “foresta” di tetraedri);
- II. La ricerca dei sei punti di partenza;
- III. La successiva diminuzione in una nuvola di punti più piccola tramite il metodo dei campi di forza;
- IV. La creazione della geometria sui punti appena trovati.

Il primo punto da realizzare è stato quello di determinare i punti dove i tetraedri si sarebbero andati a collocare per la creazione della foresta. Questo passaggio è stato particolarmente impegnativo in quanto, per una corretta ricerca degli  $n$  punti della foresta, si è dovuto pensare in maniera differente alla comune logica costruttiva che segue ogni architetto o ingegnere. Infatti, inizialmente si è cercata la successione di punti uno dopo l'altro crescendo in altezza (come in una normale fase di costruzione: si posano gli elementi partendo dal basso e progressivamente ci si alza verso l'alto), ma la risposta dell'algoritmo era incorretta. Ricordando che, come studiato nei capitoli precedenti, la costruzione di un algoritmo parte dalla soluzione finale e successivamente si programma in maniera “logica”, allora si è costruita dapprima la nuvola di punti tramite un complesso sistema di griglie e dopodiché si sono potuti ricercare all'interno della nuvola di punti (con

all'interno 13.284 punti) i soli punti necessari alla costruzione della foresta (punto III). Così facendo la risposta ottenuta è risultata soddisfacente. Risultando piuttosto complessa e lunga la spiegazione con precisione di ogni singolo componente utilizzato durante questa prima fase e si preferisce lasciare l'interpretazione visiva dell'algoritmo in Grasshopper. Ogni collegamento fatto è dovuto ad uno studio logico-geometrico della struttura (si veda Appendice, Tav. 3).

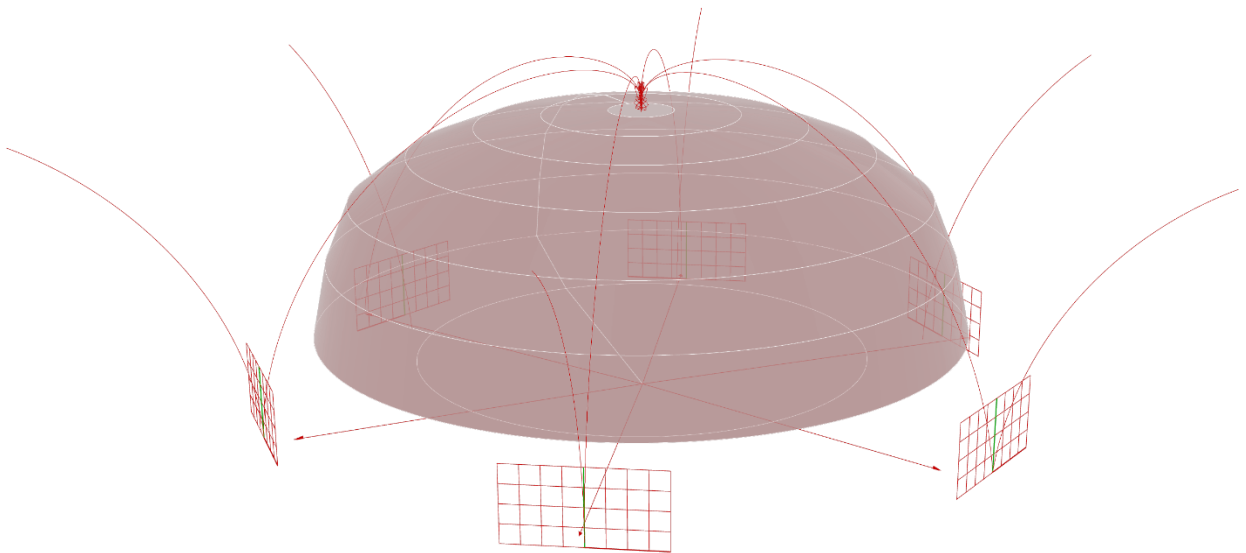
Per la ricerca dei sei punti iniziali da dove parte la foresta di tetraedri si è proceduto alla creazione di una circonferenza con lo stesso centro della geometria del teatro e raggio variabile, in quanto si è osservato che nel Toshiba IHI Pavilion i punti di partenza sono esattamente posizionati attorno al teatro in modo circolare. Essendo il cerchio costruito in modo parametrico è possibile, variando i dati, manipolare a proprio piacimento ogni aspetto facendo così che i sei punti iniziali siano totalmente manovrabili. È anche vero che, però, i punti iniziali del padiglione costruito sono equidistanziati; si è deciso, pertanto, per una maggiore verosimiglianza, di seguire regole ben definite e mantenere una minor possibile libertà<sup>15</sup>.

Per determinare la successiva diminuzione della nuvola di punti trovata in precedenza si è proceduto tramite il metodo dei campi di forza. La nuvola di punti finale dai 13.284 punti iniziali diminuisce a circa 200-600 punti (il range così elevati di punti all'interno della nuvola è dovuto alla scelta dei parametri; a seconda dei valori che si inseriscono all'interno dell'algoritmo i punti possibili cambieranno. Il lavoro di ottimizzazione della quantità di punti, e quindi di tetraedri, avrà luogo nel paragrafo 6.3.1.5). Tramite componenti specifici si sono

---

<sup>15</sup> Pur non sapendo se la decisione così regolare di tali punti effettuata dall'architetto sia data da una scelta di tipo estetica oppure da una scelta costruttiva, da questo passaggio si può evincere come le potenzialità di Grasshopper e più in generale del parametricismo permettono una maggiore libertà progettuale, in quanto, questa decisione non avrebbe in nessun modo creato preoccupazioni.

andati a creare dei *campi*, i quali lavorano come dei veri e propri campi magnetici. Caricando con valori gaussiani positivi e negativi punti strategici all'interno e all'esterno del padiglione, tramite il comando *Traverse On Field* si sono creati dei percorsi, sui quali poi verranno posizionati i tetraedri per la foresta finale. Infatti i punti trovati per la foresta finale sono quelli che saranno posizionati con la minor distanza da questi percorsi. La distanza che determina la scelta dei punti e i punti da escludere è parametrica e ottimizzata tramite il componente Wallacei nel paragrafo 6.3.1.5.

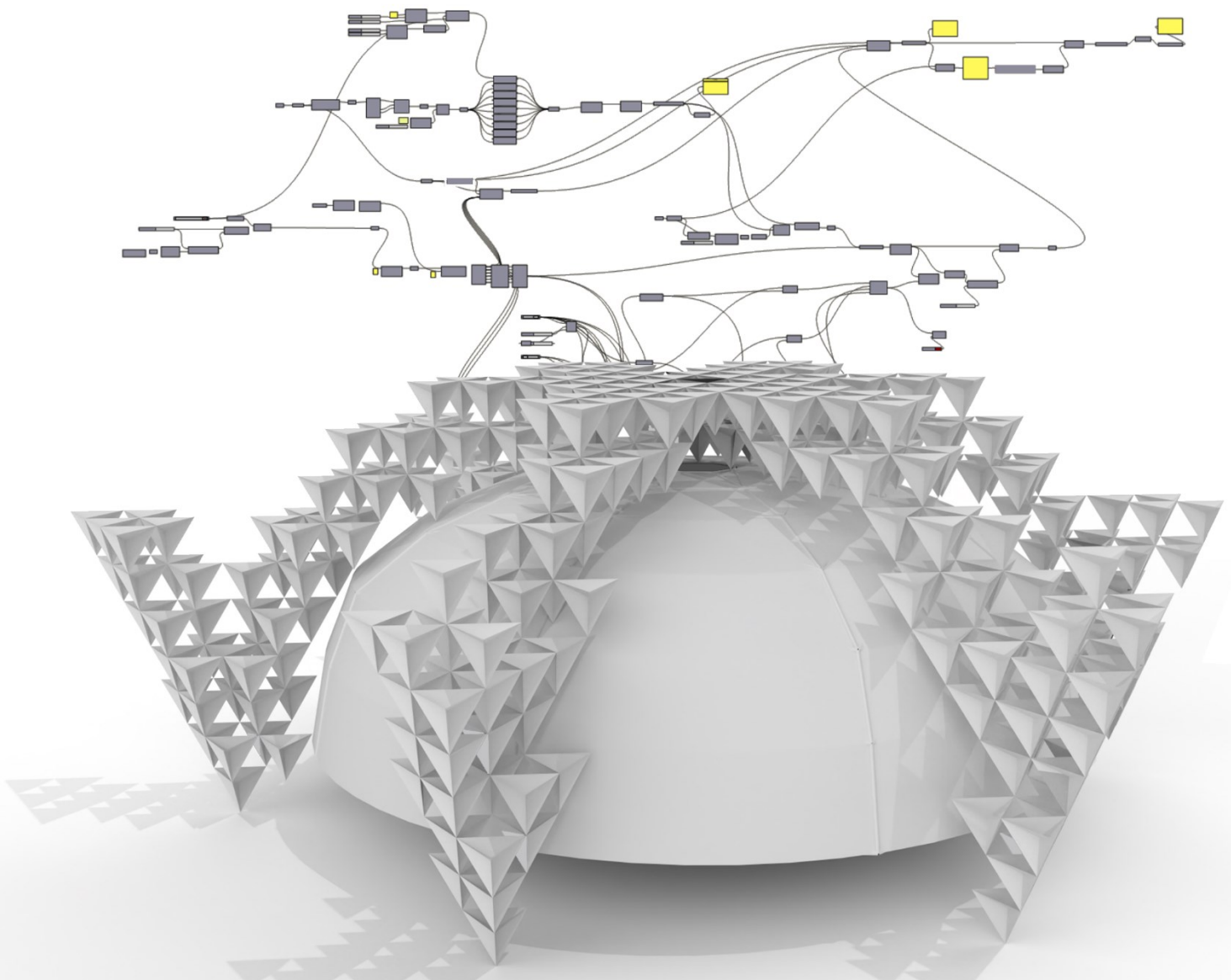


*Figura 54: La costruzione dei percorsi tramite l'utilizzo di campi di forza.*

È possibile notare che i percorsi creati dai campi di forza dove poggeranno i tetraedri non sono solamente “ad arco” sopra il teatro (strutturalmente il teatro è sospeso grazie ad un telaio costituito dalla foresta di tetraedri che concettualmente funzionano come delle travi reticolari ricurve; inoltre sono posizionati dei tetraedri più pesanti rispetto ad altri in posizioni che creano contrappesi alle spinte degli archi), ma si proiettano anche verso l'esterno. Questo è dovuto al fatto che si è cercato di ricercare quei punti che fungono da contrappeso.



Infine come ultimo passaggio si posizionano i tetraedri con il proprio vertice basso (la punta della geometria) su tutti i punti che sono compresi all'interno della nuvola di punti ridotta. Così facendo si completa lo svolgimento di questa sezione e si può guardare il risultato (uno dei tantissimi possibili risultati, considerando il numero di parametri e i valori che possono assumere: tenendo conto che sono parametri indipendenti, le possibili combinazioni sono più di 12.000, quindi una diversa configurazione per ognuna di esse) in figura.



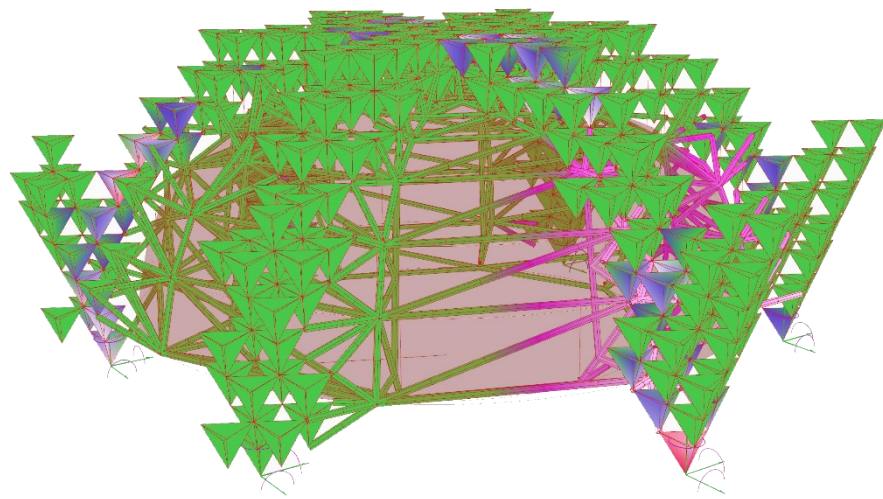
#### 6.3.1.4. Costruzione dei punti di ancoraggio fra tetraedri e teatro e l'utilizzo del plug-in Karamba 3D

Per concludere la riproduzione del padiglione ci si è adoperati per inserire le barre metalliche che collegano, proprio come una trave reticolare come citato prima, i tetraedri fra di loro e la struttura del teatro. I parametri modificabili in questo caso sono il numero di barre presenti nella struttura: infatti, è possibile modificare tale dato per aumentare o diminuire la quantità di aste.

Si procede ora a contestualizzare l'utilizzo di Karamba3D all'interno di Grasshopper. Karamba3D non è stato utilizzato per eseguire un'analisi strutturale approfondita e dettagliata, come potrebbe essere richiesta in una fase finale di progettazione ingegneristica, in quanto non è lo scopo di questo lavoro. Si delineerà meglio e si trarranno le conclusioni nel paragrafo 6.3.2. Lo strumento è stato adottato principalmente per due scopi: la verifica preliminare della stabilità strutturale in modo da non ricercare una forma e una soluzione fuori dalla realtà e la preparazione al processo di ottimizzazione successivo.

Per funzionare il plug-in Karamba ha bisogno di alcuni dati in input, quali: l'elemento da analizzare, il materiale, la posizione e la tipologia dei supporti, la posizione dei carichi e il loro valore. In Grasshopper è necessario seguire una serie di passaggi chiave: il processo inizia con la definizione di una *mesh* o con l'importazione di geometrie 3D, che vengono poi trasformate in un modello strutturale mediante la componente "Assemble Model". In questa fase si specificano anche le proprietà dei materiali e delle sezioni trasversali degli elementi strutturali utilizzando componenti dedicate come "Beam-Joints" e "Material Properties". Successivamente, si definiscono le condizioni di vincolo e di carico, inclusi supporti, carichi concentrati o distribuiti, e prescrizioni di spostamento. Questi elementi sono essenziali per

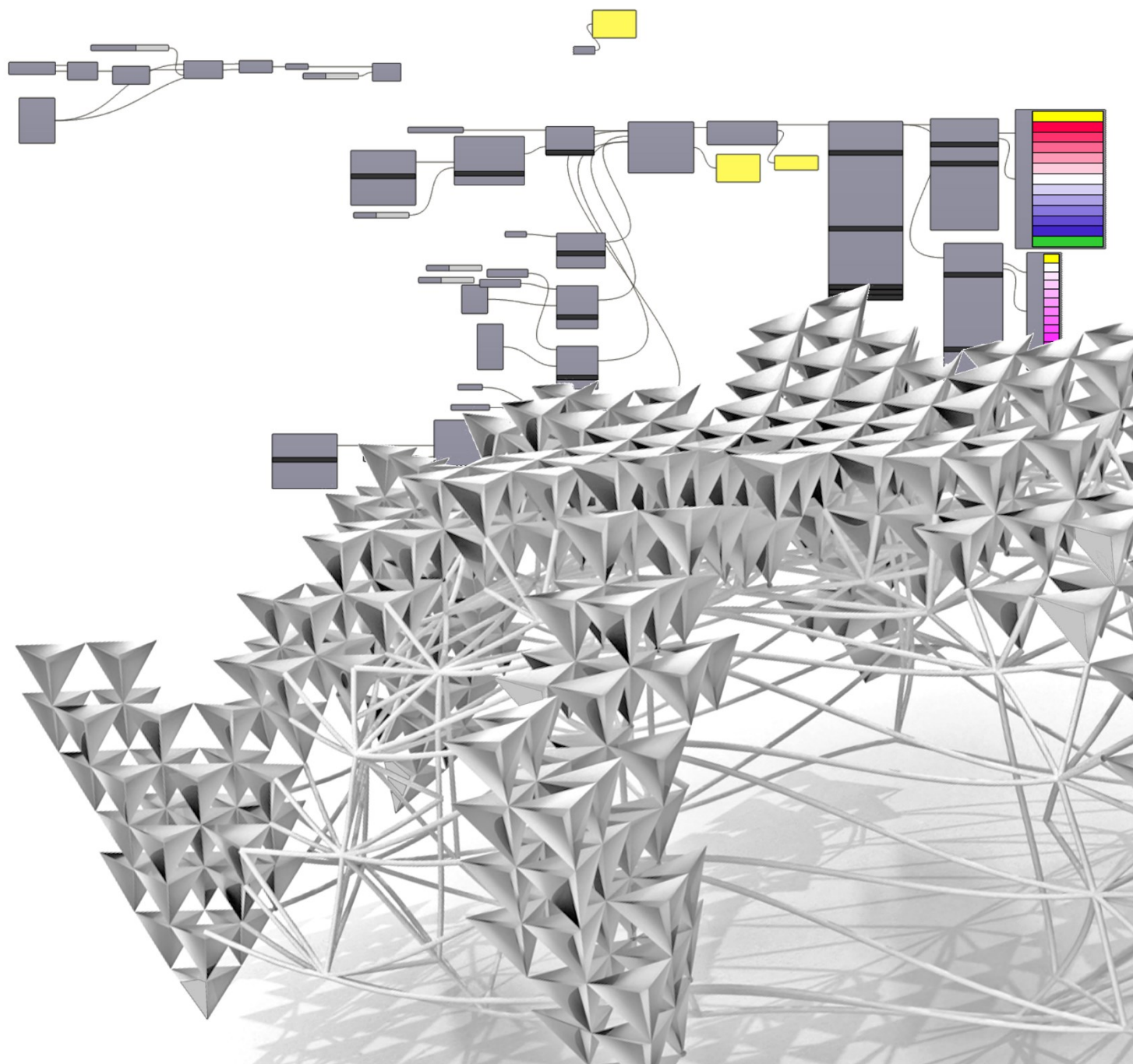
simulare accuratamente le condizioni operative della struttura. Una volta completata la configurazione, si esegue l'analisi strutturale tramite il componente "Analyze", che consente di valutare diversi aspetti, come deformazioni, forze interne e reazioni ai vincoli. Infine i risultati dell'analisi possono essere visualizzati e interpretati utilizzando componenti come "ModelView" e "BeamView", che forniscono una rappresentazione grafica intuitiva delle deformazioni e delle sollecitazioni.



*Figura 55: Visualizzazione dell'analisi di Karamba3D della struttura.*

Questo approccio ha permesso di ottenere rapidamente una visione generale delle prestazioni strutturali e di predisporre il modello per l'ottimizzazione, senza entrare nei dettagli complessi che caratterizzano una completa analisi strutturale. Questo plug-in è efficace e pertinente al lavoro perché offre la possibilità di visualizzare rapidamente una vasta gamma di dati strutturali, come deformazioni, tensioni e forze interne, consentendo un'analisi immediata delle condizioni di carico e delle risposte della struttura. La visualizzazione dei risultati in tempo reale facilita l'identificazione di eventuali problematiche e la valutazione delle prestazioni dell'edificio. In

secondo luogo, l'impiego di Karamba3D è stato propedeutico all'ottimizzazione strutturale, che sarà trattata nel paragrafo seguente.



### 6.3.1.5. Ottimizzazione tramite il plug-in Wallacei

L'ultima fase della riproduzione del Toshiba IHI Pavilion consiste nell'ottimizzazione del progetto utilizzando il plug-in Wallacei, descritto nel paragrafo 5.3.3. Per funzionare correttamente, questo strumento richiede due tipi di dati fondamentali: i "Genes", ovvero le variabili, che possono essere rappresentate da Number sliders e/o Genopools, e gli "Objectives", ossia le funzioni di fitness, che corrispondono agli obiettivi che si desidera raggiungere attraverso l'ottimizzazione.

Nel caso studio del padiglione, sono state definite quattro variabili parametriche diverse e due obiettivi, dando luogo a un'ottimizzazione multi-obiettivo (descritta nel paragrafo 3.4). La selezione delle variabili da includere nel componente è stata effettuata considerando l'impatto diretto di queste sulla struttura. Per quanto riguarda gli obiettivi, la scelta è stata guidata dal desiderio di ottenere un'ottimizzazione della forma che fosse coerente dal punto di vista strutturale e in linea con i principi filosofici dell'architetto Kurokawa. L'intento è stato quello di riflettere i valori del Metabolismo, cercando di sviluppare una struttura che risultasse economica, facile da costruire, funzionale e ottimizzata.

Successivamente, è stata configurata correttamente l'interfaccia di Wallacei, definendo la dimensione della popolazione e il numero di generazioni per le possibili soluzioni. Completato il processo iterativo eseguito dal plug-in, è stato possibile analizzare le migliaia di soluzioni generate e selezionare quelle più promettenti. L'interfaccia di Wallacei consente di selezionare le soluzioni ottimali secondo diversi criteri. In questo lavoro sono state selezionate dieci soluzioni utilizzando due metodi principali. Il primo metodo ha sfruttato l'analisi del grafico a coordinate parallele (PCP), che offre quattro approcci analitici: l'identificazione dei valori di fitness più frequenti, la selezione delle soluzioni con i valori di fitness più ricorrenti, l'analisi della differenza

relativa tra i gradi di fitness e la valutazione del grado di fitness medio. Il secondo metodo si è basato sulle soluzioni del fronte di Pareto, che possono essere estratte dalla generazione finale o dall'intera popolazione, calcolando i valori di dominanza di tutte le soluzioni presenti. Una volta esportate, le soluzioni selezionate sono state analizzate in dettaglio, consentendo di scegliere quella che meglio rispondeva agli obiettivi di progetto. Inoltre, durante il processo di ottimizzazione è stato fondamentale considerare l'interazione tra le variabili parametriche e gli obiettivi di progetto.

Ogni variazione nei parametri selezionati ha potenzialmente influenzato la configurazione strutturale del padiglione, richiedendo una continua valutazione dei risultati generati dall'algoritmo genetico. Questo approccio iterativo ha permesso di esplorare una vasta gamma di soluzioni progettuali, identificando quelle che meglio rispondevano agli obiettivi prefissati. La capacità di Wallacei di gestire simultaneamente più obiettivi ha facilitato l'identificazione di un equilibrio ottimale tra forma, efficienza strutturale e principi estetici. Questo equilibrio non solo rispecchia la visione architettonica di Kurokawa, ma dimostra anche l'efficacia degli algoritmi genetici come strumenti di supporto decisionale nel design parametrico avanzato.

### 6.3.2. Studio delle soluzioni e risposta critica

L'obiettivo di questo lavoro non è una mera riproduzione digitale del padiglione Toshiba IHI Pavilion di Kurokawa, bensì un'esplorazione profonda del pensiero che ha guidato la sua creazione attraverso l'uso di algoritmi come strumenti di indagine. L'intento, infatti, non è replicare la struttura architettonica in sé, ma investigare le ipotesi teoriche e le idee filosofiche alla base della visione dell'autore. Questo progetto si colloca all'incrocio tra arte, tecnologia e filosofia, cercando di rispondere alla domanda: "Cosa sarebbe successo se l'autore avesse pensato in un altro modo?".

Attraverso la modellazione algoritmica, esploriamo le possibili variazioni e interpretazioni del concetto architettonico, mettendo in luce come diverse scelte di pensiero potrebbero aver influito sul risultato finale. Tale approccio ci consente di comprendere meglio non solo il progetto originale, ma anche le dinamiche intellettuali e creative che influenzano il processo di design.

L'uso degli algoritmi non è quindi finalizzato alla semplice riproduzione, ma costituisce un mezzo per svelare la ricchezza del pensiero architettonico di Kurokawa, permettendoci di immaginare alternative e varianti che riflettono diverse angolazioni filosofiche. La tecnologia, in questo contesto, diventa uno strumento per l'esplorazione teorica e artistica, offrendo nuove prospettive sull'interazione tra l'idea e la sua realizzazione.

Attraverso questo studio, si vuole stimolare una riflessione più ampia sul ruolo del pensiero critico nell'architettura e sulla capacità degli strumenti digitali di amplificare e trasformare la nostra comprensione delle idee architettoniche. In sintesi, il lavoro mira a trascendere la semplice analisi strutturale per abbracciare una visione più complessa e articolata del pensiero architettonico.

L'indagine si articola attraverso la creazione di tre algoritmi genetici distinti, ognuno basato su un diverso approccio teorico e filosofico al design del padiglione di Kurokawa. All'interno di questo lavoro, però, è stato descritto ampiamente solo il terzo metodo (paragrafi 6.3.1.1 - 6.3.1.5), il quale ha ottenuto risultati migliori degli altri, in quanto sarebbe risultato troppo lungo e inefficace la spiegazione approfondita di ogni metodo non utilizzato.

Nel corso di questa tesi, nonostante l'impiego primario del terzo metodo per l'analisi e l'interpretazione, è stato fondamentale esplorare anche i primi due metodi. Tale approccio ha permesso di comprendere meglio l'evoluzione del pensiero che ha guidato alla creazione dell'opera in esame. Infatti la finalità principale del lavoro è stata proprio quella di scoprire, attraverso i vari passaggi, anche quelli che si sono rivelati meno efficaci o apparentemente sbagliati, il processo intellettuale sotteso alla realizzazione dell'opera. L'analisi dei primi due metodi, seppur non applicata in modo esaustivo, ha svolto un ruolo fondamentale nel tracciare il percorso di ricerca. Ogni metodo esplorato, infatti, ha contribuito a chiarire e a sfumare le varie ipotesi iniziali, portando a una comprensione più profonda e articolata del soggetto studiato. È stata proprio questa progressione, caratterizzata anche da errori e ripensamenti, a delineare un quadro complesso e dettagliato del pensiero creativo dell'autore. Di seguito vengono riassunti brevemente le ipotesi definite:

- **Modo 1:** Questo approccio utilizza sp-line che variano casualmente per cercare la soluzione ottimale dai punti base al vertice del teatro. L'obiettivo è trovare la foresta di tetraedri più vicini ai vari percorsi. Questo metodo simula un processo evolutivo in cui le forme si sviluppano e si affinano attraverso iterazioni di tentativi ed errori, riflettendo una dinamica di adattamento e selezione naturale.
- **Modo 2:** Qui la creazione del padiglione avviene generando tetraedri dal piano di appoggio e salendo progressivamente. Le posizioni vengono determinate tramite le sette possibilità di



movimento del singolo tetraedro in ogni passo. Questo metodo esplora la crescita strutturale modulare, in cui ogni scelta di posizione si basa su regole predefinite, simile alla crescita cristallina che combina ordine e variabilità.

- **Modo 3:** In questo approccio il padiglione è concepito attraverso archi sovrastanti dai quali esso è sospeso. Questi archi sono determinati dalla formazione di campi magnetici o energetici richiamandosi alla filosofia del metabolismo di Kisho Kurokawa. L'uso dei campi energetici permette di simulare una struttura organica e dinamica, in cui le forme sono influenzate da forze invisibili che ne guidano l'evoluzione.

Tutti e tre i metodi incorporano una componente di casualità, ma variano nel grado di indirizzamento dell'autore dell'algoritmo al fine di trovare la soluzione migliore. Il modo 3 è filosoficamente il più affascinante e vicino alla reale costruzione del padiglione, poiché riflette la complessità e l'interconnessione della natura e della tecnologia che caratterizzano l'approccio metabolista di Kurokawa.

In seguito all'analisi dei tre metodi, è stata eseguita un'ottimizzazione delle forme per identificare la combinazione migliore con tutti i parametri in gioco, con l'obiettivo di ottenere il numero minimo di tetraedri possibile e la minor deformazione possibile. Questo processo di ottimizzazione ha permesso di risparmiare materiale e ridurre i costi di costruzione, garantendo al contempo la stabilità della struttura. Il risultato finale rappresenta un equilibrio tra estetica e funzionalità e incarna anche una visione sostenibile dell'architettura, in linea con il pensiero avanzato e innovativo di Kisho Kurokawa.



# 07 • Conclusioni

Questo lavoro si divide in due parti: la prima (capp. 2 – 5) presenta gli aspetti storici e teorici dell'analisi della geometria delle superfici, dell'ottimizzazione, del design parametrico e del suo utilizzo con i nuovi mezzi informatici. Questo studio appare necessario per introdurre in modo preciso l'applicazione pratica in un caso reale. Infatti tali aspetti si configurano come una novità nell'ambito della progettazione ingegneristica e architettonica in quanto permettono di operare con notevoli vantaggi rispetto al passato.

Innanzitutto si può lavorare in modo più veloce, risparmiando tempo e rendendo coerente e lineare il flusso di lavoro. Contestualmente si eliminano quelle azioni manuali che richiedono i comuni software, come tracciare linee o ridimensionare angoli; nei modelli parametrici invece, proprio perché sono parametrici e i componenti sono connessi fra loro, ciò avviene spontaneamente. Un ulteriore beneficio del design parametrico, che lo rende molto produttivo per i progettisti, è il fatto che si possono creare forme, geometrie, modelli, aspetti, disegni, progetti, ecc., molto complessi, senza rinunciare a un grado di precisione elevato. Anche in questo modo sia l'architetto sia l'ingegnere sono maggiormente spinti a sperimentare soluzioni che un tempo erano limitate. Pertanto progettare in questo modo diventa molto più efficiente in tutti gli aspetti del processo, crea un mezzo di comunicazione molto più efficace tra le parti interessate, consente di eseguire un maggior numero di analisi prima che il processo di progettazione sia concluso, riduce notevolmente il tempo dedicato alle modifiche e il progetto può essere analizzato per verificarne la prosperità molto prima che possa essere costruito.

La seconda parte della tesi (cap. 6) si concentra sull'applicazione di queste nuove modalità a un caso specifico. È stato scelto il Toshiba IHI Pavilion di Kisho Kurokawa per una molteplicità di ragioni. In primo luogo perché rappresenta un'opera di altissimo valore artistico, che nasce da una interessante filosofia architettonica, il Metabolismo, in un contesto particolare costituito dall'Expo'70 in Giappone che fu un momento storico di grande rilievo. Il Toshiba IHI Pavilion fu il simbolo di quella stagione storica e culturale del Giappone e del movimento architettonico di cui Kurokawa fu uno dei protagonisti. In secondo luogo la struttura e la forma dell'opera sono state ideate e disegnate senza poter utilizzare quegli strumenti che il progresso tecnologico ha ora messo a disposizione.

L'applicazione al caso – studio delle tecnologie come Grasshopper per Rhinoceros ha portato ai seguenti vantaggi, che riassumo brevemente. L'utilizzo degli algoritmi in questo contesto ha permesso rapidità nella costruzione geometrica, il che ha implicato a sua volta la possibilità di avere a disposizione più scelte in ogni step del processo, potendo in ogni caso ritornare sui propri passi senza dover riprendere il lavoro daccapo ogni volta, o anche visualizzare contemporaneamente soluzioni diverse. Questa metodologia operativa ha fatto intravedere nuove modalità di progettazione con grandi potenzialità. Se finora l'intuizione estetica e le scelte artistiche di un progetto si manifestavano in schizzi preparatori con disegni manuali (con carta e penna), ora questi aspetti si esprimono in un linguaggio diverso, quello dell'algoritmo, ma portano al medesimo esito. In altri termini, con il disegno parametrico la dimensione artistica è ugualmente fondamentale nell'intuizione progettuale, per quanto si traduca nel linguaggio informatico, che impone, ancor prima di iniziare l'iter progettuale vero e proprio, di aver chiara l'idea estetica che sta alla base del lavoro. Ho, inoltre, potuto constatare che, durante lo stesso processo parametrico, è possibile, con gli strumenti forniti dal software utilizzato (nel mio caso Grasshopper), controllare e manovrare a proprio piacimento e in qualsiasi momento i dati e i parametri, modificando il progetto in modo facile e veloce. I software parametrici offrono molteplici possibilità di studio dell'edificio progettato. In questo lavoro ne ho

essenzialmente sfruttate due: la *performance* strutturale e l'ottimizzazione di obiettivi multipli. Ma tali software potrebbero anche permettere di studiare l'illuminazione, l'efficienza energetica, le simulazioni digitali, la fabbricazione, permettendo una migliore qualità del prodotto finale.

Nel contesto dell'architettura contemporanea, il design parametrico non rappresenta solo un'innovazione tecnica, ma una vera e propria rivoluzione culturale e metodologica. Esso offre la possibilità di esplorare nuovi paradigmi estetici e funzionali che, senza tali strumenti, sarebbero rimasti inaccessibili o impraticabili. Tuttavia, questa trasformazione impone una riflessione critica sugli strumenti stessi e sulle implicazioni del loro uso, infatti l'introduzione di algoritmi complessi e di software avanzati non è priva di sfide. Pertanto, il futuro del design parametrico potrebbe vedere una sinergia sempre più raffinata tra capacità umane e tecnologiche, dove l'architetto non si limita a essere un semplice operatore di software, ma diventa un orchestratore di processi complessi. Questi aspetti aprono nuovi scenari per la ricerca e la pratica architettonica, invitando a esplorare come le intuizioni creative possano interagire con le capacità analitiche dei nuovi strumenti per dare vita a opere innovative e significative.

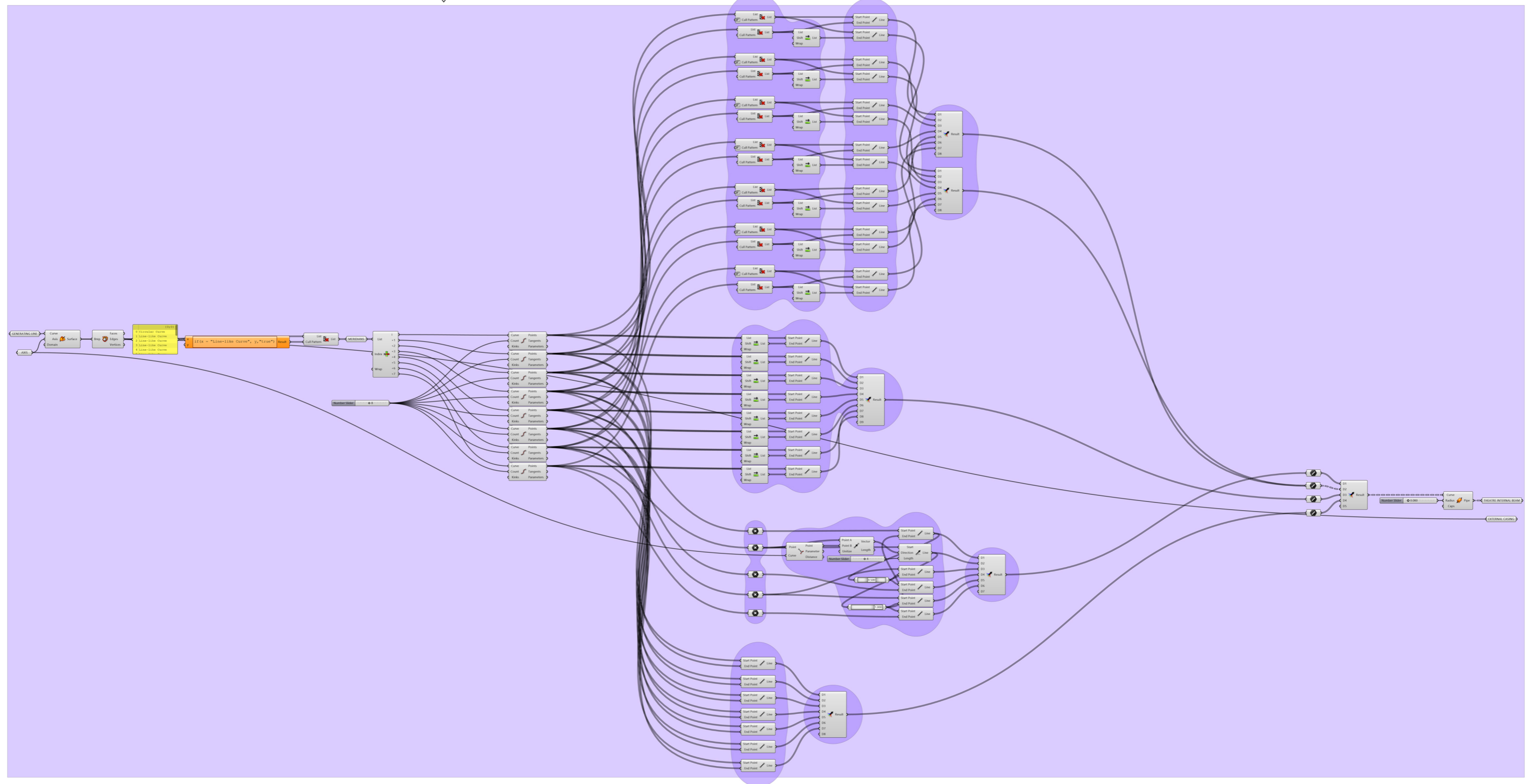
Per quanto riguarda le conclusioni specifiche dell'applicazione dei procedimenti parametrici al caso preso in esame si veda il paragrafo 6.3.2.



# APPENDICI

# THEATER

# INTERNAL BEAM

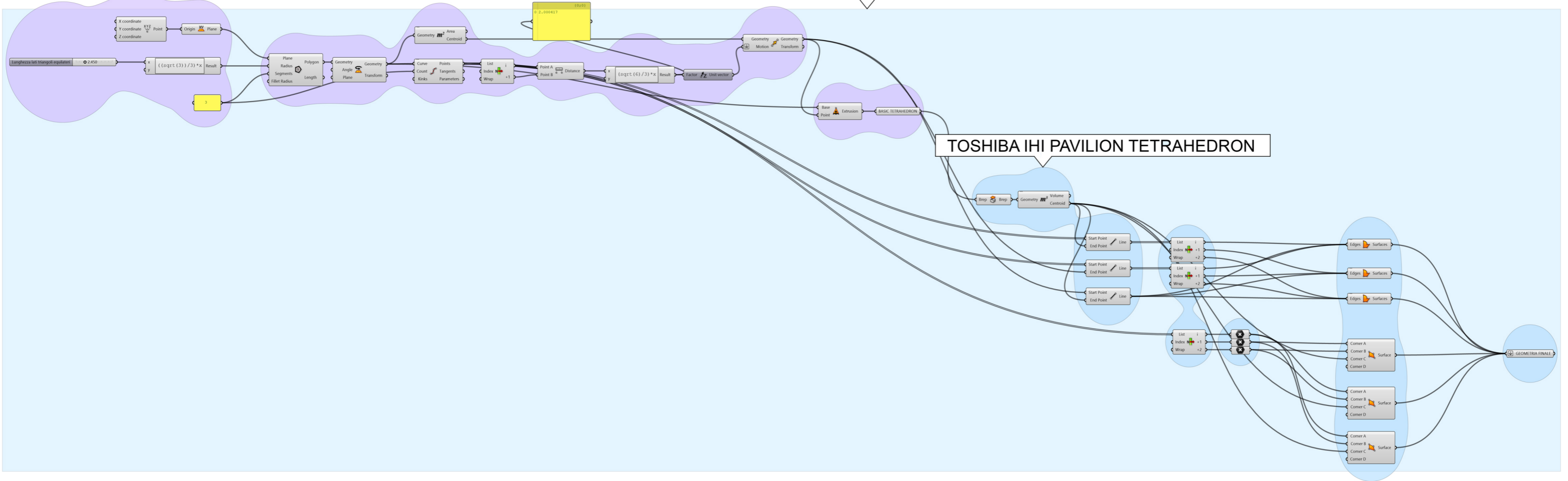


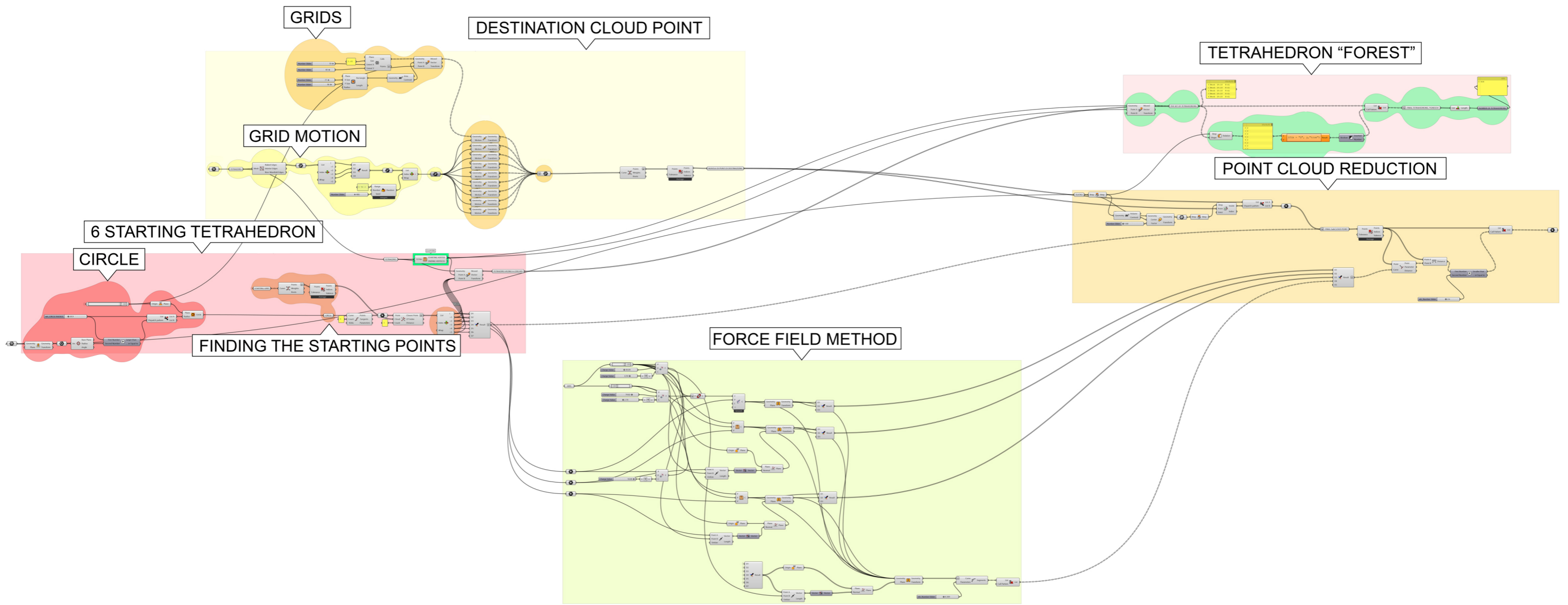


# REGULAR TETRAHEDRON

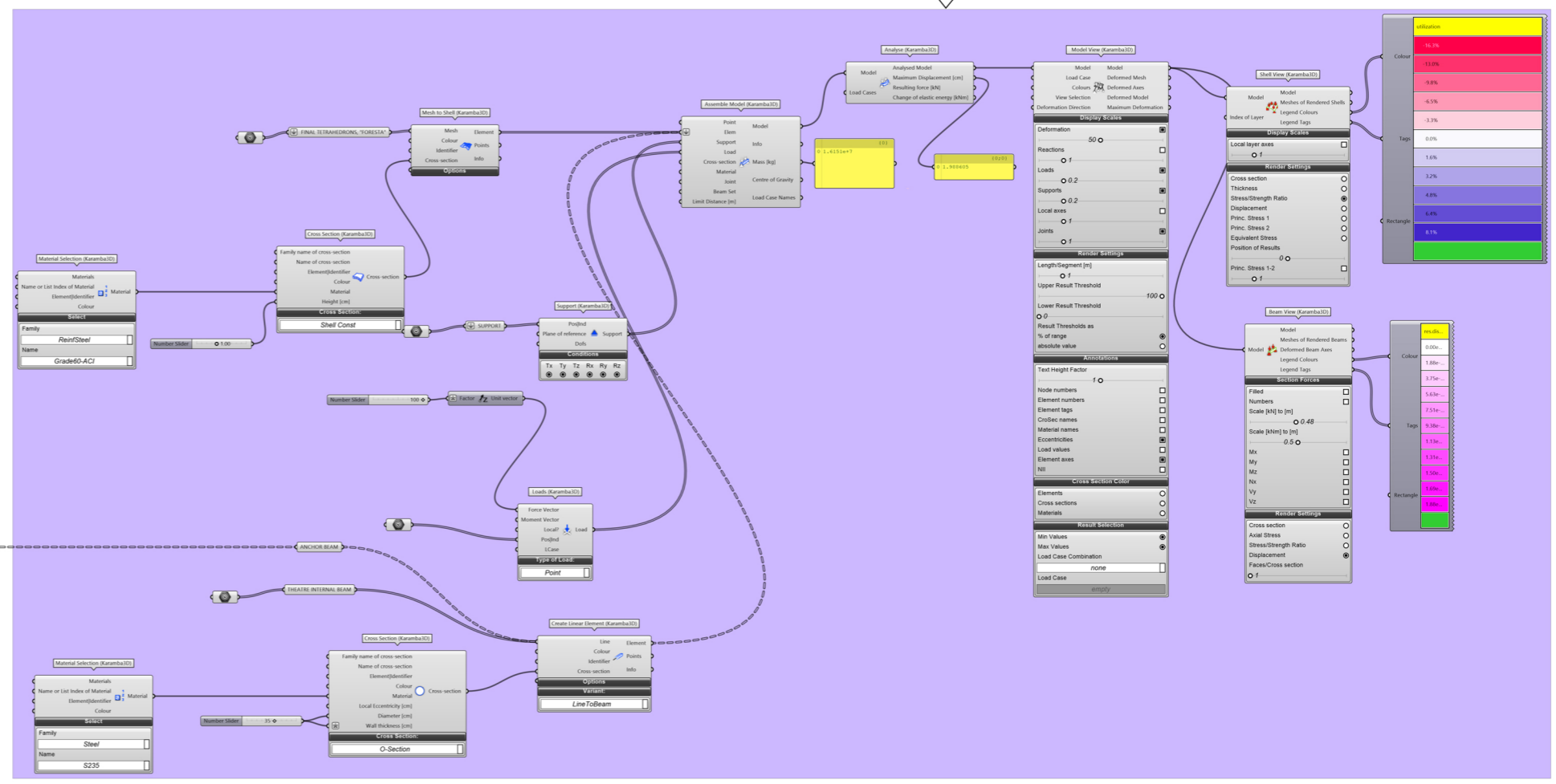
# TETRAHEDRON

# TOSHIBA IHI PAVILION TETRAHEDRON

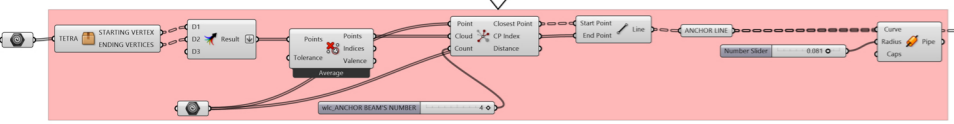




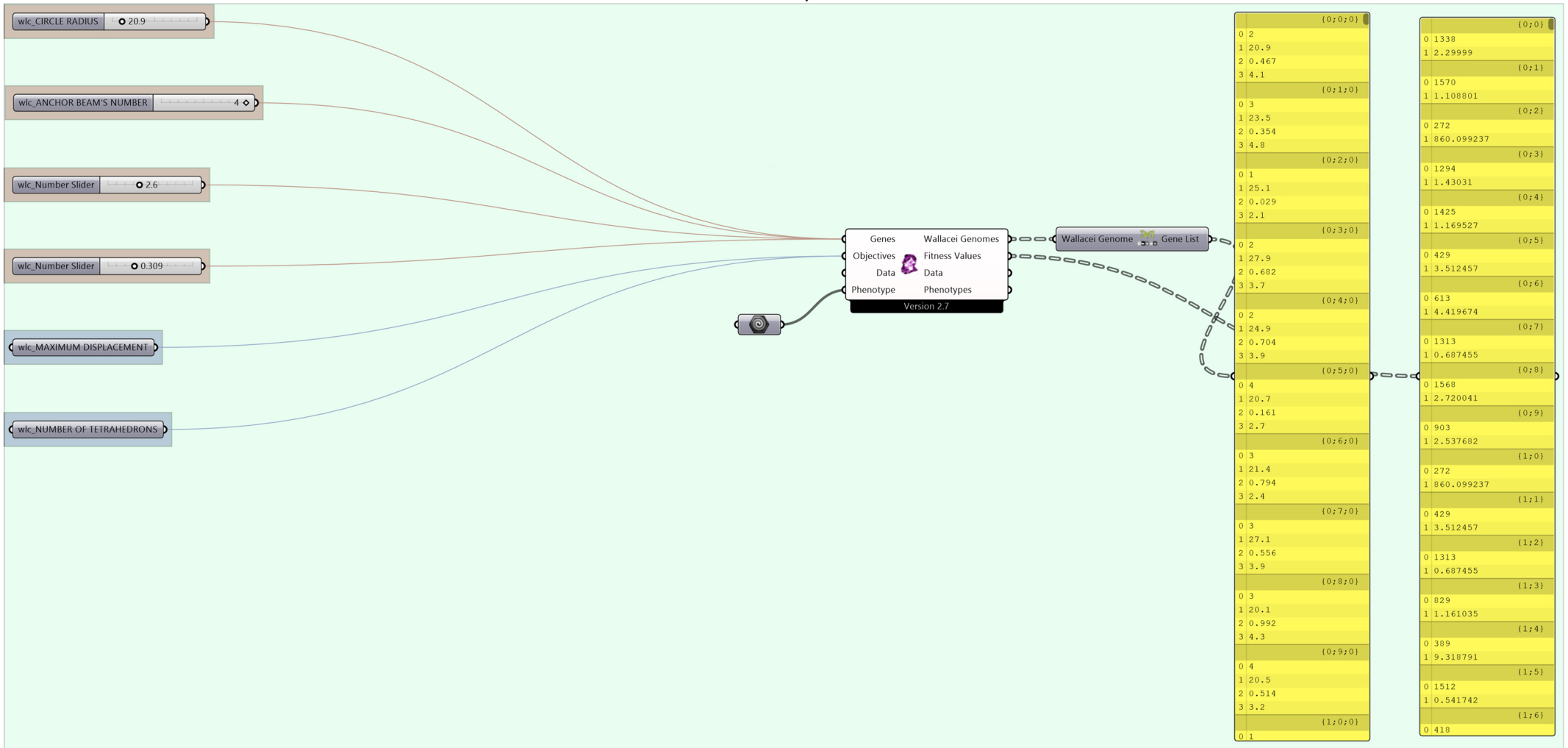
# KARAMBA3D ANALYSE



# ANCHOR POINTS



# WALLCEI OPTIMIZATION



# 08 • Bibliografia

- R. G. Alvarado, A. Lyon, *Diseño paramétrico en Arquitectura; método, técnicas y aplicaciones*, ARQUISUR, 1, 2013.
- C. Arbib, *Ottimizzazione* in [https://www.treccani.it/enciclopedia/ottimizzazione\\_\(XXI-Secolo\)/](https://www.treccani.it/enciclopedia/ottimizzazione_(XXI-Secolo)/), 2010
- M. P. Bendsøe, P. Martin, O. Sigmund, *Optimization of structural topology, shape and material*, Vol. 414, Berlin, Springer, 1995.
- M. P. Bendsøe, O. Sigmund, *Topology Optimization: Theory, Methods, and Applications*, Springer Science & Business Media, Berlin, 2013.
- R. A. Bernardello, P. Borin, A. Giordano, F. Panarotto, *From Historical and Theoretical Analysis of Representation and Geometry to Topology for Structural Optimization*, In Ródenas-López, M.A., Calvo-López, J., Salcedo-Galera, *Architectural Graphics*, EGA 2022.
- A. A. Bertossi & M.A. Bonuccelli, *Code assignment for hidden terminal interference avoidance in multihop packet radio networks*, IEEE/ACM Transactions on Networking, vol. 3, n. 4, 1995, pp. 441-449.
- G. E. P. Box, N. R. Draper, *Evolutionary Operation*, John Wiley & Sons, Inc., New York, 1969.
- J. Bremermann Hans, *Optimization through Evolution and Recombination*, In M.C. Yovits, G.T. Jacobi, G.D. Goldstein G.D. (a cura di), *Self-Organizing Systems*, Spartan Books, Washington D. C., 1962.
- C. Ceccato, *MICROGENESIS. The Architect as Toolmaker: Computer-Based Generative Design Tools and Methods*, CAADRIA proceedings., 1999.

- J. Cichocka, W. N. Browne, R. R. Edgar, *Optimization in the Architectural Practice - An International Survey*, caadria, 2017, p.387.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduzione agli algoritmi e strutture dati*, McGraw Hill, 2010.
- D. Davis, J. Burry, M. Burry, *Understanding Visual Scripts: Improving collaboration through modular programming*, *International Journal of Architectural Computing*, Vol. 9, N° 4, Londres: Multiscience, 2011, pp. 361–375.
- *Design parametrico e progettazione algoritmica* (FabLab Venezia), Disponibile Online: <https://www.fablabvenezia.org/design-parametrico-e-progettazione-algoritmica/>. [Consultato il giorno 19 febbraio 2024]
- L. M. García Ballesteros, *La parametrización del espacio procesos de diseño paramétrico*, Escuela superior técnica de arquitectura de Madrid, 2017.
- M. Gestal, *Introduccion a los Algoritmos Geneticos*, Universidade da Coruña, España, 2013.
- A. Giordano, *Cupole volte e altre superfici. La genesi e la forma*, UTET, Torino 1999.
- A. Giordano, G. D'Acunto, *Descrivere lo spazio*, Italia: Progetto Libreria, 2017.
- A. Giordano, R. A. Bernardello, C. Monteleone, P. Borin, *The Teaching of Surfaces, Between Geometry and Stereotomy*, In: Hermida González, L., Xavier, J.P., Amado Lorenzo, A., Fernández-Álvarez, Á.J. (eds) *Graphic Horizons. EGA 2024*, vol 43.
- F. Giovanardi, *Strutture fuori dal coro*, Giovannardierontini, 2010.
- D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co. Inc., Boston, 1989.
- S. Kalantari, S. Tajik, F. Deambrosis, *Japanese Architecture after World War II: Significant events and Ideologies from 1955-1970*, Master of Science in Architectural Design, Politecnico di Milano, 2017.

- D. Knuth, *The art of computer programming*, volume 1, *Fundamental algorithms*, Addison-Wesley Professional, Massachusetts, 1968.
- S. Köhn, *Presentation for the Japanology seminar "Resistance in Japan"*, University of Cologne, 2 Nov 2021).
- L. Kubiak, L. Sgambi, *Why architects should learn to code. The 7th International Conference on Architecture and Built Environment*, proceeding of the 7th International Conference S.ARCH, 7-9 April 2020, Tokyo, Japan, 2020.
- K. Kurokawa, *Metabolism in Architecture*, Studio Vista, Londra, 1977.
- K. Kurokawa, *The philosophy of symbiosis*, Academy Editions, Londra 1994.
- R. Issa, *Essential Algorithms and Data Structures for Computational Design*, Robert McNeel & Associates, 2020.
- Y. H. Lee, *A Study on the Concepts in Kisho Kurokawa's Architecture*, Journal of the Korea Academia-Industrial cooperation Society, Vol. 16, No. 1, 2015, pp. 827-836.
- E. Martín-Gutiérrez, *El movimiento metabolista: Kisho Kurokawa y la arquitectura de las cápsulas*, *Boletín Académico*, Escola Técnica Superior de Arquitectura da Coruña, 1990.
- D. Madeddu, *Architetture genetiche. Gli algoritmi genetici come strumento di ottimizzazione in architettura*, Ricerche di Architettura, Gangemi, 2011.
- F. Manara, G. Lucchini, *L'ottimizzazione: dimensioni storico-culturali e matematiche*, 2013. Disponibile in: <https://www.carlofelicemanara.it/public/file/File/Biografia/Ottimizzazione%20dimensioni%20storico-culturali%20e%20matematiche%202013.pdf> [data di accesso: 27/08/2022]
- K. Martini, *Optimization and Parametric Modelling to Support Conceptual Structural Design*, *International Journal of Architectural Computing*, 2011.

- M. Masquelier, *Kisho Kurokawa and the Metabolists, Paving the Way of Tokyo's Post-War Architecture*, Undergraduate History Journal at Illinois, 2022.
- T. J. Moleta, *Exposition, Societal Responsitivity and the Aesthetics of Impermanence: Temporal Findings from the 1970 World Exhibition*, Master degree, Victoria University of Wellington, 2010.
- S. Navarrete, *Diseño paramétrico: El gran desafío del siglo XXI*, Cuadernos del Centro de Estudios en Diseño y Comunicación, Ensayos, n.49, 2014, pp.63-72.
- S. Oh, Y. Jung, S. Kim, I. Lee, N. Kang, *Deep Generative Design: Integration of Topology Optimization and Generative Models*, ASME, Journal of Mechanical Design, 2019.
- C. Preisinger, *Linking Structure and Parametric Geometry*, Architectural Design, Vol. 83, Issue 2, 2013
- P. Schumacher, *Parametricism Manifesto*, in [www.patrickschumacher.com](http://www.patrickschumacher.com), Londres, 2008.
- M. A. Schnabel, *Parametric Designing in Architecture*, A. Dong, A.V. Moere, J.S. Gero, *Computer-Aided Architectural Design Futures (CAADFutures)*, Springer, Dordrecht, 2007.
- Y. C Sebnem, O. A. Fulya, Y. Tugrul, *Computational design, Parametric modelling and architectural education*, 2008.
- O. Sigmund, K. Maute, *Topology optimization approaches: A comparative review*, *Structural and Multidisciplinary Optimization*, Vol. 48, No. 6, University of Colorado, 2013, pp.1031-1055.
- S. N. Sivanandam, S. N. Deepa, *Introduction to Genetic Algorithms*, Springer Berlin Heidelberg New York, 2008, pp. 20-30, 33- 35.
- L. Sui, *Spatial Intelligent Design and Innovation for Indoor Environment - Visualization and Analysis of Web3D Technology*, Applied Mathematics and Nonlinear Sciences, 9, 2024.



- A. Tedeschi, *Architettura parametrica. Introduzione a Grasshopper: il plugin per la modellazione generativa in Rhino*, Le Penseur, Brienza (PT), 2010.
- A. Tedeschi, S. Andreani, F. Wirz, *AAD\_Algorithms-aided design: Parametric strategies using grasshopper*, Le Penseur Publisher, Brienza (PT), 2016.
- K. Terzidis, *Algorithmic Architecture*, Architectural Press, Oxford, 2006.
- A. Tettamanzi, *Algoritmi Evolutivi: Concetti e Applicazioni*, Mondo Digitale, Fascicolo 1, 2005, pp. 3-17.
- R. Vierlinger, A. Hofmann, *A Framework for Flexible Search and Optimization in Parametric Design*, Deesign Modelling Symposium, Berlin, 2013.
- J. I. Zarategui, *Intelligent design objects applied to the spatial allocation problem*, Master Degree, Middle East Technical University, 2014.
- [https://www.treccani.it/enciclopedia/ottimizzazione\\_\(XXI-Secolo\)/](https://www.treccani.it/enciclopedia/ottimizzazione_(XXI-Secolo)/)



## *Ringraziamenti*

*Ho scritto buona parte di questa tesi durante il mio soggiorno in Spagna presso l'Universidad Politécnica de Cartagena seguito dal professor Manuel Alejandro, Ròdenas Lòpez. L'idea della tesi e di questa collaborazione è stata del professor Andrea Giordano. Al professor Giordano e al professor Rodenas, che hanno sempre dimostrato gentilezza e attenzione vanno i miei più sentiti ringraziamenti.*

*Desidero esprimere la mia gratitudine alla mia famiglia, che ha reso possibile il mio percorso di studi.*

*Un ringraziamento speciale va ai miei amici di sempre, che mi hanno accompagnato in questo lungo viaggio, offrendomi il loro affetto e la loro vicinanza; ai miei compagni di corso, che sono diventati veri amici lungo il cammino. Condividere con voi le gioie e le sfide di questi anni ha reso tutto più leggero e gratificante.*

*Infine, un pensiero affettuoso va a Pietro, la cui presenza è stata per me fondamentale. Grazie per essere stato al mio fianco, sempre.*