

Uso de Herramientas Paramétricas de Optimización Evolutiva y Simulación Energética en el Diseño Basado en Performance

Using Evolutionary Optimization and Energy Simulation Tools in Performance-based Design

Sebastián Caldera

Universidad de Chile, Chile
caldera@ug.uchile.cl

Mauricio Loyola

Universidad de Chile, Chile
mloyola@uchile.cl

Gonzalo Silva

Universidad de Chile, Chile
gonzsilva@ug.uchile.cl

Abstract

This article describes three cases of integration of technologies of evolutionary optimization and energy simulation in performance-based design. The aim is to share the details of the process of creation, validation and use of the various techniques and tools, with an emphasis on the mistakes and successes obtained, so that experiences can be useful for non-specialist users interested in working with these methodologies.

Keywords: Diseño Paramétrico; Grasshopper 3D; Ecotect; GECO; Galápagos.

Introducción

Durante los últimos años el uso de herramientas de programación gráfica en arquitectura ha tenido un crecimiento exponencial cuyo resultado más evidente ha sido una proliferación global de proyectos con geometrías complejas. Una de las herramientas más difundidas, especialmente en el contexto académico latinoamericano, es Grasshopper 3D (GH), un plug-in para el software NURBS Rhinoceros 3D, gratuito y de rápido aprendizaje (Herrera, 2010; 2011). Dentro de las cualidades atractivas de GH que han contribuido a su rápida masificación, está la existencia de una gran cantidad de extensiones o add-ons desarrolladas por terceras partes que expanden las capacidades de la herramienta hacia funcionalidades complementarias específicas (Davis y Peters, 2013). Por ejemplo, existen extensiones para realizar form-finding usando principios físicos (Kangaroo), para realizar análisis estructural (Karamba), para vincular GH con la placa Arduino (Firefly), para hacer análisis topológico de mallas (Weaverbird) o para hacer simulación acústica (Pachyderm), entre otros. La mayoría de estas extensiones son gratuitas y vinculables entre sí, constituyendo un verdadero ecosistema de funcionalidades complejas y potencialmente enlazables que, en teoría, permiten el desarrollo de manera sencilla de algoritmos que si fueran realizados de forma tradicional resultarían técnicamente prohibitivos para la mayoría de los usuarios de estas herramientas.

Un tipo de extensiones que aparentan tener un potencial no suficientemente explotado son aquellas destinadas a realizar optimización evolutiva en problemas multi-objetivo. La optimización evolutiva es un conjunto de metodologías dentro del

campo de la inteligencia artificial que utilizan los principios de la evolución biológica para la búsqueda de soluciones óptimas a problemas complejos (Bentley, 1999; Rutten, 2010). Los problemas multi-objetivo, por su parte, son aquellos que tienen objetivos parciales que están en conflicto entre sí: la mejora de uno produce un empeoramiento en otro. La eficacia del uso de optimización evolutiva en problemas multi-objetivo está muy bien documentada en otras disciplinas desde los años 70 (ver, por ejemplo, a John Koza o Ingo Rechenberg). En arquitectura, una serie de investigaciones más recientes muestran que la optimización evolutiva es también adecuada y pertinente. Sin embargo, se trata de estudios liderados por especialistas que utilizan herramientas complejas, usualmente desarrolladas exclusivamente para esos fines, y por lo tanto, no fácilmente masificables entre usuarios no expertos. (Renner, 2003; Caldas, 2006).

En este contexto, este estudio intenta poner a prueba la facilidad y eficacia de la integración de extensiones para GH por medio de la realización de proyectos que consideren el uso intensivo de metodologías de optimización evolutiva en combinación con herramientas de análisis energético. El objetivo es compartir los detalles del proceso de creación, validación y uso de las distintas técnicas y herramientas, con un énfasis en los errores y aciertos obtenidos, a fin que las experiencias puedan ser útil para usuarios no especialistas interesados en trabajar con estas metodologías.

Metodología

El estudio consistió en el planteamiento de 3 problemas de diseño arquitectónico en que existiera la necesidad de realizar optimización energética multi-objetivo, y en su posterior resolución utilizando GH en combinación con extensiones de optimización evolutiva y simulación energética.

Para la optimización evolutiva se utilizó el componente *Galapagos Evolutionary Solver*, desarrollado por David Rutten. Este componente, hoy integrado a GH, permite realizar optimización evolutiva usando algoritmos genéticos de una manera sencilla, con pocas opciones de configuración que privilegian la facilidad de uso (Rutten, 2010). Para la simulación energética se utilizó la extensión *GECO*, desarrollada por Ursula Frick y Thomas Grabner. Esta es una extensión gratuita que permite vincular en tiempo real a GH con el software de diseño sustentable *Ecotect* de Autodesk, orientándose especialmente al diseño conceptual en las etapas iniciales de proyecto arquitectónico. (Grabner y Frick, 2013).

Adicionalmente, desde la versión 0.8.0051 en adelante, *Galapagos* incorpora una segunda metodología de optimización denominada *Simulated Annealing Solver (SAS)*, basada en los procesos termodinámicos que ocurren a nivel atómico durante el recocido de los metales (Rutten, 2011). Aunque este trabajo se enfoca en el uso del *Evolutionary Solver (ES)*, también se realizaron pruebas comparativas iniciales con el SAS, las que se comentan globalmente en algunos de los casos.

Los casos de estudio corresponden a 2 problemas de diseño que replican condiciones de encargos de proyectos reales: el diseño de la celosía de una fachada con fuerte radiación en un edificio de oficinas y el diseño de la volumetría general de una vivienda mínima para diferentes zonas geográficas de Chile. Previamente, se realizó un caso preliminar de exploración, con resultados previsible, para ensayar y validar el correcto funcionamiento de todo el *software* y extensiones involucrados.

En cada caso el proceso de desarrollo fue documentado paso a paso, cuantitativamente (tiempo de trabajo, intentos fallidos, tiempo de optimización, etc.) y cualitativamente (experiencias e impresiones personales de los investigadores).

Resultados

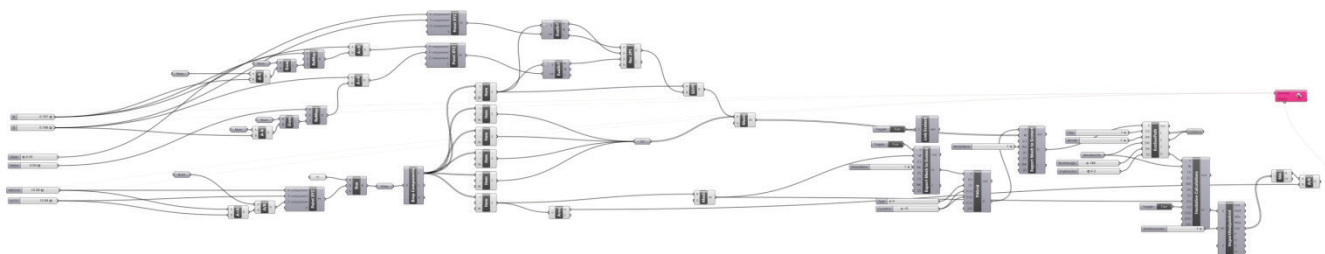


Figura 1: Algoritmo de vinculación entre GH y Ecotect, vía GECO, para el caso 1 preliminar.

Caso 1: Cuboide elemental

Como primer acercamiento al uso en conjunto de las herramientas *Galapagos*, *GECO* y *Ecotect* se planteó un problema de diseño elemental a fin de poder predecir un resultado óptimo y evidenciar fácilmente su correcto funcionamiento. El problema fue disminuir al mínimo posible la cantidad de radiación solar recibida por un recinto cuboidal de volumen constante y dimensiones variables que cuenta con un vano rectangular en una de sus caras. El alto y el largo de cuboide son variables independientes, mientras que la profundidad es dependiente del volumen. La posición del vano es una variable independiente, pero sus dimensiones son dependientes del tamaño del cuboide. De esta forma, las proporciones del cuboide y el tamaño del vano quedaron relacionados conflictivamente de manera no lineal, pero con una solución predecible.

El proceso de modelación presentó la primera dificultad de operación: *Rhinoceros*, es un programa NURBS, y por lo tanto, GH funciona generando geometría de esa forma. *Ecotect*, en cambio, necesita modelados construidos con mallas (*meshes*). El modelo paramétrico desarrollado en GH tuvo que ser convertido a mallas, manteniendo por separado a las mallas de la envolvente y la malla del piso, a fin de poder usar esta última con una grilla de análisis independiente en *Ecotect*.

El proceso de vinculación de GH con *Ecotect* vía *GECO* fue satisfactorio, con un algoritmo final que quedó constituido de la siguiente manera (Figura 1):

- 1º Vincular GH-Ecotect → *Eco link* (True para activar).
- 2º Exportar malla piso → Malla GH a *Export mesh*
- 3º Generar grilla de análisis → *Export mesh* a *Eco fit grid*
- 4º Exportar malla envolvente → *Fit grid* a un 2º *Export mesh*.
- 5º Fijar día y ángulo del sol → *Export mesh* a *Eco sun path*.
- 6º Activar cálculo solar en Ecotect → *Sun path* a *Eco sol cal*.
- 7º Acceder a grilla de análisis → *Sol cal* a *Eco grid request*
- 8º Visualizar colores → *Grid request* a *Eco mesh grid*

Para la optimización se usaron como variables las dimensiones de la cara principal del cuboide y las coordenadas de posición del vano. La función de aptitud corresponde a la sumatoria de la radiación incidente dividida en el área de la superficie horizontal del cuboide. El análisis se hizo usando una data climática arbitraria neutral.

Los resultados de optimización fueron satisfactorios, con una correcta integración y funcionamiento de todas las herramientas, y por lo tanto, cumpliendo con el propósito del ejercicio. Durante el proceso de construcción del algoritmo se realizaron análisis triviales (*dummy tests*) como método de verificación constante de la correcta operación del sistema. Los resultados óptimos se alcanzaron aproximadamente luego de 45 generaciones de soluciones, cada una de 50 individuos. La geometría final tendió a un volumen bajo y ancho, con una ventana pequeña en su parte superior, coincidiendo con la solución predicha. Un resumen de los resultados parciales a las generaciones 15, 30, 45 y 60 se muestran en la Tabla 1.

Tabla 1: Evolución de los resultados de optimización para el caso preliminar.

Generación	Menor radiación obtenida (Wh/m ²)	Dimensiones cuboide [m]	Área del vano [m ²]
15	58.47	12.89 x 9.81	0.466
30	18.12	13.10 x 11.95	0.406
45	18.06	13.15 x 11.94	0.403
60	17.91	13.22 x 11.98	0.389

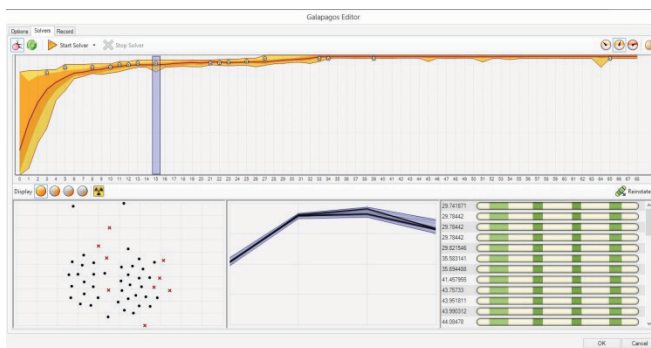


Figura 2: Galápagos Evolutionary Solver durante el proceso de optimización del caso 1.

Es importante indicar que la data energética obtenida es sólo referencial e indicativa de la correcta integración de las herramientas, y no debe ser interpretada en su fondo. Se trata de un modelo muy simplificado que no toma en consideración una importante cantidad de variables relevantes (ej. materialidad).

Caso 2: Fachada edificio de oficinas.

El segundo problema de estudio consistió en la optimización del diseño de una celosía de protección solar para disminuir al mínimo posible la radiación recibida por un grupo de oficinas con orientación poniente en un edificio de oficinas estándar. Sólo el diseño de la celosía se definió con variables independientes, quedando todo el resto de condiciones de diseño como parámetros fijos. El caso replica las condiciones y limitantes de un proyecto de arquitectura real encargado al Centro de Proyectos Externos de la Facultad de Arquitectura y Urbanismo de la Universidad de Chile.

Para realizar la parametrización de la celosía se asignaron variables de posición a los puntos que definen los planos de la geometría base y variables de ángulo para controlar su rotación sobre su propio eje. Estas pocas variables fueron suficientes para lograr un diseño complejo que impedía predecir una solución óptima.

El modelo de estudio se construyó respetando las lecciones aprendidas luego del ejercicio preliminar: mallas independientes para cada una de las superficies de análisis: piso, ventanas y muros. El modelo también incluyó el contexto inmediato del proyecto a fin de poder simular los sombreados pertinentes (Figura 3)

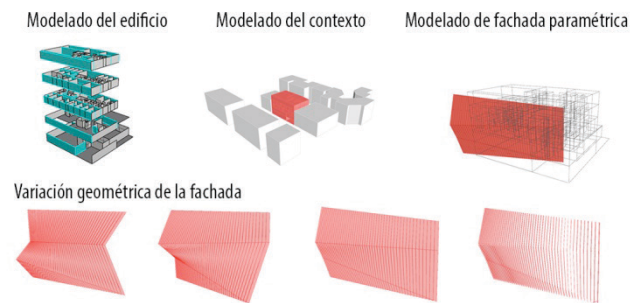


Figura 3: (Arriba) Modelo del edificio, del contexto y de la fachada parametrizada. (Abajo) Variación geométrica de la fachada.

El proceso de vinculación con *Ecotect* vía *GECO* se realizó de manera similar al caso preliminar, pero exportando sólo un piso como grilla de análisis, debido a que *Ecotect* sólo permite disponer de una grilla a la vez. Como objeto de estudio se definió un grupo de oficinas independientes en el 3er piso. El proceso de exportación no estuvo exento de dificultades, ya que algunas mallas del modelo no se enviaron correctamente a *Ecotect*. Presumiblemente, debido al gran volumen de información que debe ser compartida en tiempo real. Para subsanar este inconveniente, el modelo fue modificado de manera de exportar sólo aquella geometría que tuviera incidencia en el análisis.

Para la optimización se usaron como variables la posición e inclinación de la celosía y como función de aptitud la suma agregada de la ganancia solar total. El análisis se hizo usando la data climática correspondiente a la ubicación del proyecto (Comuna de Providencia, Santiago), durante el día más caluroso del año. Se hicieron 8 sesiones de 20 minutos cada una, documentando el mejor resultado obtenido en cada análisis.

El proceso de optimización tuvo un desarrollo sustantivamente diferente al primer caso. En una primera fase, el optimizador detectaba un valor de ángulo definido y luego ensayaba infinitamente variaciones de posición sin parecer acercarse a ningún óptimo. La razón era una mala definición del problema y su traducción al algoritmo geométrico, ya que la variable de ángulo de la celosía permitía cerrar completamente la fachada al punto de impedir el ingreso de radiación. Evidentemente, en este contexto la optimización se hacía redundante ya que con una

fachada totalmente cerrada, la posición de la celosía no tenía efecto alguno el resultado final. El problema fue solucionado restringiendo el ángulo de rotación de la celosía para impedir su cierre total.

En una segunda fase, el optimizador, luego de un tiempo de análisis y progresiva aproximación hacia una aparente solución óptima n1, repentinamente comenzaba a desviarse hacia otra nueva aparente solución óptima, y así, varias veces durante cientos de generaciones. Este comportamiento, al parecer errático, se debía a que el caso de diseño planteado tenía múltiples óptimos locales. Esto no es un problema o error en sí, sino una característica propia de la definición del problema (y bastante común en problemas multi-objetivo). El EV operó detectando los nuevos óptimos locales cuando un resultado espontáneamente generado en una generación posterior caía dentro de una cuenca de atracción diferente a la anterior. Sin embargo, luego de 4 análisis con similares tiempos de cálculo fue posible descubrir un grupo de soluciones óptimas locales, que con valores radicalmente distintos en sus variables independientes, alcanzaban un valor óptimo similar. En la práctica, estas soluciones representan opciones de diseño que comparten un nivel de respuesta equivalente frente a los requerimientos planteados (óptimos de Pareto), dejando al diseñador la posibilidad de escoger el más apropiado según otros criterios no inicialmente contemplados.

La Figura 4 muestra el momento del proceso de optimización cuando una nueva cuenca es descubierta y se comienza una fase de generación de resultados heredados para descubrir el óptimo local.

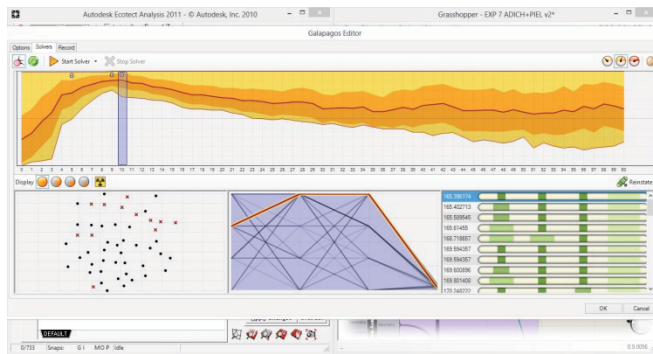


Figura 4: Galapagos Evolutionary Solver durante el proceso de optimización del caso 2.

En este caso, el uso de SAS tendió a simplificar el problema aislando las variables menos incidentes (posición) y concentrando la búsqueda sólo en la variable más influyente (ángulo). El valor óptimo obtenido por el SAS fue menos eficiente que el valor obtenido por el EV.

Caso 3: Volumetría de vivienda básica

El tercer caso de estudio consistió en la definición de un problema de superior complejidad que involucrara mayor cantidad de

variables, como una forma de explorar con más profundidad el fenómeno de múltiples óptimos equivalentes. El problema consistió en la determinación de la volumetría y fenestraciones óptimas de una unidad de vivienda básica teórica a fin de minimizar o maximizar su radiación solar en un clima cualquiera.

La parametrización contempló las siguientes variables independientes:

- Área del rectángulo base
- Largo de lado A del rectángulo
- Altura, ancho y posición de 5 vanos en muros
- Angulo de rotación de la vivienda con respecto al Norte.
- Posición vano puerta sobre pared
- Triangulación techo por medio de puntos con variación de altura en cada vértice.

La función de aptitud se definió como la suma agregada de la media aritmética de la radiación solar incidente en el plano inferior de la vivienda, a la que se le definió un óptimo mínimo, máximo o equivalente a un cierto valor dado (radiación ideal dada).

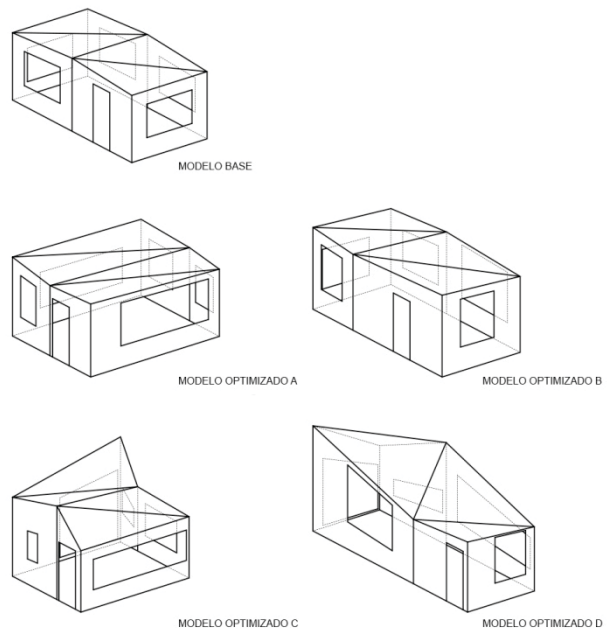


Figura 5: Variaciones del modelo para el caso de estudio 3.

En este caso, el comportamiento del optimizador fue similar al caso 2. Luego de varios ciclos de optimización se llegó a un conjunto de soluciones muy diferentes en su geometría pero equivalentes en su resultado valorizado final. En este caso, sin embargo, fue imposible visualizar un conjunto de soluciones similares. La cantidad de variables independientes (más de 13) aparece como excesiva para un problema con una función de aptitud amplia. Existe un grupo muy grande de posibles soluciones que tienen valores de radiación cercanos al ideal, lo que impide agrupar las soluciones en un grupo manejable de opciones de diseño.

Un aspecto importante fueron las diferencias producidas entre el uso de EV y SAS. Las soluciones generadas por el ES tendieron a simplificar la geometría de la envolvente, para concentrarse en la disposición y tamaño de los vanos. Por otra parte, el SAS obtuvo resultados más complejos y diversos, modificando todas las variables sin concentrarse en una variable por sobre otra.

Conclusiones

El objetivo inicial de poner a prueba la facilidad y eficacia de la integración de herramientas de optimización evolutiva y simulación energética en GH por usuarios no expertos fue cumplido satisfactoriamente.

En general, el proceso de modelación e intercambio de información fue fluido y sin mayores dificultades. Existieron algunos pocos inconvenientes, fundamentalmente relacionados con problemas de pérdida de información en la exportación de geometría desde GH a *Ecotect*, que fueron resueltos con cambios en el modelo geométrico. Los algoritmos corrieron sin problemas en equipos estándar (Intel i7 2.45 Ghz, 8Gb RAM), permitiendo un uso masivo de las herramientas sin limitantes de *hardware*.

Las observaciones respecto al proceso de aprendizaje de uso integrado de *GECO* y *Galápagos* pueden dividirse en dos etapas. La primera fase, destinada a conocer las herramientas y su modo de utilización, fue breve y sencilla debido a la abundancia de ejemplos disponibles online en sitios personales de entusiastas con la tecnología. La segunda fase, destinada a aplicar la metodología a casos más sofisticados, fue más lenta y complicada debido a la complejidad implícita que existe en la tarea de definir un problema de optimización multi-objetivo. Para un usuario no experto, esta tarea representa la verdadera dificultad del uso de estas tecnologías. Los casos aquí presentados muestran los errores cometidos en el planteamiento de la función de aptitud y de las variables de control, los cuales alteran significativa y peligrosamente los resultados finales. Esto representa un riesgo que puede llevar a un usuario no experto y atraído por la tecnología a deslumbrarse cándidamente por un resultado aparentemente “óptimo” en un contexto de un problema mal planteado desde su origen.

Los resultados de optimización particulares obtenidos muestran que en problemas planteados de forma ambigua o donde la cantidad de variables es muy alta, puede generarse una cantidad elevada de soluciones de similar nivel de optimización. La precisión en la definición de los criterios de selección y su

instrumentalización en la función de aptitud es central. Cuando se generan varios óptimos más o menos equivalentes, estas metodologías actúan sólo como una herramienta informativa al diseñador bajo una lógica de trabajo GFCL (*Generate First – Choose Later*). El diseñador queda con la libertad de escoger la opción más adecuada dentro de la gama de óptimos generados recurriendo a criterios adicionales a los contemplados inicialmente, o si se prefiriera, modificando la función de aptitud para incluir nuevas condicionantes del problema.

Agradecimientos

Agradecemos a la Prof. MSc. Alejandra Cortes, del área de sustentabilidad del Departamento de Arquitectura de la Universidad de Chile por su significativa ayuda durante el desarrollo de este estudio.

Referencias

- Caldas, L. (2006). GENE_ARCH: an evolution-based generative design system for sustainable architecture. In *Intelligent Computing in Engineering and Architecture* (pp. 109-118). Springer Berlin Heidelberg.
- Davis, D. y Peters, B. (2013) *Design Ecosystems: Customising the Architectural Design Environment with Software Plug-ins*. *Architectural Design*, 83 (2), pp 124-131.
- Herrera, P. (2010). Tecnologías disruptivas: programación y fabricación en Latinoamérica. XIV Congreso de la Sociedad de Gráfica Digital (SiGraDi), Bogotá, Colombia, 17-19 de Noviembre, pp. 213-216.
- Herrera, P. (2011). Rhinoscripting y Grasshopper a través de sus instructores: un estudio de patrones y usos. XV Congreso de la Sociedad de Gráfica Digital (SiGraDi), Santa Fé, Argentina 16-18 de Noviembre, pp.
- Bentley, P. (Ed.). (1999). *Evolutionary design by computers*. Morgan Kaufmann.
- Fogel, L.J., Owens, A.J., y Walsh, M.J. (1966) *Artificial Intelligence through Simulated Evolution*. New York: John Wiley.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. U Michigan Press.
- Renner, G. (2003). Genetic algorithms in CAD. *Computer-Aided Design*, 35(8), 707-708.
- Rutten, D. (2010) *Evolutionary Principles applied to Problem Solving*. AAG10 conference, Vienna
- Rutten, D. (2011) *Simulated Annealing, a brief introduction*. En: <http://ieatbugsforbreakfast.wordpress.com/2011/10/14/simulated-annealing-a-brief-introduction/> [Acceso 01 Septiembre 2013].
- Grabner, T y Frick, U. (2013). *GECO: Architectural design through environmental feedback*. *Architectural Design*, 83 (2), pp 142-143.