

Algoritmos evolutivos como modelo propositivo de diseño

Evolutionary algorithms as an innovative design model

Carlos Ignacio de la Barrera Poblete*

<Resumen>

Las respuestas arquitectónicas vinculadas al emergente mundo virtual, descubren un cambio a la hora de enfrentarse y finalmente resolver los problemas de diseño. Una arquitectura generadora de procesos y no de hechos finitos. Un sistema de respuestas flexibles, capaces de mutar y modificarse, incorporando decididamente el manejo del tiempo en pos de la autotransformación.

<Abstract>

In Architecture there have been changes in the way design problems are resolved, as a response to the expansion of the «virtual world». Architecture now deals with on going processes rather than limited facts and objects. Architecture has become a system of flexible response that is able to change and modify itself. Time is a key variable in this process of auto transformation.

<PALABRAS CLAVE>

DISEÑO GENERATIVO / ESTRATEGIA DE EMERGENCIA /
ALGORITMOS EVOLUTIVOS / SISTEMA AUTOORGANIZADO
/ DISEÑO EXPANSIVO

<KEY WORDS>

GENERATIVE DESIGN / EMERGENCY STRATEGY /
EVOLUTIONARY ALGORITHM / AUTO ORGANIZED SYSTEM
/ GROWING DESIGN

Antecedentes

La evolución de las especies es la historia de los procesos que han sucedido desde la aparición de la vida en el planeta hasta la actualidad. Esta historia representa una serie de cambios, que buscan solucionar una infinidad de problemas. Su éxito está basado en mecanismos que, actuando de una manera aleatoria, consiguen resolver estos problemas. Durante la década de los 50 y principios de los 60, programadores de computadoras y biólogos comenzaron a buscar, de manera experimental, cómo realizar modelos representativos de la evolución natural a través de sistemas informáticos. En 1965, Ingo Rechenberg, de la Universidad Técnica de Berlín, introdujo una técnica que denominó estrategia evolutiva, la cual plantea

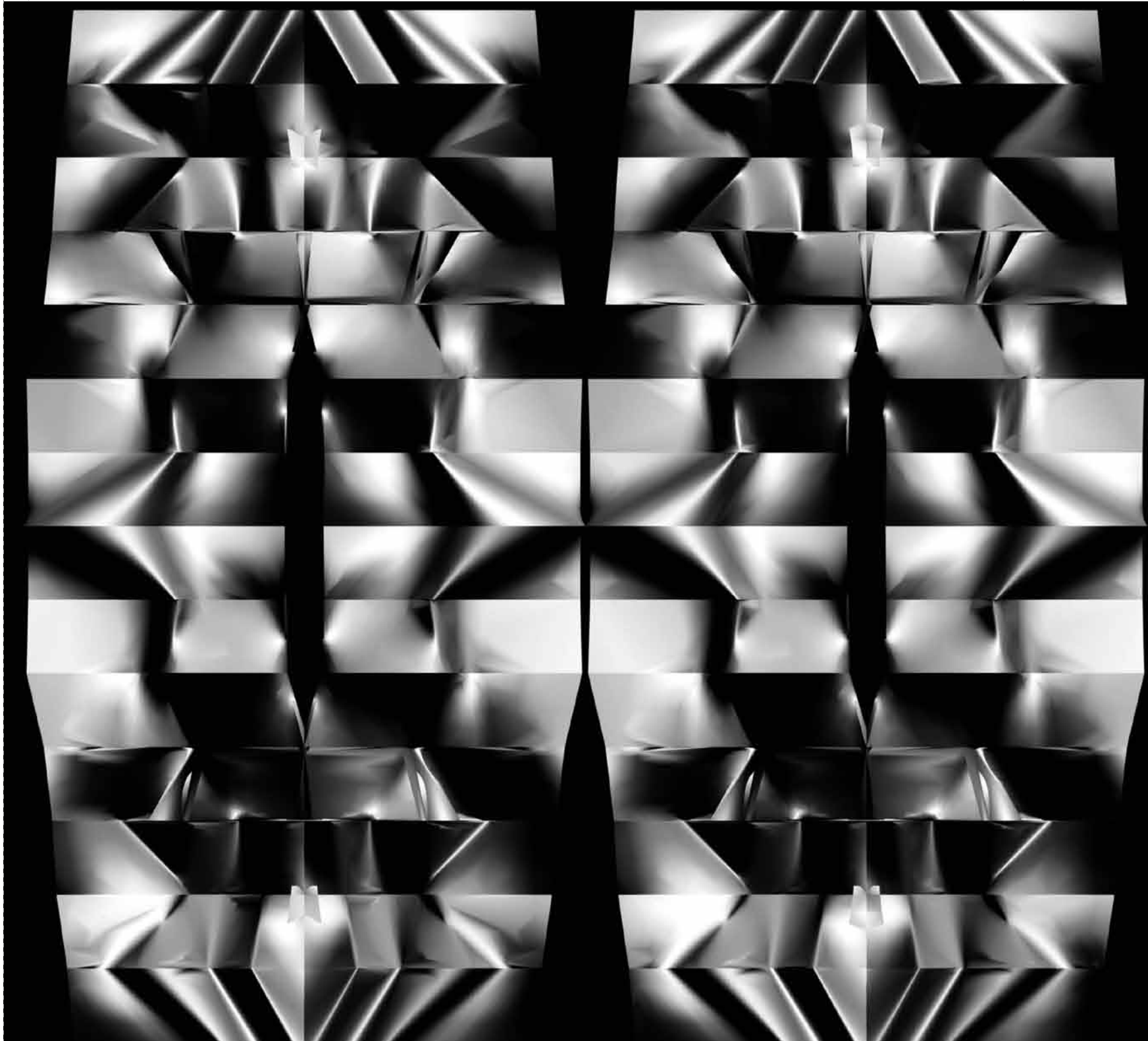
una técnica de programación que imita a la evolución biológica como estrategia para resolver problemas. El sistema de Rechenberg se basaba en un padre, el cual mutaba para producir un descendiente, y que luego de una evaluación (selección) se conservaba el mejor de los dos, convirtiéndose en el padre de la siguiente ronda de mutación. Versiones posteriores del experimento de Rechenberg introdujeron la idea de población.

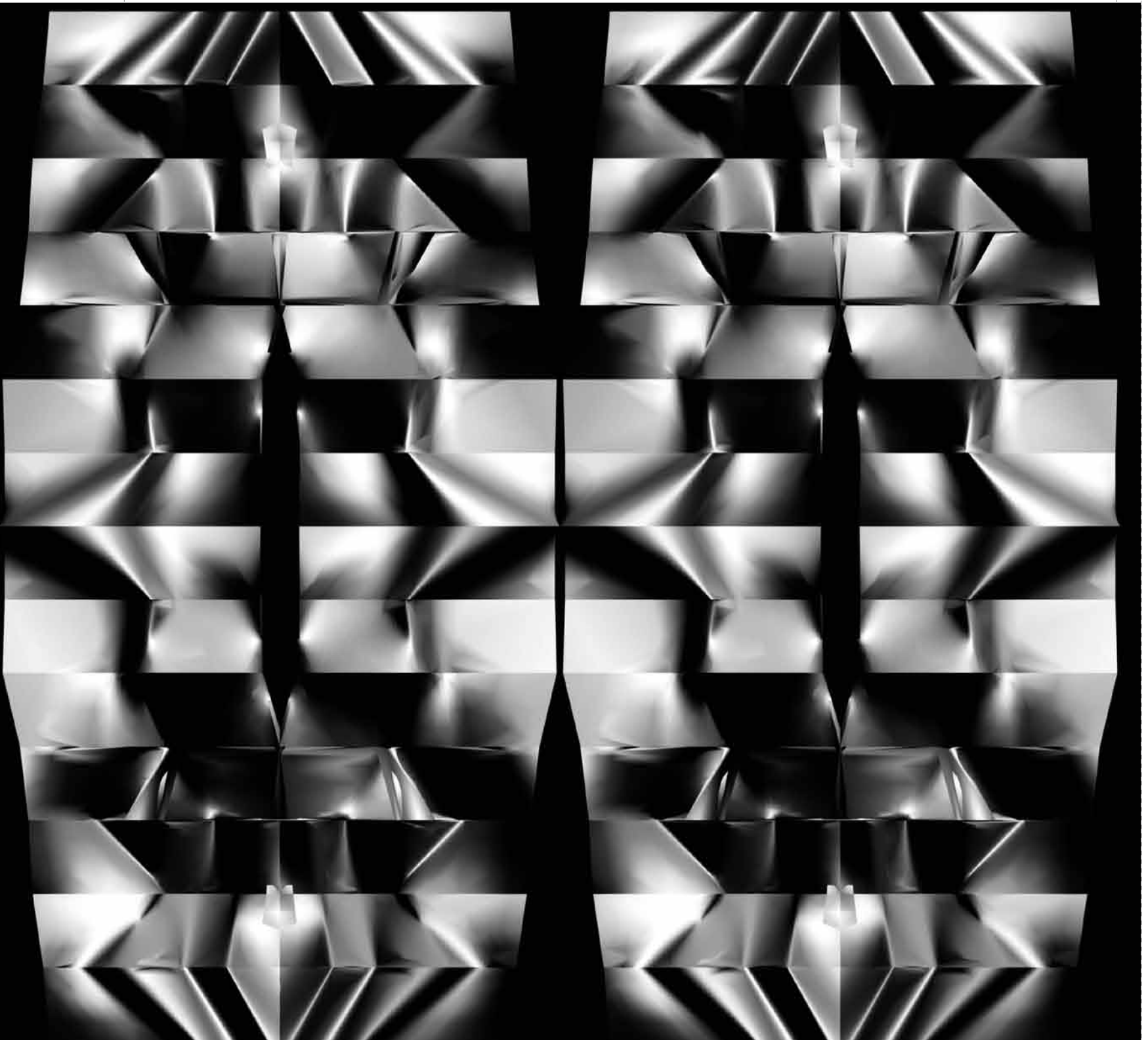
Otras fechas importantes en el estado del arte son:

1962: John Holland trabaja sobre sistemas adaptativos y establece las bases para desarrollos posteriores. Propone el cruzamiento y otros operadores de recombinación.

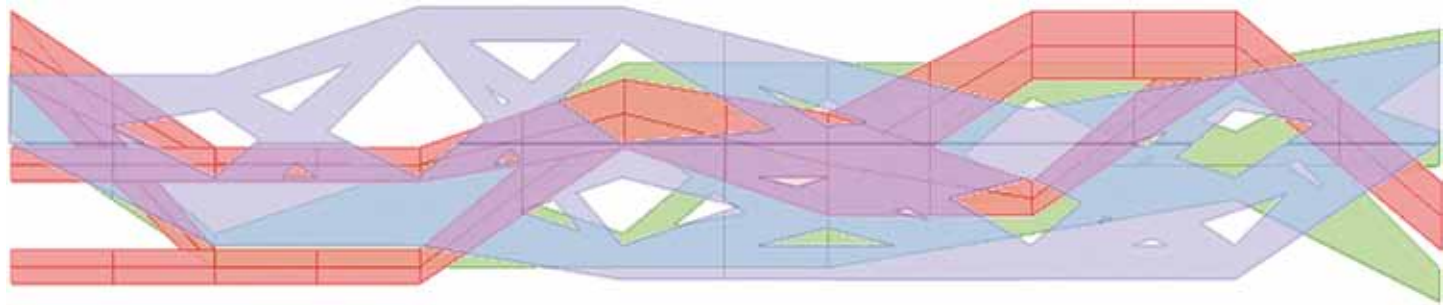
* M. Arch. Ph.D. candidate. Escola Tècnica Superior D'Arquitectura, Universidad Internacional de Catalunya.

Algoritmo 2.





a + b + c.



1966: J. Fogel, A. Owens y M. Walsh introducen en América una técnica bautizada programación evolutiva, la cual consistía en la representación de máquinas. Su algoritmo funcionaba mutando aleatoriamente una de estas máquinas simuladas, eligiendo la más apta para continuar.

1975: John Holland, publica el libro *Adaptación en sistemas naturales y artificiales*, basado en sus investigaciones en la Universidad de Michigan, siendo el primero en postular el concepto de sistemas digitales adaptativos simulando el proceso de la evolución biológica como estrategia para resolver problemas.

A éstos, siguieron otros experimentos en áreas tan diversas como astronomía, química, mercados financieros, juegos, geofísica, ingeniería de materiales, matemáticas, biología molecular, estadística, robótica, diseño de rutas y horarios, todos con mejores resultados que los alcanzados por diseñadores humanos. En 1966, debido a la similitud con el área de la arquitectura (en su búsqueda de una forma eficiente para resolver problemas estructurales), cabe destacar el trabajo realizado por Keane y Brown en el campo de la Ingeniería aeroespacial. En dicho trabajo se utilizó un algoritmo para diseñar un brazo de grúa para transportar y montar satélites, estaciones espaciales y otros proyectos de construcción aeroespacial. El resultado fue una estructura orgánica similar al fémur humano. Desde hace algunos años, la arquitectura ha comenzado a ser planteada como un proceso matemático. Esta investigación se centra en la programación de reglas aritméticas y geométricas que imitan a la evolución biológica como estrategia para resolver problemas estructurales y de arquitectura. Al mismo tiempo se busca desarrollar la fabricación utilizando las más recientes tecnologías de producción digital, transmitiendo los datos del ordenador a máquinas CNC (*computer numeric control machines*), que reciben la

información y la fabrican, en un proceso que traerá grandes modificaciones en la forma que hacemos arquitectura. De este modo la concepción, el diseño y la fabricación son parte de un proceso de acción continua. El futuro será fabricar los propios materiales de acuerdo a necesidades y con la ayuda de un equipo multidisciplinar.

Objetivos

El método de trabajo es la experimentación de sentencias programadas en C++ como proceso generativo de diseño. El material se autoorganiza dependiendo de reglas y de su interacción con elementos macroambientales incorporados en la programación. La forma digital es constituida por el material, que es investigado de manera experimental con el fin de encontrar propuestas arquitectónicas. Las formas obtenidas sustituyen la organización espacial acostumbrada, a cambio de un desorden aparente del espacio. Utilizando algoritmos computacionales basados en estrategias evolutivas, se busca desarrollar formas arquitectónicas flexibles, basado en dos series de experimentos. Los primeros intentos buscan organizar el espacio según relaciones propias de la arquitectura, como lo es la jerarquía espacial. De este modo el algoritmo será capaz de relacionar tres escalas espaciales diferentes. Una primera escala considerada como espacio íntimo, para un grupo de personas reducido. Una segunda escala intermedia para un grupo de mayor volumen y finalmente un macroespacio. En la segunda serie de experimentos, el algoritmo es condicionado a responder a elementos que representen un ambiente biológico real. De esta manera los resultados obtenidos están relacionados no con el proceso interno del algoritmo sino con elementos incorporados en la programación, pero que influyen en el cómputo generativo modificando el crecimiento del proyecto.

Desarrollo

El estudio utiliza una regla repetitiva de expresión geométrica y aritmética, basada en el movimiento del caballo en el ajedrez, como desplazamiento continuó dentro de una grilla dividida en cuadrados de iguales dimensiones. Cada salto es una iteración del algoritmo, y hace que un gen iniciador mute, variando su información genética en su cromosoma. Este algoritmo evolutivo es utilizado como un explorador del espacio, que tiende a moverse según un ambiente preestablecido en la programación. En un inicio se definen las dimensiones de la grilla dividida en cuadrados de iguales dimensiones. La densidad de la trama está relacionada con el nivel de precisión que se requiera para cada solución. De esta manera una trama más densa generara mayor cantidad de información que una trama de menor densidad. Los límites de la grilla pueden no tener límites ortogonales entre sí, permitiendo abarcar superficies de diferentes morfologías. La grilla es definida por coordenadas en X e Y.

El siguiente paso es posicionar un valor cualquiera, representado por la letra A en el centro de un cuadrado conformado por la grilla. Este punto inicial es dado por el cultivador, y será el inicio por donde el algoritmo comenzará a explorar el espacio.

La posición de cualquier punto para un campo regular cuadrado compuesto por una grilla de 8 x 8 será:

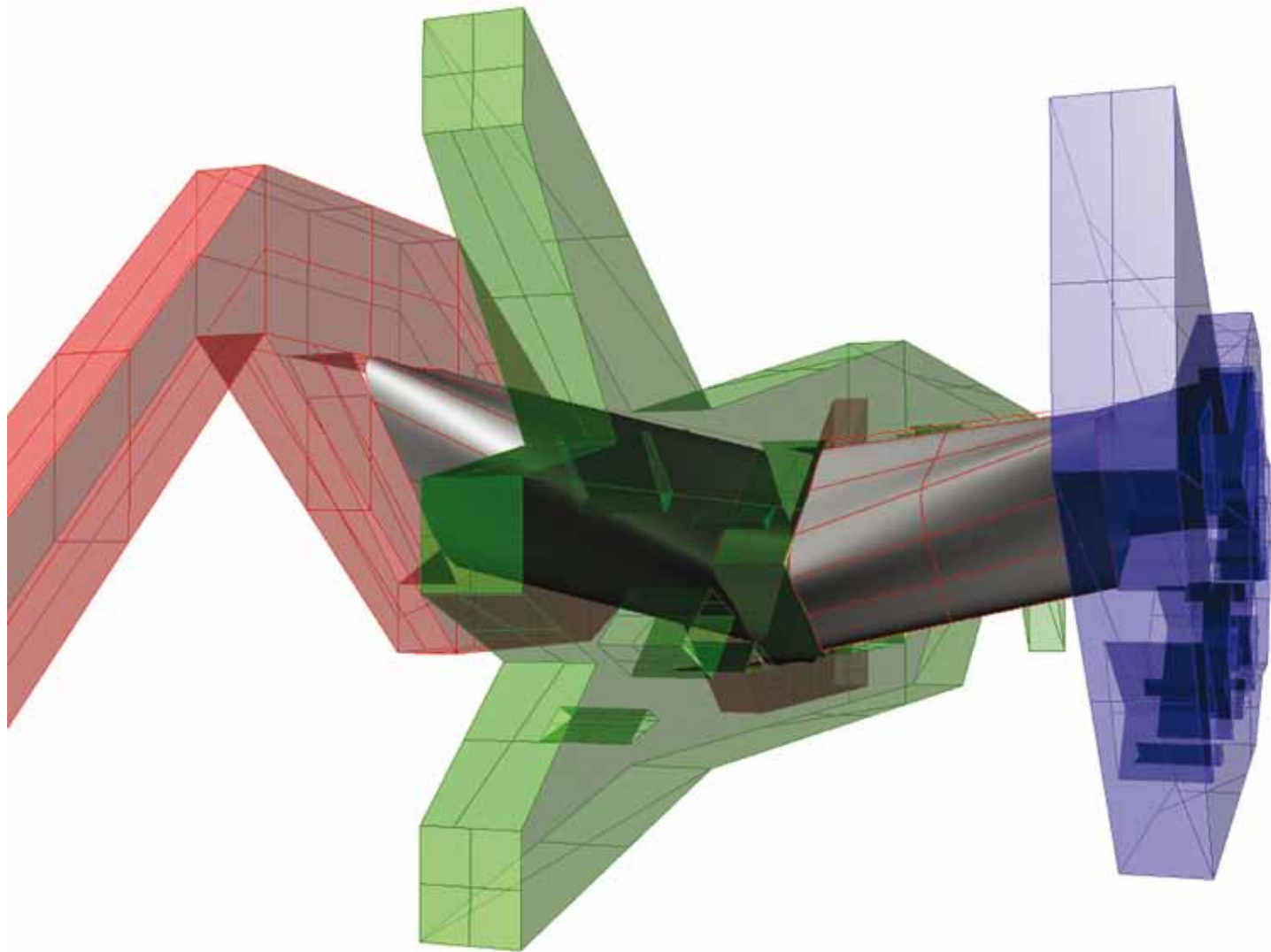
$$A = \text{eje } X (0 \leq X \leq 7), \text{ eje } Y (0 \leq Y \leq 7).$$

La posición del siguiente valor (B) en la coordenada X será:

$$B = ([X \times -1] \times -1).$$

Si $X = 1$ el resultante de B será 1 por lo tanto situará a B en el siguiente eje paralelo en Y donde existen (en este caso) dos posibilidades, en cambio, si $X = 2$ situará a

a + b + c (relación).



B en el subsiguiente eje paralelo a Y donde también existen dos posibilidades, en cambio si $X = -1$ ($[-1 \times -1 = +1] \times -1$) será el eje paralelo en Y anterior donde se encuentre el valor inicial representado por la letra A.

Para el eje Y será: $([Y \times -1] \times -1)$.

Se definirán 2 ejes horizontales (X) y 2 ejes verticales (Y) a cada lado del último valor marcado por el algoritmo, con 16 intersecciones en total. Las 8 intersecciones que se encuentran en los dos ejes diagonales que tienen como intersección el último valor marcado se anularán con valor «0» quedando solo como resultado las posibles casillas donde puede caer el algoritmo. Si el valor inicial se encuentra en una esquina del campo es

posible que solamente tenga dos o una sola opción de caer, pero si se encuentra en el centro del campo donde los límites están por fuera de su alcance de salto, sus posibilidades máximas son, 8. En este último caso, el siguiente número de posibilidades como máximo será 7.

El valor puesto inicialmente busca de una manera aleatoria donde caer en la siguiente casilla, la evaluación comienza identificando las posibles alternativas, enumerándolas en sentido contrario del reloj: (Posición de B) = $X \leq 7$, para la primera iteración, donde B es el valor de la siguiente casilla. X es el número que representa cada una de las posibles casillas, que siempre será menor o igual a 7. El valor en X es obtenido por un «motor» en

la programación y su resultado que va de 1 a 8 (máxima cantidad de casillas) es totalmente aleatorio. Para la siguiente iteración la fórmula es: (Posición de C) = $X \leq 7$. Siguiendo este proceso el algoritmo explora el espacio. Cada vez que se marca un valor dentro de la grilla, esta casilla es fijada y no puede cambiar su estado por otro. Esto hace que el proceso de generación tenga un fin cuando el algoritmo no encuentre más casillas a su alcance para poner un valor. Las casillas vacías se representan con valor «0» porque no existe valor en ellas.

La primera condición permite anular ciertas casillas de la grilla, identificándolas antes del proceso generativo como valor «0», de este modo, si una posible casilla

ha2 - 4.



está anulada, el algoritmo no pone valor en ella, teniendo que elegir una de las otras candidatas. Esta condición es permanente durante todo el proceso generativo, y es utilizada para posicionar cargas verticales en la estructura que ha emergido del proceso. De esta manera, existen dos formas de obtener un valor «0» (casilla vacía) en la grilla. La primera manera es que el algoritmo no encuentre intersecciones en sus ejes X e Y por lo que se detendrá. Y las casillas que no se encuentren con una marca, tendrán valor cero. Estos resultados, pueden ser aleatorios, o ser controlados en parte por la segunda condición. La segunda manera es determinar los casilleros con valor «0» (casilla vacía) antes del proceso.

La segunda condición es aumentar las posibilidades de selección de una cierta cantidad de casillas en un área determinada. Esta condición sirve para dirigir de una manera parcial el recorrido del algoritmo. Para 8 casillas las posibilidades repartidas equitativamente son el 12,5% para cada una, porque el algoritmo está eligiendo de entre las 8 candidatas 8 posibilidades ($[Casilla] = X \leq 8$).

Para el mismo número de casillas, las posibilidades entre ellas pueden variar: ($[Casilla] = [1 \leq X \leq 6]$), en este caso el algoritmo está eligiendo entre 6 valores (el 16,6% de posibilidades entre las 6 candidatas), porque ha excluido las dos posibilidades que tienen el nombre de ubicación 7 y 8. Por lo tanto como el sentido

de evaluación es en contra del sentido del reloj, estamos tendiendo a orientar el desplazamiento hacia el área superior derecha del campo. Esta condición es aplicada según iteración, por lo que cuando el valor va de A-B puede evaluar 6 candidatas pero de B-C puede evaluar nuevamente 8.

Las condiciones externas, son utilizadas una vez que el proceso generativo ha terminado, éstas determinan nuevas reglas, de por ejemplo, agrupación y asociación, control de proporciones y cambios de capa en el eje Z. El tipo de resultado de este proceso es una larga sucesión de datos, y se pueden interpretar de dos maneras básicamente.

Data de *single* generaciones:

En este tipo de generaciones la data obtenida es entendida como un todo, y como resultado es una sola entidad.

- ACCCBABAAAACABCBBBACCCBCCBBOACCAOBBOCABBAAO0ABBCCACBAAABACBCB

Data de múltiples generaciones:

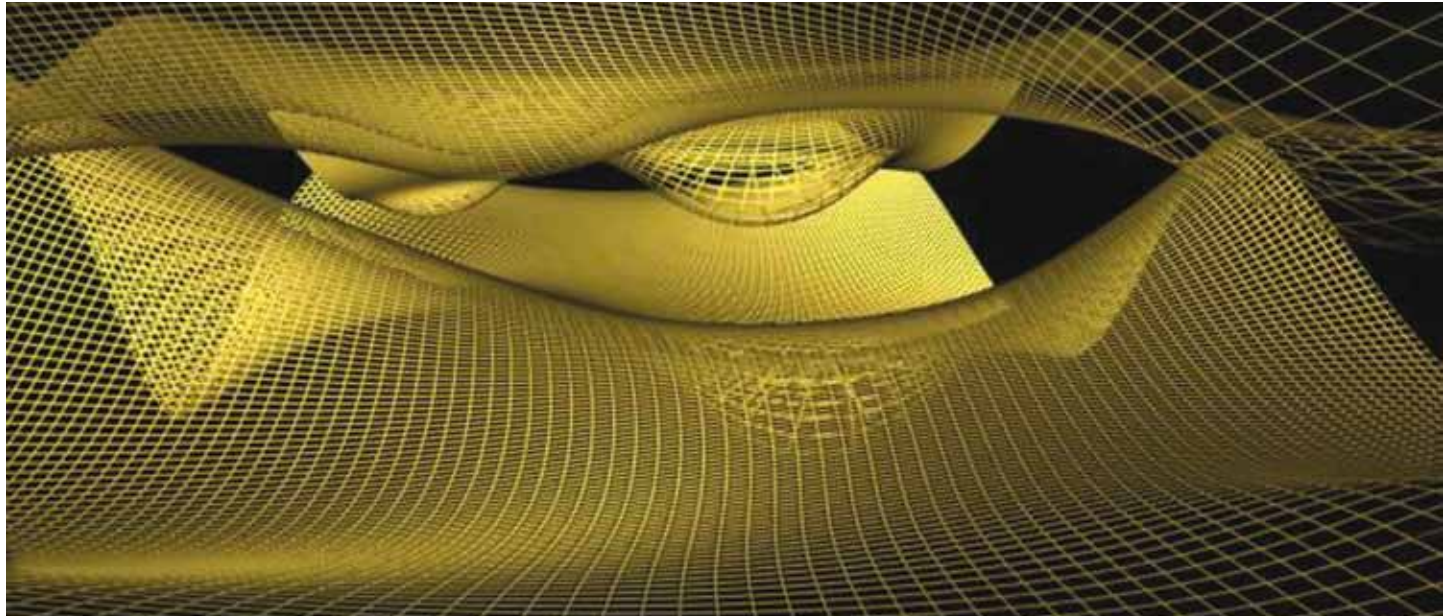
- Línea 1 (ACCCBBAB);
- Línea 2 (AAAOCABC);
- Línea 3 (BBOOCCCB);
- Línea 4 (CCBBOAAC).

En este caso los resultados son analizados por capa o sección, y cada línea obtenida es una diferente posibilidad.

Observaciones

Utilizar el ordenador como modelo propositivo de diseño implica crear una programación que tras «N» iteraciones, producirá un resultado inesperado. Esta forma de crear sugiere cambiar nociones de control y composición, por ideas como reacciones a fuerzas y procesos de crecimiento. El diseñador no tiene control sobre los procesos, más bien pasa a cultivar los diseños. El algoritmo evolutivo imita la evolución biológica como estrategia para resolver

Superficie relación.



problemas de diseño. Su respuesta inesperada y sin intervención directa de un diseñador, es una familia de formas con pequeñas variaciones entre ellas, donde cada miembro es una posible solución al problema. El Cómputo Generativo depende de sus reglas, y en este sentido es tan genuino como el comportamiento de cualquier sistema biológico natural.

Los procesos basados en la imitación de la naturaleza, tienen como resultados características biológicas observables a simple vista. Por ejemplo, si un proceso ha sido condicionado a desarrollarse en una zona de la grilla (siguiendo el sol), es posible detectar valores en las casillas que se encuentran en el opuesto de esta zona. Entonces, ¿Por qué el algoritmo, que si bien tendió a desarrollarse más en una zona dada, puso valores en la zona opuesta? Ciertas plantas, desarrollan la mayoría de sus ramas donde reciben más sol, sin embargo también hacen crecer ramificaciones en donde hay oscuridad. Esta estrategia señala que la planta está en un ambiente en constante cambio, porque si el sol cambia de posición e ilumina la zona en penumbra, la planta estará preparada para ello. Por lo tanto la naturaleza concibe sus formas como inacabadas, porque para mantenerse viva, debe ser capaz de reaccionar a los cambios externos de su entorno en el tiempo.

La naturaleza es observada como patrones geométricos y aritméticos de repetición cíclica, como un algoritmo simple, que tras

cada iteración desarrollan o incorporan aleatoriamente una variación en su programa natural de autoorganización. Esta estrategia es la clave de la subsistencia de una especie.

Conclusión

El arquitecto y la arquitectura tradicionalmente han operado bajo la gravedad, que opera siempre en un mismo sentido y han manipulado la materia para definir los límites del espacio, permitiendo ordenar actividades desde un punto físico. La arquitectura ha sido vista como un proceso por el cual se define la organización de actividades en el espacio físico o virtual. Entonces lo que entendemos por edificio, es el resultado de este proceso que llamamos arquitectura. Los edificios sin construir físicamente también son resultado de este proceso llamado arquitectura. La construcción, es un escenario de procesos, con un fin. Construir significa agotar un proceso, porque el edificio se debe terminar. El edificio terminado es el resultado de estos procesos, de esta manera la arquitectura y la construcción, son hechos finitos.

Sin embargo la naturaleza y los sistemas más eficientes no son así. Porque sus formas son siempre inacabadas, permitiendo adaptarse al entorno cambiante. El estado ideal de un edificio entonces sería visto como una forma en constante construcción, soportando las actividades humanas. Definir recorridos, llenos y vacíos, espacios, lugares

de encuentro, la manera en como controlamos la información mediante códigos binarios o estrategias emergentes es una actividad propia del proceso de la arquitectura. Aunque los resultados, no se parezcan a las construcciones de las formas en el mundo real.

La arquitectura en un ambiente digital es un proceso abstracto que relaciona información con materia, en el espacio tiempo. En el mundo digital el tiempo también pertenece a la arquitectura. Porque el diseño al ser flexible es capaz de mutar y transformarse, y esto implica tiempo y espacio.

La nueva arquitectura debe relacionar lo físico y lo digital, donde los espacios empiezan a incluir de una manera más activa el tiempo y la autotransformación. De esta manera la arquitectura será creadora de procesos y no de hechos finitos.

Bibliografía

- Balmond, C. *Informal*. Prestel Verlag, 2002.
 Ceballos, J. *Enciclopedia del lenguaje C++*. Rama, 2003.
 Mandelbrot, B. *La geometría fractal de la naturaleza*. Barcelona: Tusquets, 1997.
 Segarra, J. *Vida artificial: Del caos al orden*. Alzira: Algar Editorial, 2002.
 Wolfram, S. *A new kind of science*. Wolfram Media, 2002.