Algoritmos Genéticos

Angel Kuri

Centro de Investigación en Computación

Mayo de 2000

Algoritmos Genéticos[1]

LA NATURALEZA COMO OPTIMIZADORA

"AUNQUE EL INGENIO HUMANO PUEDE LOGRAR INFINIDAD DE INVENTOS, NUNCA IDEARÁ NINGUNO MEJOR, MÁS SENCILLO Y DIRECTO QUE LOS QUE HACE LA NATURALEZA, YA QUE EN SUS INVENTOS NO FALTA NADA Y NADA ES SUPERFLUO", ESCRIBIÓ LEONARDO DA VINCI (*CUADERNO DE NOTAS, LA VIDA Y ESTRUCTURA DE LAS COSAS, EL EMBRIÓN*). UN SIGLO DESPUÉS, JOHANNES GOLDSCHMIDTH, MEJOR CONOCIDO POR SU NOMBRE LATINO *FABRICIUS AB AQUAPENDENTE* (QUIEN SE DISPUTARA CON GALILEO EL DESCUBRIMIENTO DE LOS DEFECTOS SOLARES), AFIRMARÍA, ANTICIPÁNDOSE A DARWIN: "LA NATURALEZA PERPETÚA AQUELLO QUE RESULTA MEJOR." (*DE MOTU LOCALI ANIMALIUM*). TODOS LOS SERES VIVOS QUE HABITAMOS EN ESTE PLANETA SOMOS, DE ALGUNA MANERA, OBRAS CASI PERFECTAS. CUANDO NOS PERCATAMOS DE LA VASTA COMPLEJIDAD QUE SOMOS, DE CÓMO NUESTRO CUERPO ES UNA ENREDADA MADEJA DE RELACIONES DELICADAMENTE ESTABLECIDAS ENTRE LOS DISTINTOS ÓRGANOS, NO PODEMOS MÁS QUE MARAVILLARNOS Y OTRO TANTO OCURRE CUANDO PERCIBIMOS AL RESTO DE LOS SERES VIVOS DE LA MISMA MANERA. RECIENTEMEN-TE, UN GRUPO DE INGENIEROS QUE DISEÑABA UN SUBMARINO SE PERCATÓ DE QUE EL PINGÜINO POSEE UNA FORMA TAL, QUE AL FLUIR EL AGUA A SU ALREDEDOR, PRÁCTICAMENTE NO GENERA TURBULENCIAS EN LAS QUE SE PIERDA LA ENERGÍA INVERTIDA POR EL AVE AL NADAR Y AL MISMO TIEMPO, POSEE UN VOLUMEN CERCANO AL MÁXIMO, LO QUE PERMITE AL PINGÜINO ACUMULAR GRAN CANTIDAD DE GRASA QUE LO AÍSLA DEL FRÍO EXTERIOR Y CONSTITUYE SU RESERVA DE ENERGÍA. SE PODRÍAN CITAR MILES DE EJEMPLOS MÁS: "[...] CUANTO MÁS INFORMADOS E ILUMINADOS ESTEMOS ACERCA DE LAS OBRAS DE DIOS, MÁS INCLINADOS ESTAREMOS A ENCONTRARLAS EXCELENTES Y TOTALMENTE CONFORMES A CUANTO SE HUBIERA PODIDO DESEAR", DIRÍA LEIBNIZ (*DISCOURS DE METAPHYSIQUE*).

La naturaleza genera seres óptimos, seres perfectamente adaptados a su entorno, constituido por una infinidad de circunstancias: temperatura, presión atmosférica, precipitación, velocidad del viento, altitud, nivel de insolación, depredadores, alimentos, etc. En función de su entorno el pingüino es un animal perfecto para vivir en las soledades antárticas; en el Amazonas perecería. Su adaptación es perfecta porque, a lo largo de miles de generaciones se ha perfeccionado a pequeños saltos infinitesimales. Lo que vemos hoy en día es el resultado acumulado de miles de experimentos exitosos que han ido refinando paulatinamente alguna creación primigenia. Los experimentos fallidos, probablemente algunos órdenes de magnitud más numerosos que los exitosos, no los vemos. Los individuos resultantes perecieron compitiendo al lado de otros más aptos para sobrevivir. "La selección natural obra solamente mediante la conservación y acumulación de pequeñas modificaciones heredadas, provechosas todas al ser conservado.", escribió Darwin y dio nombre a este proceso (*El origen de las especies, selección natural*).

Estas "modificaciones heredadas", señaladas por Darwin como las generadoras de organismos mejores, son llamadas mutaciones hoy en día y constituyen el motor de la evolución. Un organismo mutante ha sufrido una modificación que lo hace diferente al resto de sus congéneres. Esta modificación puede ser un in-conveniente para él (la falta de algún miembro útil de su cuerpo, por ejemplo), pero puede ocurrir también que le confiera alguna cualidad que le permita sobre-vivir más fácilmente que al resto de los individuos de su especie. Este organismo tendrá mayor probabilidad de reproducirse y heredará a sus descendientes la ca-racterística que le dio ventaja. Con el tiempo, gracias a la competencia, los organismos que en un principio eran raros se volverán comunes a costa de la desaparición del "modelo anterior". Se habrá dado entonces un paso en el proceso evolutivo.

Esta cualidad del proceso natural de la evolución (generar organismos óptimos sobre los que influyen infinidad de variables), llamó la atención de algunos científicos en las décadas de los cincuenta y sesenta. Un alemán, de apellido Rechenberg, introdujo en 1965 lo que denominó *evolutionsstrategie*, o estrategias evolutivas, como un método para optimizar funciones de varias

[1] Este texto es una adaptación de extractos del libro "Algoritmos Genéticos", de Angel Kuri y José Galaviz, a ser publicado por el Fondo de Cultura Económica.

variables que definían dispositivos tales como perfiles de alas de avión. En 1966 los investigadores Fogel, Owens y Walsh se dieron a la tarea de dejar evolucionar má-quinas de estados finitos sometiendo sus diagramas de transición a cambios alea-torios (mutaciones), creando con ello lo que se conoce como *programación evolutiva*. También en los sesenta, John Holland, junto con algunos de sus colegas y alumnos de la Universidad de Michigan, desarrolló lo que se conoce como *algoritmos genéticos* (AGs). Sin embargo, el objetivo de Holland no era inventar un método de optimización basado en los mecanismos de la evolución natural, sino elaborar un estudio formal acerca del fenómeno de adaptación en sistemas naturales y artificiales, es decir, definir un modelo para la adaptación. El algoritmo genético definido por Holland en su trabajo [7] es un intento de abstraer las características esenciales del proceso evolutivo tal como se observa en la naturaleza, con vistas a importarlo a sistemas de cómputo.

En la actualidad, los AGs son preferentemente utilizados como métodos de búsqueda de soluciones óptimas que simulan la evolución natural y han sido usados con éxito en la solución de problemas de optimización combinatoria, op-timización de funciones reales y como mecanismos de aprendizaje de máquina (*machine learning)*. Esto último les ha ganado un lugar en el campo de la inteligencia artificial.

UN POCO DE BIOLOGÍA

EN ESTA SECCIÓN SE PRESENTARÁ SUPERFICIALMENTE LO QUE SE SABE ACERCA DE LOS MECANISMOS DE LA HERENCIA Y LA TERMINOLOGÍA UTILIZADA EN ESTA ÁREA. EL OBJETIVO ES EVIDENCIAR LA ANALOGÍA ENTRE LOS ALGORITMOS GENÉTICOS Y AQUELLO QUE PRETENDEN SI-MULAR. EL LECTOR INTERESADO EN PROFUNDIZAR PUEDE CONSULTAR [13].

Cada individuo de cada una de las especies que habitan en nuestro planeta poseen ciertas características que lo identifican. Si hablamos de seres humanos, cada uno posee cierta estatura, cierto color de ojos y de cabello y cierto tipo san-guíneo, entre otras muchas. Estas características “externas”, aunque algunas de ellas no se puedan ver, como el tipo sanguíneo, constituyen lo que se denomina el *fenotipo* de un individuo. Cada una de estas características es igual a la co-rrespondiente de alguno de los antecesores del individuo, es decir, nos son dadas por herencia, o por lo menos nos es dada cierta predisposición a ella (como la diabetes, por ejemplo). El fenotipo es resultado de la interacción del medio am-biente en que se desarrolla un individuo y la herencia que éste recibe de sus an-cestros. La herencia impone ciertos límites o predisposiciones que, al sumarse con el medio, generan el fenotipo. A veces el medio no importa, por ejemplo, no puede intervenir en nuestro color de ojos, pero en otras influye de manera deter-minante. Si se posee cierta predisposición a padecer enfermedades cardiovasculares pero se tiene una excelente condición aeróbica desde pequeño, posiblemente éstas nunca se padezcan. El fenotipo de cada individuo está determinado por las proteínas que produce, y esto a su vez está definido en la información genética de cada una de sus células.

La información acerca de cuáles proteínas se producirán está contenida en los cromosomas del individuo. En cada célula somática (aquellas que constituyen el organismo) existen dos juegos de cromosomas que definen las mismas características; un juego es aportación del padre del individuo y el otro lo es de la madre. Un ser humano posee 23 pares de cromosomas.

En términos sencillos, un cromosoma es una larga molécula de ADN (Ácido Desoxirribonucleico), formada por cuatro distintos compuestos más simples lla-mados bases o nucleótidos: adenina (A), guanina (G), citosina (C) y timina (T). Cada subcadena de tres nucleótidos codifica un aminoácido diferente que al unirse con los generados por otros tercetos, formará una proteína. A las subcade-nas de tres nucleótidos se les llama *codones* y al conjunto de nucleótidos que codifican una proteína completa se les llama *genes*. El valor que posee un gene determinado se denomina *alelo.*

Las células que poseen dos juegos de cromosomas se denominan *diploides,* ya que en éstas los cromosomas se encuentran en pares; también los genes deben estar en pares homólogos. Si ambos tienen el mismo alelo se dice que son *homocigos,* si no, son *heterocigos,* y en este último caso sólo uno de los alelos se manifestará en el fenotipo (éste se denomina *dominante*, su homólogo que no se manifiesta, se llama *recesivo*). El conjunto de todos los cromosomas, es decir, toda la información

genética de un individuo se llama *genoma* y el conjunto de genes contenidos en el genoma *genotipo*. Es éste el que determina, en buena medida, el fenotipo del individuo.

Hay unas células especiales llamadas *gametos*, que intervienen en la reproducción (los espermatozoides y los óvulos humanos pertenecen a esta categoría). Los gametos no se reproducen por *mitosis* como el resto de las células, el proceso de división se llama en este caso *meiosis*. En la mitosis las células producidas son diploides, mientras que en la meiosis el resultado, los gametos, son *haploides*, sólo tienen un juego de cromosomas.

Partiendo de una sola célula diploide el proceso meiótico es como sigue (figura F.I.2.A):

1. Se duplica el número de cromosomas en la célula, esto es, se hace una copia de cada cromosoma. Al final quedan dos juegos correspondientes al padre y dos a la madre.
2. Se cruzan un juego de cromosomas del padre con uno de la madre, formándose dos juegos de cromosomas híbridos. El resultado es un juego de cromosomas puros del padre, un juego puro de la madre y dos juegos de cromosomas híbridos.
3. Se divide la célula dos veces y al final del proceso quedan cuatro células haploides: una con cromosomas puros del padre, una con cromosomas puros de la madre y dos con cromosomas híbridos.

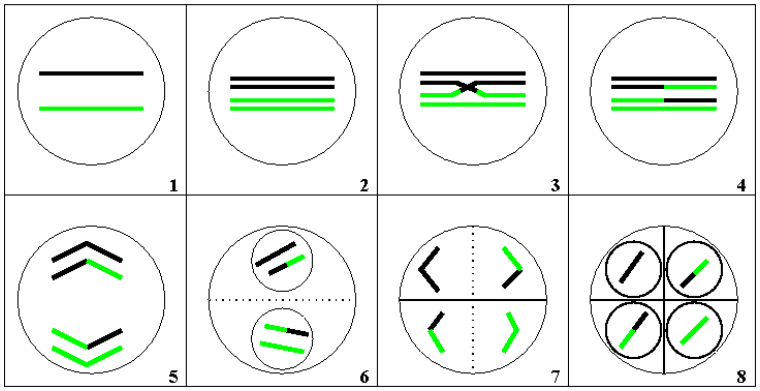


Figura F.I.2.A: *REPRODUCCIÓN CELULAR POR MEIOSIS: EN 1 APARECE LA CÉLULA DIPLOIDE ORI-GINAL, EN 2 LOS CROMOSOMAS SE REPLICAN, EN 3 SE FORMAN QUIASMAS, EN 4 YA HAY CUATRO JUEGOS DE CROMOSOMAS (DEL PADRE, DE LA MADRE Y DOS HÍBRIDOS), EN 5 SE SEPARAN EN PARES, EN 6 SE FORMA UNA PRIMERA MEMBRANA CELULAR, EN 7 SE SEPARAN NUEVAMENTE LOS CROMOSOMAS, FINALMENTE EN 8 SE FORMA UNA NUEVA MEMBRANA QUE SEPARA LOS NUEVOS NÚCLEOS (AHORA CADA UNO TIENE SÓLO UN JUEGO DE CROMOSOMAS), CADA NUEVA CÉLULA ES HAPLOIDE.*

En el paso dos del proceso de meiosis se mezclan las características del padre y la madre. Para el cruzamiento de dos cromosomas se forman entre ellos puntos de ruptura y unión de las cadenas de ADN. Estos puntos, llamados *quiasmas*, cortan el cromosoma en segmentos llamados *cromátidas* y unen cromátidas complementarias de dos distintos cromosomas. Al final cada cromosoma que participó en la cruza queda constituido por segmentos que ya poseía y por otros que eran de su análogo (figura F.I.2.B).

En el paso uno del proceso se deben replicar los cromosomas existentes. Hay una enzima encargada de copiarlos, ésta es la *ADN-polimerasa*. La molécula de ADN tiene forma de una doble hélice, como una escalera de caracol. La enzima abre por en medio los “escalones” de esta hélice y ensambla en cada mitad los nucleótidos que debe ensamblar (figura F.I.2.C). Ocasionalmente esta enzima comete un error, que puede ser causado por radiaciones energéticas externas o sustancias extrañas. La alteración de la molécula de ADN original constituye una mutación que puede manifestarse en el fenotipo y hacer al individuo diferen-te del resto de sus congéneres. Es muy poco probable que cambiar al azar un tro-zo de información que la naturaleza ha refinado cuidadosamente a lo largo de millones de años resulte en algo bueno. Por lo general las mutaciones son desfavorables, incluso letales, para el organismo

mutante. Pero ocasionalmente pueden no serlo y conferirle a dicho organismo alguna ventaja que le permita sobrevivir más fácilmente en su medio. Esta característica será transmitida a sus descendientes y un pequeño paso evolutivo se habrá dado.

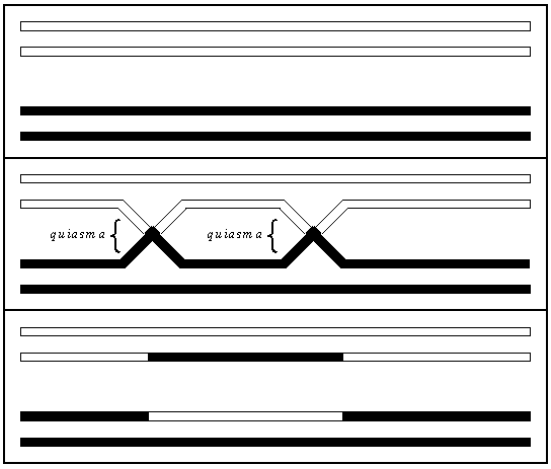


Figura F.I.2.B: *EL PROCESO DE CRUZAMIENTO DE DOS CROMOSOMAS. AL PRINCIPIO EXISTEN DOS PARES DE CROMOSOMAS, UN PAR DEL PADRE Y OTRO DE LA MADRE; ENTRE ELLOS SE FORMAN PUNTOS DE RUPTURA Y UNIÓN (QUIASMAS) QUE DEFINEN SEGMENTOS LLAMADOS CROMÁTIDAS. SE UNEN CROMÁTIDAS COMPLEMENTARIOS PARA FORMAR DOS CROMOSOMAS HÍBRIDOS.*

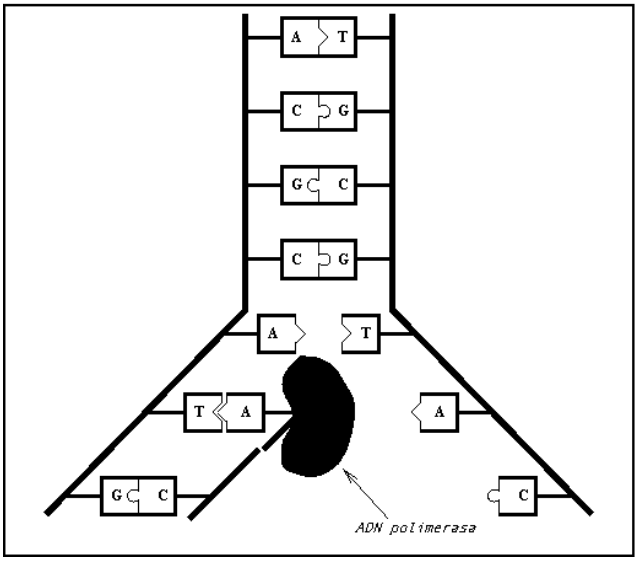


Figura F.I.2.C: *PROCESO DE REPLICACIÓN DE LA MOLÉCULA DE ADN. LA MOLÉCULA SE ABRE CO-MO UNA CREMALLERA. UNA ENZIMA (ADN-POLIMERASA) SE ENCARGA DE COLOCAR EN CADA FILA LOS NUCLEÓTIDOS QUE CORRESPONDEN. DE CADA FILA SE OBTIENE UNA NUEVA MOLÉCULA DE ADN.*

ALGORITMOS GENÉTICOS

SE MENCIONÓ YA QUE LOS ALGORITMOS GENÉTICOS SIMULAN EL PROCESO DE EVOLUCIÓN NATURAL. EN ESTA SECCIÓN SE PROCURARÁ ACLARAR LA MANERA COMO SE LLEVA A CABO ESTA SIMULACIÓN.

Codificación del dominio

EN LA NATURALEZA LAS CARACTERÍSTICAS DE LOS SERES VIVOS, INCLUSO AQUÉLLAS QUE LOS HACEN ÓPTIMOS PARA HABITAR EN SU MEDIO, ESTÁN DETERMINADAS POR LAS PROTEÍNAS QUE PRODUCEN. A SU VEZ, COMO HEMOS VISTO, ESTAS PROTEÍNAS (O MÁS BIEN, LOS AMINOÁCIDOS QUE LAS FORMAN) SE

CODIFICAN EN EL MATERIAL GENÉTICO CONTENIDO EN CADA UNA DE LAS CÉLULAS DEL INDIVIDUO. ASÍ PUES, LA NATURALEZA HA MAPEADO CADA POSIBLE SOLUCIÓN AL PROBLEMA DE CREAR UN INDIVIDUO ÓPTIMO EN UNA SECUENCIA PARTICULAR DE BASES QUE PRODUCIRÁ CIERTAS PROTEÍNAS, HA *CODIFICADO* EL DOMINIO DEL PROBLEMA (TODOS LOS POSIBLES INDIVIDUOS) MAPEÁNDOLO AL CONJUNTO DE TODAS LAS POSIBLES SE-CUENCIAS DE NUCLEÓTIDOS.

Así, para un algoritmo genético lo primero que se requiere es determinar en qué espacio se encuentran las posibles soluciones al problema que se pretende resolver. En caso de tener un problema de optimización de una función cuyo do-minio es un subconjunto de los números reales, entonces este subconjunto es al que nos referimos. Pero el algoritmo opera sobre "códigos genéticos", sobre ge-notipos que se deberán mapear al espacio de soluciones. Es decir, es necesario *codificar* de alguna manera el dominio del problema para obtener estructuras manejables que puedan ser manipuladas por el AG. Cada una de estas estructuras constituye el equivalente al genotipo de un individuo en términos biológicos. El elemento del dominio del problema al que se mapea este genotipo es el análo-go al fenotipo. Es frecuente que el código de los elementos del dominio del pro-blema utilice un alfabeto binario (0's y 1's).

Una vez que se ha definido la manera de codificar los elementos del dominio del problema y se conoce la forma de pasar de un elemento a su código y viceversa, es necesario fijar un punto de partida. Los algoritmos genéticos manipulan conjuntos de códigos en generaciones sucesivas. Nuevamente haciendo una ana-logía, manipulan poblaciones de códigos. En éstas un código puede aparecer más de una vez. El algoritmo se encargará de favorecer la aparición en la población de códigos que correspondan a elementos del dominio que estén próximos a resolver el problema. En resumen, el algoritmo recibirá como entrada una población de códigos y a partir de ésta generará nuevas poblaciones, donde algunos códigos desaparecerán mientras que otros, que se mapean en mejores soluciones posibles, aparecen con más frecuencia hasta que se encuentra una satisfactoria o hasta que se cumple alguna otra condición de terminación. A lo largo del texto, los códigos en una población, es decir, los elementos de ésta serán llamados *individuos* y a los códigos en general, ya no en el contexto exclusivo de una población, se les denominará indistintamente *cromosomas, genotipo, genoma* o *código genético*, por analogía con los términos biológicos de donde surgen.

Evaluación de la población

EN LA NATURALEZA HAY INDIVIDUOS MÁS HÁBILES QUE OTROS PARA SOBREVIVIR. EN UNA MANADA DE GACELAS HAY UNAS MÁS RÁPIDAS QUE OTRAS, HAY ALGUNAS ENFERMAS O PROPENSAS A ENFERMAR, HAY ALGUNAS MÁS DÉBILES QUE OTRAS. TODAS LAS CARACTERÍSTICAS MENCIONADAS SEÑALAN ALGUNA DIFERENCIA ENTRE LOS INDIVIDUOS. ADEMÁS, ESTA DIFERENCIA ES *RELATIVA*, ES DECIR, SIEMPRE ESTÁ REFERIDA AL RESTO DE LA POBLACIÓN DE GACELAS. SE DICE: "ÉSTA ES MÁS RÁPIDA QUE EL RESTO DE LA POBLACIÓN" O "AQUÉLLA ES MÁS SALUDABLE QUE EL PROMEDIO DE SUS CONGÉNERES". DE ALGUNA MANERA, SIEMPRE SE RELACIONA AL INDIVIDUO CON LA POBLACIÓN A LA QUE PERTENECE. SI SE CONSIDERA CADA UNA DE ESTAS CARACTERÍSTICAS COMO MEDIDAS DEL DESEMPEÑO DE CADA INDIVIDUO, SE ESTÁ HABLANDO DE QUE EL DESEMPEÑO DE CADA INDIVIDUO DE LA POBLACIÓN ESTÁ EN FUNCIÓN DEL DESEMPEÑO DE SUS CONGÉNERES. POR EJEMPLO, ¿QUÉ TAN VELOZ DEBE SER UNA GACELA PARA EVITAR SER CAZADA POR UN CHEETAH? SI ES MÁS RÁPIDA QUE EL CHEETAH ESTÁ SALVADA. PERO LOGRAR ESO ES DIFÍCIL, CASI NO HABRÍA GACELAS. MÁS BIEN LA RESPUESTA CORRECTA ES RELACIONAR AL INDIVIDUO CON EL RESTO DE SU POBLACIÓN Y DECIR: DEBE SER MÁS RÁPIDA QUE LA GACELA MÁS LENTA, DE ESTE MODO EL DEPREDADOR PERSEGUIRÁ A OTRA (CLARO QUE DESPUÉS DE LA PRIMERA CACERÍA LA MEDIDA DE DESEMPE-ÑO CAMBIA PORQUE LO HA HECHO LA POBLACIÓN MISMA). AL IGUAL QUE EN LA NATURALEZA, EN LOS ALGORITMOS GENÉTICOS ES NECESARIO ESTABLECER ALGÚN CRITERIO QUE PERMITA DECIDIR CUÁLES DE LAS SOLUCIONES PROPUESTAS EN UNA POBLACIÓN SON MEJORES RESPEC-TO DEL RESTO DE LAS PROPUESTAS Y CUÁLES NO LO SON. ES NECESARIO ESTABLECER, PARA CADA INDIVIDUO, UNA MEDIDA DE DESEMPEÑO RELATIVA A LA POBLACIÓN A LA QUE PERTENECE.

Para determinar cuáles de estos individuos corresponden a buenas propuestas de solución y cuáles no, es necesario calificarlos de alguna manera. Cada individuo de cada generación de un algoritmo genético recibe una *calificación* o, para usar el término biológico, una medida de su *grado de adaptación* (*fitness*). Éste es un número real no negativo tanto más grande cuanto mejor sea la solución propuesta por dicho individuo. El objetivo de este número es que permita distinguir propuestas de solución buenas de aquéllas que no lo son. Si el problema a resolver consiste en maximizar una función, entonces la calificación asignada a un individuo determinado debe indicar qué tan alto es el valor de la función en el elemento de su dominio codificado por el individuo. Si, en cambio, el problema es determinar la ruta más corta entre dos puntos, la calificación deberá ser tanto más alta cuanto más corto sea el camino codificado en el individuo que esté siendo calificado.

Evidentemente, al hablar de que a cada individuo de la población se le asigna una y sólo una calificación, se está hablando de una función que se denomina *función de adaptación*, cuya evaluación puede no ser sencilla y es, de hecho, lo que en la mayoría de los casos consume más tiempo en la ejecución de un algoritmo genético. Hay que tener en cuenta que se evalúa una vez en cada individuo de cada generación. Si un AG es ejecutado con una población de tamaño 100 du-rante 100 generaciones, la función es evaluada 10,000 veces. Además, puede darse el caso de que la función de evaluación no tenga una regla de correspondencia explícita, esto es, una expresión algebraica, y puede ocurrir incluso que la función cambie de generación en generación.

Selección

UNA VEZ CALIFICADOS TODOS LOS INDIVIDUOS DE UNA GENERACIÓN, EL ALGORITMO DEBE, AL IGUAL QUE LO HACEN LA NATURALEZA Y EL HOMBRE, SELECCIONAR A LOS INDIVIDUOS MÁS CALIFICADOS, MEJOR ADAPTADOS AL MEDIO, PARA QUE TENGAN MAYOR OPORTUNIDAD DE RE-PRODUCCIÓN. DE ESTA FORMA SE INCREMENTA LA PROBABILIDAD DE TENER INDIVIDUOS "BUENOS" (CON ALTA CALIFICACIÓN) EN EL FUTURO. SI DE UNA DETERMINADA GENERACIÓN DE INDIVIDUOS SE SELECCIONARAN SÓLO AQUELLOS CON UNA CALIFICACIÓN MAYOR O IGUAL QUE CIERTO NÚMERO *C* PARA PASARLOS A LA SIGUIENTE GENERACIÓN, ES CLARO QUE EN ÉSTA LA CALIFICACIÓN PROMEDIO SUPERARÁ *C* Y POR TANTO AL PROMEDIO DE LA GENERACIÓN AN-TERIOR. LA SELECCIÓN OCASIONA QUE HAYA MÁS INDIVIDUOS BUENOS, EXPLOTA EL CONOCI-MIENTO QUE SE HA OBTENIDO HASTA EL MOMENTO, PROCURANDO ELEGIR LO MEJOR QUE SE HAYA ENCONTRADO, ELEVANDO ASÍ EL NIVEL DE ADAPTACIÓN DE TODA LA POBLACIÓN. MÁS ADELANTE EN EL TEXTO SE VERÁ QUÉ TAN IMPORTANTE ES ESTA EXPLOTACIÓN.

En principio podría parecer que es conveniente tener una estrategia de selección estricta para que mejore rápidamente la población y converja el algoritmo, es decir, que la población se acumule alrededor de un genotipo óptimo. Esto no es cierto. Lo que ocurrirá es que la población se acumulará rápidamente alrededor de algún individuo que sea bueno, comparativamente con el resto de los individuos considerados a lo largo de la ejecución del algoritmo, pero este individuo puede no ser el mejor posible. A esto se le suele llamar *convergencia pre-matura*. No se puede asegurar pero sí procurar que lo anterior no ocurra. Además de la explotación es necesario que exista exploración. El AG debe, no sólo seleccionar de entre lo mejor que ha encontrado, sino procurar encontrar mejores individuos. A esto se dedican los operadores que serán descritos a continuación, los que aseguran que en todo momento exista cierto grado de variedad en la población, procurando con ello que no se "vicie".

En la estrategia de selección normalmente se incluye un elemento extra que sirve de "ancla". Si sólo se hace selección forzando que sea más probable elegir al mejor individuo de la población pero sin asegurarlo, es posible que este individuo se pierda y no forme parte de la siguiente generación. Para evitar lo anterior se fuerza la selección de los mejores *n* individuos de la generación para pasar intactos a la siguiente. A esta estrategia se le denomina *elitismo* y puede ser generalizada especificando que permanezcan en la población los *n* mejores individuos de las pasadas *k* generaciones.

Cruzamiento

DURANTE LA MEIOSIS OCURRE EL PROCESO DE PRODUCCIÓN DE GAMETOS. EL CÓDIGO GENÉTICO DE LOS PADRES DE UN INDIVIDUO SE MEZCLA PARA PRODUCIR GAMETOS CUYO CONTENIDO GENÉTICO ES HÍBRIDO, ES DECIR, UNA MEZCLA. DE ESTA MANERA ES POSIBLE QUE UN INDIVIDUO HEREDE A SUS DESCENDIENTES LAS CARACTERÍSTICAS MEZCLADAS DE SUS PROPIOS PADRES, POR EJEMPLO: EL COLOR DE OJOS DEL PADRE Y EL DE CABELLO DE LA MADRE O, PA-RA APROVECHAR EL EJEMPLO MENCIONADO ANTES, ES POSIBLE QUE UNA GACELA HEREDE LA VELOCIDAD DE SU ABUELO PATERNO Y LA FUERZA DE SU ABUELA PATERNA, LA SALUD DE SU ABUELO MATERNO Y LA AGUDEZA VISUAL DE SU ABUELA MATERNA. SI ESTAS CARACTERÍSTICAS LE CONFIRIERON A SUS ANCESTROS UNA ALTA APTITUD DE SOBREVIVENCIA, ENTONCES ESTE IN-DIVIDUO SERÁ, CON ALTA PROBABILIDAD, UN INDIVIDUO EXITOSO EN SU MANADA. LA CRUZA DE LOS CÓDIGOS GENÉTICOS DE INDIVIDUOS EXITOSOS FAVORECE LA APARICIÓN DE NUEVOS INDIVIDUOS QUE HEREDEN DE SUS ANCESTROS CARACTERÍSTICAS DESEABLES.

En el contexto de los algoritmos genéticos reproducirse significa que, dados dos individuos seleccionados en función de su grado de adaptación, éstos pasen a formar parte de la siguiente generación o, al menos, mezclen sus códigos gené-ticos para generar *hijos* que posean un código híbrido. Es decir, los códigos genéticos de los individuos se *cruzan*. Existen muchos mecanismos de cruzamiento. En esta sección sólo se presenta uno de ellos pero todos tienen por objeto que el código de un individuo *A* y el de uno *B*, previamente seleccionados, se mezclen, es decir, se fragmenten y recombinen para formar nuevos individuos con la esperanza de que éstos hereden de sus progenitores las características deseables. El mecanismo de cruzamiento más común es el llamado *cruzamiento de un punto*, el cual se describe con detalle más adelante. Por ahora considérese como una analogía directa del proceso de cruzamiento descrito en la sección anterior, a propósito de la generación de gametos, con la formación de un único quiasma entre los cromosomas (individuos). A este quiasma es al que se le denominará más adelante *punto de corte*.

Mutación

ALGUNAS VECES, MUY POCAS DE HECHO, LA ADN-POLIMERASA (LA ENZIMA ENCARGADA DE REPLICAR EL CÓDIGO GENÉTICO), SE EQUIVOCA Y PRODUCE UNA MUTACIÓN, UNA ALTERACIÓN ACCIDENTAL EN EL CÓDIGO GENÉTICO DE LOS SERES VIVOS.

Ocasionalmente algunos elementos del código de ciertos individuos de un algoritmo genético se alteran a propósito. Éstos se seleccionan aleatoriamente en lo que constituye el símil de una *mutación*. El objetivo es generar nuevos individuos, que exploren regiones del dominio del problema que probablemente no se han visitado aún. Esta exploración no presupone conocimiento alguno, no es ses-gada. Aleatoriamente se buscan nuevas soluciones posibles que quizá superen las encontradas hasta el momento. Esta es una de las características que hacen aplicables los algoritmos genéticos a gran variedad de problemas: no presuponer conocimiento previo acerca del problema a resolver ni de su dominio, no sólo en la mutación sino en el proceso total. De hecho, el problema a resolver sólo deter-mina la función de evaluación y la manera de codificar las soluciones posibles (la semántica de los códigos genéticos de los individuos). El resto de los subprocesos que constituyen el algoritmo son independientes y universalmente aplicables.

El algoritmo genético simple (AGS)

EN SU TRABAJO [7], HOLLAND PROPONE UNA MANERA DE SELECCIONAR INDIVIDUOS Y DE CRUZARLOS. ACTUALMENTE EXISTEN MUCHAS OTRAS PROPUESTAS, PERO LAS DE HOLLAND CONSTITUYEN AÚN HOY LA BASE DE MUCHOS DESARROLLOS TEÓRICOS Y PRÁCTICOS EN EL ÁREA. GOLDBERG [6] RETOMÓ EL ALGORITMO DE HOLLAND Y LO POPULARIZÓ LLAMÁNDOLO *ALGORITMO GENÉTICO SIMPLE* (AGS). EN ÉSTE SE CONSIDERA QUE LOS CÓDIGOS GENÉTICOS ESTÁN EN BINARIO. EXPLICADO CON DETALLE, EL PROCESO DE UN AGS ES:

1. Decidir cómo codificar el dominio del problema.

2. Generar un conjunto aleatorio (población inicial) de *N* posibles soluciones codificadas al problema. A ésta se le llamará la población actual.
3. Calificar cada posible solución (individuo) de la población actual.
4. Seleccionar dos individuos de la población actual con una probabilidad proporcional a su calificación.
5. Lanzar una moneda al aire (con probabilidad p_c cae cara).
6. Si cayó cara mezclar los códigos de los dos individuos seleccionados para formar dos híbridos, a los que llamaremos *nuevos individuos*.
7. Si cayó cruz llamamos a los individuos seleccionados *nuevos individuos*.
8. Por cada bit de cada nuevo individuo lanzar otra moneda al aire (con probabilidad p_m cae cara).
9. Si cae cara cambiar el bit en turno por su complemento.
10. Si cae cruz el bit permanece inalterado.
11. Incluir a los dos nuevos individuos en una *nueva población*.
12. Si la nueva población tiene ya *N* individuos, llamarla *población actual* y regresar al paso 3, a menos que se cumpla alguna condición de terminación.
13. Si no, regresar al paso 4.

En el algoritmo se utiliza el término “lanzar una moneda al aire” para decir un experimento de Bernoulli (aquel en el que pueden ocurrir exclusivamente dos eventos posibles, uno con probabilidad *p* y otro con probabilidad *1-p*). Es decir, el lanzamiento de una moneda “extraña” en la que no necesariamente ambas ca-ras son equiprobables.

La condición de terminación, a la que se hace referencia en el paso 12, puede definirse de muchas maneras. Se puede fijar un número máximo de generaciones que se pretende ejecutar el algoritmo, o puede decidirse hacer alto cuando la ma-yoría de la población, digamos el 85%, tenga una calificación que esté dentro de 0.6 desviaciones estándar de la media. En fin, opciones hay muchas. Generalmente depende del problema o de las preferencias personales la decisión acerca de cuándo es conveniente detenerse.

En el paso 4 se menciona que hay que seleccionar dos individuos con probabilidad proporcional a su calificación. Este tipo de selección proporcional es también llamado de “ruleta” (*roulette wheel selection*) por lo siguiente: supóngase que se suman las calificaciones de todos los individuos de la población y esta suma es considerada el 100% de una circunferencia. Luego, a cada individuo se le asigna el trozo que le corresponde de ésta según su aportación a la suma de las calificaciones. Es decir, si la calificación de un individuo es x_i entonces le corresponde un segmento de circunferencia dado por la simple regla de tres:

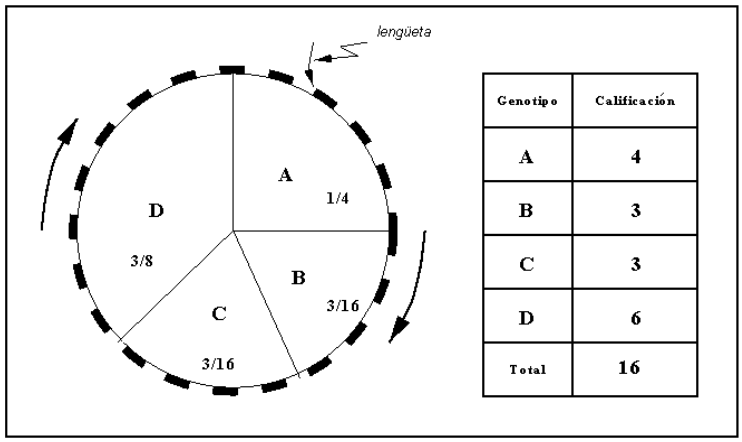
$$s = 2\pi \frac{x_i}{\sum_j x_j}$$

¿Qué ocurrirá entonces si se considera ésta cómo una ruleta y se coloca una lengüeta que roce el borde de ella? (figura F.I.3.A). La probabilidad de que dicha lengüeta quede en el arco correspondiente al individuo de calificación x_i, cuando la rueda se detenga tras realizar algunos giros, es:

$$p(i) = \frac{x_i}{\sum_j x_j}$$

LO QUE ES PROPORCIONAL A LA CALIFICACIÓN (X_I) DEL INDIVIDUO.

Aún queda por aclarar cómo es que se mezclan los códigos de dos individuos para formar dos híbridos. En general hay muchas maneras de hacerlo. Al lector mismo se le pueden ocurrir muchas, sin embargo, la que se usa en el AGS y una de las más populares, es el cruzamiento de un punto (*1-point crossover*) ya mencionado. En este tipo de cruzamiento, dados dos individuos se elige aleatoriamente un punto de corte entre dos bits cualesquiera del cromosoma. Esto define segmentos izquierdos y derechos en cada genotipo. Se procede entonces a intercambiar los segmentos derechos (o los izquierdos, indistintamente) de cada individuo. De esto resultan, al igual que en el caso del cruzamiento de cromosomas, dos híbridos (figura F.I.3.B).



Genotipo	Calificación
A	4
B	3
C	3
D	6
Total	16

FIGURA F.I.3.A: *Selección proporcional o por ruleta.*

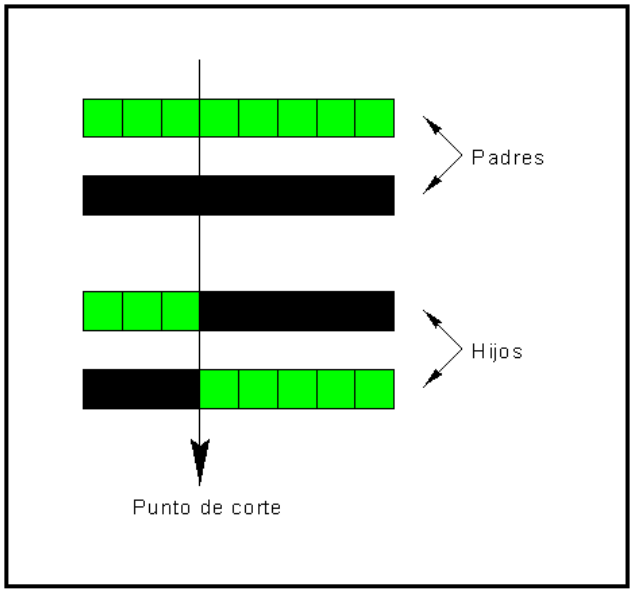


FIGURA F.I.3.B: *Cruzamiento de un punto (1-point crossover).*

A lo largo del texto se hará referencia frecuentemente al algoritmo genético simple como aquel caracterizado por tener:

1. Tamaño de población fijo en todas las generaciones.
2. Selección proporcional (de ruleta).
3. Cruzamiento de un punto. La probabilidad de cruza se mantiene fija para todas las generaciones y todas las parejas.
4. Mutación uniforme (todas las posiciones de las cadenas genéticas tienen la misma probabilidad de ser cambiadas). La probabilidad de mutación permanece fija para todas las generaciones y todas las posiciones de los individuos.
5. Selección no elitista. Esto es, no se copian individuos de una generación a otra sin pasar por el proceso de selección aleatoria (en este caso, proporcional).

Un ejemplo

SUPÓNGASE QUE SE PRETENDE ENCONTRAR EL PUNTO DEL INTERVALO [0, 0.875] DONDE LA FUNCIÓN:

$$f(x) = \left[1 - \left(\frac{11}{2}x - \frac{7}{2}\right)^2\right] \left[\cos\left(\frac{11}{2}x - \frac{7}{2}\right) + 1\right] + 2 \qquad \text{(I.3.A)}$$

TIENE SU MÁXIMO, USANDO PARA ELLO UN ALGORITMO GENÉTICO SIMPLE CON PROBABILIDAD DE CRUZA P_C = 0.70 Y PROBABILIDAD DE MUTACIÓN P_M = 0.0666... = 1/15. EN LA FIGURA F.I.3.C SE MUESTRA LA GRÁFICA DE LA FUNCIÓN.

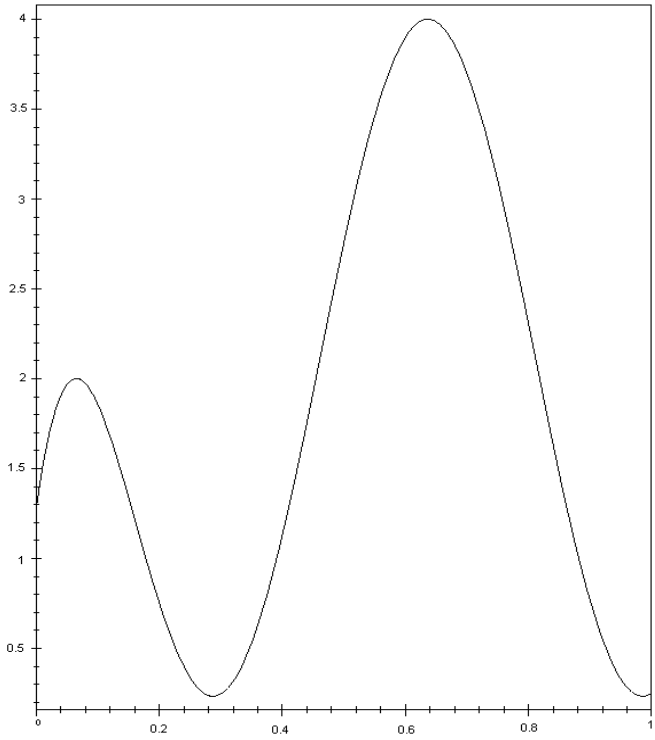


Figura F.I.3.C: *GRÁFICA DE* F(X) *EN EL INTERVALO [0, 1]. OBSÉRVESE QUE LA FUNCIÓN TIENE DOS PUNTOS ALTOS, UNO DE ELLOS ES EL MÁXIMO ABSOLUTO EN EL INTERVALO MIENTRAS EL OTRO ES SÓLO UN MÁXIMO LOCAL.*

El primer paso del AGS es codificar el dominio de la función, que en nuestro caso es el intervalo [0, 0.875]. Hay que tomar en cuenta que siempre estos códigos serán almacenados en la memoria de la computadora donde se ejecutará el algoritmo genético. Se pueden pensar muchas posibles maneras de codificar los elementos del dominio. Es posible, por ejemplo, colocar cada uno de los primeros *k* dígitos decimales del número (recuérdese que el dominio está constituido por números entre cero y 0.875) fijando el valor de *k* arbitrariamente; se puede pensar también en guardar estos números simplemente de la forma en que son representados los números reales por los compiladores de lenguajes de alto nivel, es decir, en el formato de punto flotante. No importa cuál mecanismo de codificación se utilice siempre se tendrá, por supuesto, un número finito de códigos disponibles para representar un conjunto infinito y no numerable de elementos del dominio, así que hay siempre una infinidad de elementos del dominio que no son representados. Con el fin de simplificar el ejemplo se codificarán los elementos del dominio en binario, usando 3 bits en la representación. Evidentemente se tienen entonces sólo 2^3=8 números representados (la codificación se muestra en la tabla T.I.3.A), y se puede pensar en que cada número binario de tres bits es, en realidad, la expansión en notación binaria pesada de la parte deci-mal de un número. Así por ejemplo, 101 representa $1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$ = 1/2+ 1/8 = 0.625.

Tabla T.I.3.A: *CODIFICACIÓN Y EVALUACIÓN DE LOS ELEMENTOS DEL DOMINIO DE LA FUNCIÓN* F(X) *DEL EJEMPLO.*

No.	Código	E. Dominio *(x)*	*f(x)*
0	000	0.000	1.285
1	001	0.125	1.629
2	010	0.250	0.335
3	011	0.375	0.792
4	100	0.500	2.000
5	101	0.625	3.990
6	110	0.750	3.104
7	111	0.875	1.093

El máximo de la función en el intervalo se obtiene en:

$$x_m = \frac{7}{11} = 0.6363...$$

Y EL VALOR DE LA FUNCIÓN EN ESE PUNTO ES JUSTAMENTE 4. EVIDENTEMENTE x_M NO ESTÁ REPRESENTADO EN NUESTRA CODIFICACIÓN DEL DOMINIO. EL ELEMENTO MÁS CERCANO REPRESENTADO ES 0.625, ASÍ QUE SE ESPERARÍA QUE EL ALGORITMO GENÉTICO ENCONTRARA LA SOLUCIÓN APROXIMADA 101 QUE CODIFICA AL 0.625.

El siguiente paso es generar de manera aleatoria un conjunto de posibles soluciones. Nuevamente hay muchas formas de hacer esto: generar códigos válidos uniformemente distribuidos en el conjunto de posibles códigos, o generarlos con una distribución normal o de Poisson, etc. Para que el proceso sea lo más imparcial posible, se ha elegido aquí una distribución uniforme (aunque la población es tan pequeña que esto no es notorio). Se ha seleccionado un tamaño de pobla-ción de 5. En la tabla T.I.3.B se muestra la población inicial.

Ahora se procede a calificar a cada uno de los elementos de la población actual (en este caso la población inicial). En la tabla se muestra la calificación en la tercera columna (de izquierda a derecha). En el caso que nos ocupa se pretende maximizar una función. Ésta siempre toma valores no negativos en el intervalo que se codificó, así que un buen candidato para función de calificación, adaptación o *fitness* es, justamente, el valor mismo de la función. Es decir, la calificación de cada individuo será el valor de la función en el elemento del dominio que es codificado por dicho individuo.

Ahora se debe traducir esta calificación en una cierta medida de la oportunidad para reproducirse, es decir, de ser seleccionado para pasar a la siguiente generación o, al menos, pasar algunos descendientes. En la cuarta columna de la tabla T.I.3.B se muestra la probabilidad de selección de cada individuo de la po-blación inicial. Para calcular la probabilidad de selección del *i*-ésimo individuo de la población basta dividir la calificación de *i* entre la suma total de las califi-caciones de toda la población. Esta operación lleva implícita el mecanismo de selección uniforme o por ruleta que ya se mencionó. Pero no es necesario calcu-lar esta probabilidad de selección.

TABLA T.I.3.B: *Población inicial P_0 generada uniformemente para el ejemplo.*

Índice	Individuos	Calificación	Prob. de selección	Calif. Acumulada
0	000	1.285	0.200	1.285
1	001	1.629	0.253	2.914
2	001	1.629	0.253	4.543
3	111	1.093	0.170	5.636
4	011	0.792	0.123	6.428
	Suma	*6.428*	*0.999*	
	Promedio	*1.286*		

Para efectuar la selección se hace lo siguiente:

1. Se genera un número aleatorio $r \in [0, 1]$.
2. Se multiplica r por la suma de las calificaciones la población (S), obteniéndose $c = r\,S$
3. Se establece la calificación acumulada (C_a) y el índice actual en cero:
   $C_a = 0$, $i = 0$
4. A la calificación acumulada se le suma la calificación del i-ésimo individuo: $C_a = C_a + calif(i)$
5. Si $C_a > c$ entonces el i-ésimo individuo es seleccionado
6. Si no, entonces se incrementa i y se regresa al paso 4

En nuestro caso, en la primera selección efectuada:

r = 0.213
c = 0.213 × 6.428 = 1.369

Dado que 1.369 está entre 1.285 y 2.914, calificaciones acumuladas para el individuo de índice 0 y el de índice 1, respectivamente, entonces el individuo seleccionado es precisamente el de índice 1.

En la segunda selección:

r = 0.314
c = 0.314 × 6.428 = 2.018

QUE NUEVAMENTE CAE ENTRE LAS CALIFICACIONES ACUMULADAS PARA EL INDIVIDUO DE ÍN-DICE 0 Y EL DE ÍNDICE 1, ASÍ QUE NUEVAMENTE ES SELECCIONADO EL INDIVIDUO DE ÍNDICE 1. CABE ENFATIZAR QUE LA SELECCIÓN SE HACE *CON REEMPLAZO*, ES DECIR, UN MISMO INDIVIDUO PUEDE SER SELECCIONADO VARIAS VECES. CADA VEZ QUE SE SELECCIONA UN IN-DIVIDUO *NO SE QUITA DE LA POBLACIÓN* DE LA QUE SE ESTÁ SELECCIONANDO, PERMANECE EN ELLA PARA QUE PUEDA SER ELEGIDO DE NUEVO.

Ahora que ya se poseen dos individuos seleccionados (que en realidad son el mismo), se procede a determinar si deben o no ser cruzados. Para ello:

1. Se genera un número aleatorio $d \in [0, 1]$.
2. Si $d < p_c$ entonces la pareja seleccionada (A, B) debe cruzarse.
3. Sea l la longitud de la cadena que constituye el código genético de cada individuo. Se genera un número aleatorio (x) entero entre 1 y l-1. Se indizan los bits de los individuos seleccionados de izquierda a derecha: 1, 2, ..., l.
4. Se pegan los bits de índices 1, ..., x-1 de A y los de índices x, ... l de B para formar un individuo híbrido llamado C y los bits 1, ..., x-1 de B se pegan con los de índices x, ..., l de A para formar un individuo híbrido D.
5. C y D son los nuevos individuos.
6. Si $d \geq p_c$ entonces la pareja seleccionada (A, B) no se debe cruzar. A y B son los *nuevos individuos.*

En el ejemplo que nos ocupa, dado que A y B son ambos iguales a 001, entonces no importa si se cruzan o no, los nuevos individuos serán nuevamente 001 ambos.

Pero en una repetición posterior del ciclo resultaron seleccionados el indivi-duo 1 (001) y el individuo 3 (111), y en este caso al llevar a cabo el procedimiento descrito arriba:

d = 0.47

DADO QUE D < 0.70 = P_C ENTONCES 001 Y 111 DEBEN CRUZARSE

x = 1, así que A = 001 queda dividido en dos subcadenas: A_1 = 0, A_2 = 01 y B = 111 queda dividido en: B_1 = 1, B_2 = 11

Pegando A_1 y B_2 queda C = 011 y pegando B_1 y A_2 queda D = 101
C y D son los nuevos individuos

Cada vez que se tienen dos nuevos individuos se procede a ejecutar el procedimiento de mutación:

1. Se indizan nuevamente los bits de cada individuo de izquierda a derecha: 1, ..., *l* y para cada uno de los bits:
2. Se elige un número aleatorio q ∈ [0, 1]
3. Si $q < p_m$ entonces el bit en turno se invierte (si valía 0 se transforma en 1 y viceversa)
4. Si $q \geq p_m$ entonces el bit en turno permanece sin cambio.

Tabla T.I.3.C: *RESULTADOS DEL PROCESO GENÉTICO APLICADO A UNA CIERTA POBLACIÓN INICIAL P_0 PARA RESOLVER EL PROBLEMA DE EJEMPLO. LOS NÚMEROS EN CURSIVAS SON EL PROMEDIO DE LA COLUMNA.*

P_0	Calif.	P_1	Calif.	P_2	Calif.	P_3	Calif.
000	1.285	001	1.629	101	3.99	101	3.99
001	1.629	001	1.629	000	1.285	101	3.99
001	1.629	100	2	001	1.629	101	3.99
111	1.093	011	0.792	001	1.629	100	2
011	0.792	001	1.629	110	3.104	101	3.99
	1.285		*1.536*		*2.327*		*3.592*

En nuestro caso p_m = 1/15, es decir, en promedio sólo habrá un bit cambiado en cada generación. De hecho, en nuestro ejemplo el bit que resultó cambiado al obtener la primera generación fue el último bit de D = 101 que se acaba de obtener por cruzamiento, como resultado de esto se obtiene el 100 que aparece como individuo de índice 2 en la generación 1 mostrada en la tabla T.I.3.C. En esta tabla se muestra el proceso completo, generación a generación, y puede observarse cómo la calificación promedio de la población (números en cursivas al final de cada columna de calificación), se va incrementando conforme se avanza en el tiempo. Si al principio se tenía una población dispersa con una calificación promedio de 1.285, al final se obtiene una población donde casi todos los individuos son 101 (el código más cercano al óptimo real) y la calificación promedio es de 3.592. De hecho, la población tiende a poseer sólo el individuo 101 (es el único punto óptimo) y solamente debido al operador de mutación hay algún individuo diferente (100 en el caso de P_3). Nótese también que, en las primeras etapas del proceso se tiende a reproducir con más frecuencia el individuo 001, donde la función tiene un máximo local. Esta situación se mantiene hasta que se encuentra el individuo 101, donde la función alcanza (aproximadamente) su máximo global.

OTROS MÉTODOS DE OPTIMIZACIÓN

EN ESTA SECCIÓN SE MENCIONAN ALGUNOS MÉTODOS DE OPTIMIZACIÓN CON EL OBJETIVO DE QUE EL LECTOR TENGA UN PANORAMA DE LAS DISTINTAS ALTERNATIVAS QUE EXISTEN PARA EL USO DE LOS ALGORITMOS GENÉTICOS. ALGUNOS DE LOS MÉTODOS PRESENTADOS EFECTÚAN LA BÚSQUEDA DE PUNTOS ÓPTIMOS MEDIANTE ESTRATEGIAS QUE, EN BUENA MEDIDA, DEPENDEN DE EVENTOS ALEATORIOS. A ESTE TIPO DE ESTRATEGIAS SE LES CONOCE COMO *HEURÍSTICAS*. (LOS ALGORITMOS GENÉTICOS OPERAN DE ESTA MANERA.)

Métodos tradicionales

LAS TÉCNICAS QUE AQUÍ SE ENLISTAN HAN SIDO SELECCIONADAS PORQUE SON REPRESENTATIVAS DE LOS DOS GRANDES GRUPOS DE MÉTODOS DE OPTIMIZACIÓN: LOS DIRECTOS (QUE NO REQUIEREN MÁS QUE LOS VALORES DE LA FUNCIÓN OBJETIVO), COMO SIMPLEX, Y LOS DE ASCENSO (QUE, ADEMÁS, REQUIEREN LOS VALORES DE LA O LAS DERIVADAS DE LA FUNCIÓN), COMO EL DE NEWTON Y EL DE ASCENSO DE MÁXIMA PENDIENTE. EXISTEN OTROS MUCHOS MÉTODOS DENTRO DE ESTA CLASIFICACIÓN QUE SON BASTANTE MÁS ÚTILES QUE LOS QUE SE MUESTRAN AQUÍ. SIN EMBARGO, EL OBJETIVO DE ESTA SECCIÓN ES PRESENTAR, CON PROPÓSITOS DIDÁCTICOS, UN PANORAMA REPRESENTATIVO DE LOS MÉTODOS DE OPTIMIZACIÓN NO HEURÍSTICOS.

Método de Newton

EN GENERAL, LA FÓRMULA DE NEWTON ESTÁ DADA POR:

$$x_r = x_{r-1} - \frac{f'(x_{r-1})}{f''(x_{r-1})} \qquad \text{(I.4.B)}$$

El lector interesado en profundizar en este método puede consultar [1] o bien algún texto de análisis numérico.

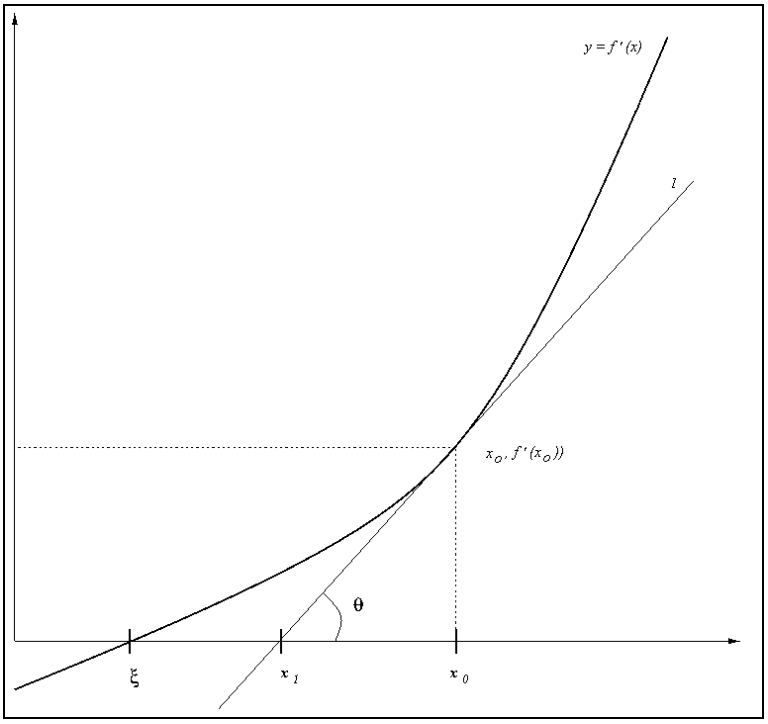


Figura F.I.4.A: *ILUSTRACIÓN DEL MÉTODO DE NEWTON PARA HALLAR LOS PUNTOS DONDE* F' *SE ANULA.* X_0 *ES EL PUNTO INICIAL, A PARTIR DE ÉSTE SE ENCUENTRA* X_1.

Búsqueda de Fibonacci

ESTE MÉTODO SIRVE PARA BUSCAR EL MÁXIMO O MÍNIMO DE FUNCIONES DE $\mathbb{R}\rightarrow\mathbb{R}$. EL OBJETIVO ES ACOTAR EL PUNTO DONDE LA FUNCIÓN ALCANZA SU VALOR EXTREMO POR INTERVALOS CADA VEZ MÁS PEQUEÑOS, DE ESTA FORMA AL FINAL SE OBTIENE UN INTERVALO $(X, X+\varepsilon)$ EN CUYO INTERIOR ESTÁ EL PUNTO BUSCADO.

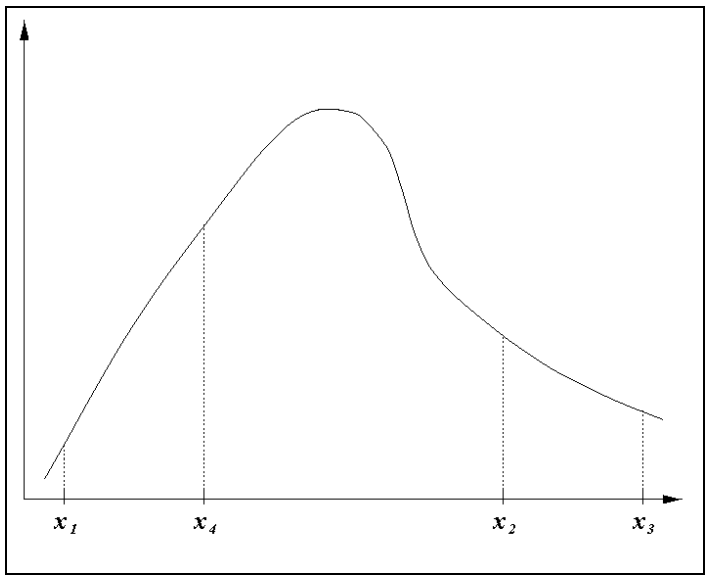


FIGURA F.I.4.B: *Ilustración del método de búsqueda de Fibonacci.*

La sucesión de números de Fibonacci es: $F_0=1$, $F_1=1$ y $F_k = F_{k-1} + F_{k-2}$ para $k=2,3,...$ Por lo que es posible escribir:

$$L_{n-j} = F_{j+1}L_n - F_{j-1}\varepsilon \qquad j=1,2,...,n-1$$

Esta es la razón del nombre búsqueda de Fibonacci. El lector interesado puede consultar [1] o bien [12].

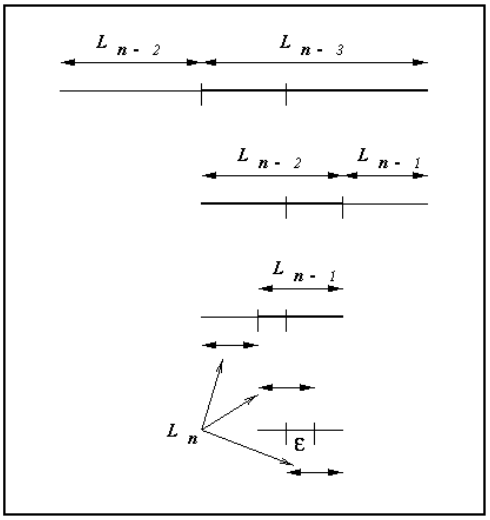


Figura F.I.4.C: *RELACIÓN ENTRE LAS LONGITUDES DE LOS INTERVALOS EN LA BÚSQUEDA DE FIBONACCI. CON LÍNEAS GRUESAS SE HAN DENOTADO LOS INTERVALOS DE INCERTIDUMBRE.*

Ascenso de máxima pendiente

EN LOS CURSOS DE CÁLCULO DIFERENCIAL DE VARIAS VARIABLES SE ENSEÑA A LOS ALUMNOS QUE EL GRADIENTE DE UNA FUNCIÓN $F(X)$, QUE SE DENOTA CON $\nabla F(X)$, ES UN VECTOR QUE APUNTA EN LA DIRECCIÓN DE MAYOR CRECIMIENTO DE F.

Se define un proceso iterativo donde:

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \lambda_{i-1}\nabla f(\mathbf{x}_{i-1}) \qquad \text{(I.4.D)}$$

En esta expresión λ_{i-1} maximiza:

$$\phi(\lambda) = f[\boldsymbol{x}_{i-1} + \lambda\nabla f(\mathbf{x}_{i-1})]$$

Este valor puede hallarse por procedimientos de búsqueda unidimensional como el de Fibonacci. En ocasiones esta "complicación" se omite y el valor de λ_{i-1} de la expresión I.4.D es determinado por procedimientos más empíricos. Por ejemplo, si en alguna iteración al utilizar un valor λ_i el resultado de la función en el nuevo punto no es mayor que en el actual, entonces se reduce el tamaño del "paso" que se debe dar desde allí.

Para aclarar un poco más este método se resolverá un ejemplo. Sea:

$$z = f(x,y) = -(x - 5)^2 - (y - 3)^2 \qquad \text{(I.4.E)}$$

SE DEBE HALLAR EL PUNTO DONDE $F(X,Y)$ ALCANZA SU VALOR MÁXIMO (FIGURA F.I.4.D). ARBITRARIAMENTE SE ELEGIRÁ COMO PUNTO INICIAL $V_0 = (1,1)$. DE LA EXPRESIÓN I.4.E SE OBTIENE:

$$\nabla f(x,y) = (-2(x\text{-}5), -2(y\text{-}3))$$

DE DONDE $\nabla F(1,1) = (8,4)$. ASÍ QUE SE DEBE ELEGIR T_0 TAL QUE SE MAXIMICE:

$$\begin{aligned} f(t_0) &= f((1,1) + t_0(8,4)) \\ &= f(1 + 8t_0, 1 + 4t_0) \\ &= -(8t_0 - 4)^2 - (4t_0 - 2)^2 \\ &= -80t_0^2 + 80t_0 - 20 \end{aligned}$$

DERIVANDO ESTO E IGUALANDO A CERO:

$$f'(t_0) = -80(2t_0 - 1) = 0$$

DE DONDE:

$$t_0 = \frac{1}{2}$$

El nuevo punto sería, entonces: $\boldsymbol{v}_1 = (1,1)+(1/2)(8,4) = (5,3)$ y como $\nabla f(5,3) = 0$ se termina el proceso. El punto donde la función tiene su máximo es (5,3), el valor de la función en ese punto es 0. El lector interesado en profundizar puede consultar [9] ó [12].

Simplex

ESTE ES UN MÉTODO MUY USUAL PARA RESOLVER PROBLEMAS DE OPTIMIZACIÓN DE PROGRAMACIÓN LINEAL. EN ÉSTE SE BUSCA EL PUNTO DONDE UNA FUNCIÓN LINEAL DE VARIAS VARIABLES (LLAMADA FUNCIÓN OBJETIVO) TIENE SU MÁXIMO O MÍNIMO CUANDO SU DOMINIO SE VE RESTRINGIDO. LAS VARIABLES DEBEN TENER CIERTO SIGNO Y SUS VALORES DEBEN SATISFACER CIERTAS RELACIONES ENTRE ELLAS, TAMBIÉN LINEALES, EXPRESADAS COMO UN CONJUNTO DE ECUACIONES O DESIGUALDADES. UN EJEMPLO DE PROBLEMA DE PROGRAMACIÓN LINEAL ES MAXIMIZAR:

$$z = 2x_1 + 3x_2$$

$$\textit{restricciones} \begin{cases} x_1 + 2x_2 \le 6 \\ 2x_1 + x_2 \le 8 \end{cases}$$

$$\textit{restricciones de signo} \begin{cases} x_1 \ge 0 \\ x_2 \ge 0 \end{cases}$$

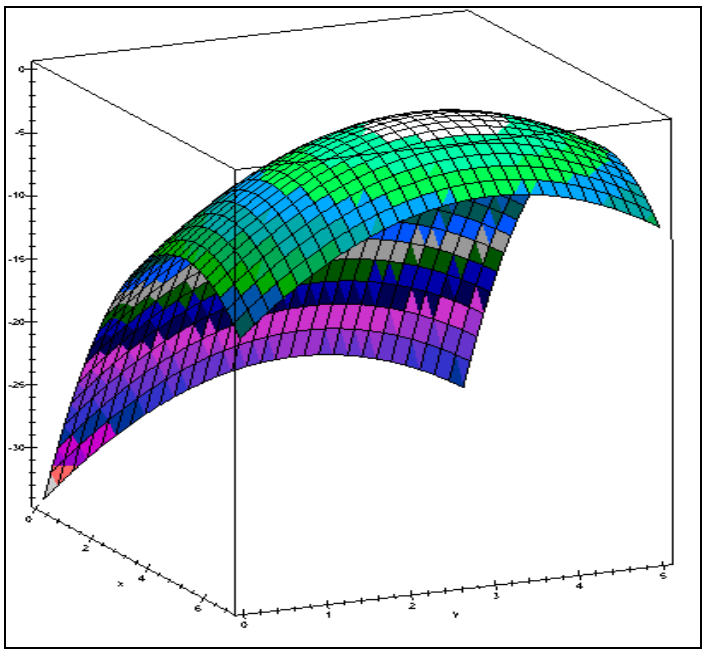


Figura F.I.4.D: *Gráfica de* $f(x,y) = -(x-5)^2-(y-3)^2$.

Los pasos del algoritmo Simplex son:

1. Convertir el problema a su forma normal.
2. Obtener una sfb, llamarla la sfb (solución factible básica) actual.
3. Determinar si la sfb actual es óptima; si lo es, detenerse.
4. Si no, determinar cuál de las actuales variables no básicas se convertirá en básica y viceversa, para obtener un mejor valor en la función objetivo.
5. Usar reducción de Gauss-Jordan para obtener la nueva sfb y regresar a 3.

Métodos heurísticos

Búsqueda tabú

ESTE MÉTODO DE BÚSQUEDA UTILIZA CIERTA MEMORIA O CONOCIMIENTO ACERCA DEL DOMINIO PARA ENCONTRAR UNA BUENA SOLUCIÓN AL PROBLEMA QUE SE PRETENDE RESOLVER, O UNA SOLUCIÓN, AL MENOS, CERCANA A LA ÓPTIMA. EL MÉTODO SE DERIVA DEL TRABAJO DE GLOVER [5], Y DESDE SU CREACIÓN HA RECIBIDO NUMEROSAS CONTRIBUCIONES Y SE HAN DESARROLLADO ALGUNAS VARIANTES.

Sea X el dominio de búsqueda relacionado con algún problema y sea $x \in X$ una posible solución. Para buscar puntos mejores en X se define, de alguna manera conveniente, la vecindad de un punto cualquiera en X. De esta manera es posible buscar soluciones mejores que x entre sus vecinos, a los que se denotará con $N(x)$. Conforme pasa el tiempo y se va explorando más el dominio, será conveniente excluir algunos puntos que se sabe no son buenos (para no buscar en vano). De manera que la vecindad de un punto x cambia con el tiempo, depende de la historia (H) del proceso de búsqueda. Así que es conveniente denotar con $N(x)$ sólo a la vecindad inicial de x, y con $N(H,x)$ la vecindad de x como función de la historia. Algunos puntos que estaban en $N(X)$ y que, al paso del tiempo, resulten no ser buenos candidatos de solución, serán marcados como *tabú* y no podrán ser alcanzados desde x, es decir, no serán elementos de $N(H,x)$. También, conforme transcurre el tiempo la función que evalúa la conveniencia de cada solución posible se puede modificar y, en general, se denota con $b(H,x)$. Al principio de la ejecución del algoritmo es deseable identificar regiones donde se obtienen buenas

soluciones y regiones donde se obtienen malas. Esto es posible si se posee suficiente diversidad, es decir, si se procura la exploración. En etapas tardías del algoritmo es mejor hacer una búsqueda más intensiva dentro de las regiones buenas para aproximarse al óptimo. Esto es, en general, válido para varias heurísticas.

Cuando es costoso determinar qué elementos están en $N(H,x)$ se selecciona una muestra representativa de la vecindad $NC(H,x) \subseteq N(H,x)$.

En la búsqueda tabú la aleatoriedad es, si no ignorada, por lo menos minimizada. Es empleada de manera muy restringida en el entendimiento de que la búsqueda ha de ser un proceso sistemático [4]. Por supuesto, existe una variante llamada búsqueda tabú probabilística (*probabilistic tabu search*) en la que, a partir de un elemento *x*, se selecciona aleatoriamente el vecino que debe ser visitado a continuación.

En general el algoritmo de búsqueda tabú es:

PROCESS (**TABÚ**)
begin
  $x = selecc(X)$
  $x^* = x$
  $H = \breve{}$
  $max = b(H,x)$
  calcular ($NC(H,x)$)
  elegir x' tal que $b(H, x') = max \{b(H,y) \mid y$ ¸ $NC(H,x)\}$
  repeat
    $x = x'$
    if ($b(H,x) > b(x^*)$)
    then
      $max = b(H,x)$
      $x^* = x$
    endif
    actualizar(H)
    actualizar($NC(H,x)$)
    elegir x' tal que $b(H, x') = max \{b(H,y) \mid y \in NC(H,x)\}$
  until (*cond. termina*)
  return (x^*)
end

El lector interesado en profundizar puede encontrar un análisis detallado de la búsqueda tabú en [4] y en [11].

Recocido simulado

AL IGUAL QUE EL MÉTODO DESCRITO EN LA SUBSECCIÓN ANTERIOR, EL RECOCIDO SIMULADO (*SIMULATED ANNEALING*) ES HEURÍSTICO, Y ESTÁ BASADO EN UNA ANALOGÍA CON EL PROCESO FÍSICO SEGUIDO PARA OBTENER SÓLIDOS CON CONFIGURACIONES DE ENERGÍA MÍNIMA. FUE PROPUESTO POR PRIMERA VEZ POR METROPOLIS EN [10] Y USADO EN OPTIMIZACIÓN COMBINATORIA POR KIRKPATRICK [8].

Probablemente el lector sepa que los sólidos cristalinos son los materiales más duros y que esto se debe a que su estructura molecular es muy simétrica, posee mínima energía. En Japón, desde hace milenios los maestros armeros elaboran los famosos sables de guerrero samurai siguiendo un proceso cuyos detalles se transmiten de generación en generación. Descrito superficialmente, este proceso consiste en calentar al blanco vivo una hoja de acero, doblarla sobre sí misma y golpearla con el martillo hasta que se integren las dos mitades y luego enfriarla sumergiéndola en agua. Estos pasos: calentar, doblar, golpear y enfriar se repiten decenas de veces y el resultado es una fina hoja de acero con el filo de una navaja de afeitar, pero con una dureza extraordinaria que la hace difícil de mellar.

Esto es porque el acero de la hoja, a través de los múltiples calentamientos y enfriamientos, adquiere una configuración molecular cuasicristalina, con una muy baja energía.

Este mismo proceso es el que se sigue en la física de materia condensada para obtener sólidos con estados de mínima energía. Se calienta el sólido hasta hacerlo líquido y luego se le enfría muy lentamente para que las partículas de éste se acomoden en su estado de mínima energía o en alguno cercano a éste. Si el enfriamiento no se hace cuidadosamente es más probable que no se alcance el estado de energía mínima. En este caso hay que volver a calentarlo (en general menos que la vez anterior) y volver a enfriarlo con lentitud.

El recocido simulado se adapta naturalmente a problemas de minimización. Evidentemente, el problema puede modificarse para convertirlo en uno de maximización, pero aquí se supondrá que se pretende resolver problemas de minimización. En términos muy generales, la simulación del proceso de recocido en el algoritmo es:

1. Sea E_i la energía actual. Se desplaza un poco una molécula del estado presente. Con esto se cambia su posición respecto a las demás y se altera la energía del sistema que ahora será $E_{i+1} = E_i + \delta$. El valor de δ es calculado.
2. Si $\delta < 0$ entonces el nuevo estado se acepta y es el nuevo estado actual. Si $\delta \geq 0$ entonces el nuevo estado es aceptado con cierta probabilidad dada por:

$$e^{\frac{-\mathrm{d}}{\mathrm{K}_B T}}$$

   donde T es la temperatura y K_B una constante física, conocida como constante de Boltzmann.
3. Si conforme transcurre el tiempo se disminuye lentamente la temperatura, entonces el sólido alcanza un equilibrio térmico en cada temperatura y la probabilidad de estar en el estado caracterizado por tener energía E_i es:

$$\frac{1}{Z(T)} e^{\frac{-E_i}{K_B T}} \qquad \text{(I.4.I)}$$

   donde $Z(T)$ es la función de partición definida como:

$$Z(T) = \sum_i e^{\frac{-E_i}{K_B T}}$$

De I.4.I es fácil ver que es mayor la probabilidad de tener un bajo nivel de energía (E_i pequeña, cercana a cero) y que esto es más difícil mientras menor sea la temperatura. De allí que sea importante disminuirla poco a poco.

El algoritmo descrito con más detalle es mostrado a continuación. En éste, X es el dominio de búsqueda, T_0 es la temperatura inicial, $N(i)$ es el conjunto de vecinos de i y $E(i)$ es la energía de la solución propuesta i.

```
PROCESS (RECOCIDO)
begin
  x = selecc(X)
  elegir T0
  T=T0
  repeat
    for (contador = 0) to L do
      elegir j ∈ N(i)
      δ = E(j)-E(i)
      if ((δ < 0) OR (aleatorio[0,1) < exp(-δ/T)))
      then
        i=j
      endif
```

```
   endfor
   reducir (T)
  until (cond. termina)
  return (i)
end
```

Nótese que, a diferencia de la búsqueda tabú, el estado actual del proceso en el recocido simulado depende sólo de su estado inmediato anterior. Esto ha hecho posible que se haga un análisis detallado de este método utilizando cadenas de Markov y, de hecho, ha sido probado teóricamente que el algoritmo converge asintóticamente a la solución global óptima. El lector interesado en profundizar puede encontrar un análisis detallado del recocido simulado en [3] y en [11].

Comparaciones

LOS MÉTODOS DE OPTIMIZACIÓN PRESENTADOS AQUÍ SON SÓLO UNA MUESTRA DE TODOS LOS EXISTENTES. EL LECTOR SE PREGUNTARÁ, ¿CUÁL DE ELLOS ES EL BUENO? EN OPTIMIZACIÓN, AL IGUAL QUE EN MUCHAS OTRAS SITUACIONES DE LA VIDA COTIDIANA, NO EXISTE "EL MÉTODO". LA CONVENIENCIA DE CADA UNO DEPENDE DEL PROBLEMA QUE SE PRETENDE RESOLVER. SI ÉSTE INVOLUCRA UNA FUNCIÓN CONTINUA Y DERIVABLE DE UNA SOLA VARIABLE INDEPENDIENTE, PROBABLEMENTE LOS MÉTODOS QUE SE ENSEÑAN EN LOS CURSOS DE CÁLCULO SERÁN SUFICIENTES. EN CAMBIO, SI EL PROBLEMA INVOLUCRA UNA FUNCIÓN MUY COMPLICADA DE DECENAS DE VARIABLES INDEPENDIENTES, DE LA QUE QUIZÁS NO SE CONOCE LA REGLA DE CORRESPONDENCIA, ENTONCES MUY PROBABLEMENTE (SI NO SE TIENE CIERTA PREDISPOSICIÓN AL SUFRIMIENTO) SE ELIJA UNO DE LOS MÉTODOS HEURÍSTICOS. ENTRE ESTOS DOS EXTREMOS HAY UNA AMPLIA VARIEDAD DE PROBLEMAS Y MÉTODOS QUE SE PUEDEN UTILIZAR. HAY MUCHO DE DONDE ESCOGER. LA ELECCIÓN DEBE IR EN PROPORCIÓN AL PROBLEMA. COLOQUIALMENTE, "NO DEBE USARSE UN CAÑÓN PARA MATAR PULGAS", PERO TAMPOCO SE DEBEN "CORTAR OLMOS CON MACHETE".

Hay, eso sí, métodos aplicables a casi cualquier problema, como los heurísticos, entre los que se encuentran los algoritmos genéticos y otros métodos que exigen mucho del problema, como los basados en el cálculo diferencial. En contraste, el grado de certeza que se puede tener en los resultados va en proporción inversa a la aplicabilidad. Casi nunca se puede estar completamente seguro de que el resultado entregado por un algoritmo heurístico sea el máximo o mínimo global o cuándo se debe detener el proceso de búsqueda.

En términos generales los métodos presentados aquí poseen las siguientes características:

Newton: La función objetivo debe ser continua, dos veces derivable, con primera derivada continua y cuya segunda derivada no se anule.
búsqueda de Fibonacci: La función objetivo debe ser continua y suave (sin saltos violentos).
ascenso de máxima pendiente: La función objetivo debe ser continua y derivable. Calcular el gradiente puede no ser fácil.
simplex: La función a optimizar debe ser lineal así como sus restricciones. Al ejecutarlo en computadora la exactitud del resultado depende de la estabilidad numérica de la matriz asociada.
métodos heurísticos: El problema general en estos métodos es determinar *a priori* los valores de los parámetros de operación del algoritmo, por ejemplo: cómo establecer la dependencia entre la vecindad y la historia en búsqueda tabú, cómo ir modificando con el tiempo la temperatura en recocido simulado o qué valor asignarle a la probabilidad de mutación en un algoritmo genético.

REFERENCIAS

[1] Bunday, B. D., *Basic Optimization Methods*, Edward Arnold Publishers, 1984.
[2] Chvàtal, V., *Linear Programming*, Freeman, 1983.
[3] Downsland, Kathryn, "Simulated Annealing", en *Modern Heuristic Techniques for Combinatorial Optimization Problems*, editado por Colin R. Reeves, John Wiley & Sons, 1993, pp. 20-69.

[4] Glover, F. y M. Laguna, *Tabu Search*, Kluwer Academic Publishing, 1997.
[5] Glover, F., "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers and Operations Research*, No. 5, 1986, pp. 553-549.
[6] Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
[7] Holland, J. H., *Adaptation in Natural and Artificial Systems*, 2a ed., MIT Press, 1992.
[8] Kirkpatrick, S., C. Gelatt y M. Vecchi, "Optimiztion by Simulated Annealing", *Science*, No. 220, 1983, pp. 671-679.
[9] Marsden, J., y A. Tromba, *Cálculo Vectorial*, Fondo Educativo Interamericano, 1981.
[10] Metropolis, N., A. Rosenbluth, A. Teller y E. Teller, "Equations of State Calculations by Fast Computing Machines", *The Journal of Chemical Physics*, Vol. 21, No. 6, 1995, pp. 1087-1092.
[11] Reeves, C. (editor), *Modern Heuristic Techniques for Combinatorial Problems*, John Wiley & Sons, 1993.
[12] Shoup, T., y Farrokh Mistree, *Optimization Methods with Applications for Personal Computers*, Prentice Hall, 1987.
[13] Smith-Keary, P., *Genetic, Structure and Function*, Macmillan Press, 1979.
[14] Winston, W. L., *Introduction to Mathematical Programming, Applications and Algorithms*, 2a ed., Duxbury Press, 1995.