

CONOCIMIENTO EN ACCIÓN: ASIGNACIÓN DE RECURSOS A FAMILIAS CARENTES MEDIANTE LA APLICACIÓN DE UN ALGORITMO GENÉTICO - PROYECTO KOINONÍA

MARIA R. DOS REIS¹ - MOISES E. BUENO² - DANIEL XODO³

Facultad de Ciencias Exactas-Grupo de Investigación en Informática de Gestión del Instituto de Investigación en Tecnología Informática Avanzada (INTIA)-Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA). ARGENTINA
dosreis@econ.unicen.edu.ar - bueno@econ.unicen.edu.ar - dxodo@exa.unicen.edu.ar

Fecha Recepción: Octubre 2013 - Fecha Aceptación: Abril 2014

RESUMEN

El proyecto Koinonía surge como iniciativa de Caritas de una parroquia de Tandil (Bs. As.-Argentina) y tiene como objetivo crear un conjunto de herramientas de gestión y ayuda a la toma de decisiones en la acción social que realiza dicha institución.

En su accionar, Caritas entrega un bolsón de alimento mensual a cada una de un conjunto de familias beneficiarias.

Ante esta realidad se evidencia el problema de realizar una asignación eficiente de los bienes disponibles para conformar un bolsón de alimentos que satisfaga la mayor cantidad de necesidades de cada grupo familiar.

El presente trabajo contempla la aplicación de un algoritmo genético cuyos principios fueron establecidos por Holland (1975) y que responde a la selección natural descrita por Darwin (1859), con la variante de que cada individuo es una asignación completa de recursos al grupo de familias asistidas y cada necesidad se instrumenta como las aptitudes que tienen los individuos. De esta forma los individuos compiten por los recursos y se resuelve el problema de asignación de los mismos con métodos adaptativos de búsqueda y optimización.

PALABRAS CLAVES: Gestión del Conocimiento – Algoritmo genético – Optimización – Asignación de Recursos – Sistema de soporte a las decisiones

¹ Docente e Investigadora de la Facultad de Ciencias Exactas y Facultad de Ciencias Económicas - UNCPBA

² Docente e Investigador del Grupo de Investigación en Informática de Gestión (INTIA) - Facultad de Ciencias Exactas - UNCPBA

³ Profesor e Investigador del Grupo de Investigación en Informática de Gestión (INTIA) – Facultad de Ciencias Exactas - UNCPBA

ABSTRACT

The Project Koinonía arises as initiative of Caritas of Tandil's parish, Buenos Aires, Argentine, and it has as aim create a set of management tools and helps to the capture of decisions in the social action that realizes the above mentioned institution.

In his to gesticulate, Caritas provides a handbag of monthly food to a set of needy families.

Given this evidence actually make the problem of efficient allocation of available goods to form a pocket of food that meets the most current needs of each family.

This paper considers the application of a genetic algorithm whose principles were established by Holland (1975) and respond to natural selection described by Darwin (1859) with the variant that each individual is a family group and each requirement is implemented as skills that individuals have. In this way individuals compete for resources and resolve the problem of allocating them with adaptive methods of search and optimization.

KEYWORDS: Management Knowledge – Genetic Algorithm – Optimization – Resource Allocation – Decision support systems.

1. INTRODUCCIÓN

El Proyecto Koinonía surge como una iniciativa de Caritas en la ciudad de Tandil, provincia de Bs. As., Argentina, y tiene como objeto crear un conjunto de herramientas de gestión y ayuda a la toma de decisiones en la acción social que realiza dicha institución (Caritas Koinonía, sitio web).

Una extensión del proyecto a otras instituciones enmarcadas en la acción social se prevé para lograr el intercambio inter-institucional en el accionar comunitario.

Caritas realiza varios programas de ayuda social (Caritas, sitio web). Uno de ellos es la entrega de un bolsón alimentario de periodicidad mensual a cada una de un conjunto de familias beneficiarias. Si bien, actualmente la implementación del proyecto se está ejecutando en etapa de testeo en una única sede, este bolsón es entregado en las distintas sedes de cada una de las parroquias de la ciudad, donde esté vigente el programa de bolsones alimentarios.

El régimen de entregas mensuales obliga a considerar minuciosamente cuáles son los recursos disponibles y cuál es la mejor forma de realizar su asignación. El objetivo del Proyecto Koinonía, con respecto a este programa en particular, es crear una herramienta de soporte a las decisiones que ayude a la gestión del armado y distribución de bolsones alimentarios a las familias beneficiarias.

A su vez, cada sede puede administrar su propio depósito o un único depósito por parroquia; o bien una combinación de estas opciones, es decir un depósito por sede y un depósito general; el manejo de *stock* de esta problemática se torna muy relevante.

En el transcurso de los últimos años han sido evaluadas técnicas de administración de *stock* y métodos de asignación de alimentos a familias con necesidades básicas insatisfechas, para ser aplicados en este proceso (Bueno *et al.*, 2010; Bueno *et al.*, 2011; Illescas *et al.*, 2013).

Los problemas que surgen en la asignación están vinculados a diversas cuestiones que permiten dimensionar diferentes aristas del problema subyacente, se pueden contemplar algunas de ellas sin ánimo de ser exhaustivos.

Por un lado se encuentran las variables que corresponden específicamente a la asignación: una demanda de recursos superior a la oferta existente, temporalidad de la demanda, variabilidad de la oferta, y un conjunto de familias heterogéneas con necesidades extremadamente diferentes entre sí, de acuerdo a mediciones sociales y de necesidades básicas insatisfechas. En este sentido, la asignación debiera ser personalizada para cubrir las necesidades particulares de cada familia y contemplar las variaciones de oferta y demanda de recursos.

Por otra parte, se encuentran los requerimientos del decisor en cuanto a las capacidades de la solución propuesta: mayor o menor subjetividad en el proceso decisorio, la amigabilidad de la interfaz y búsqueda de una solución óptima o subóptima que facilite un proceso de equidad y transparencia.

Los diferentes modelos matemáticos desarrollados garantizan una alternativa de solución aceptablemente buena para uno o más criterios considerados prioritarios, dando lugar a procesos de investigación y desarrollo continuos que permitan abordar otros criterios. Los mismos se dan en un contexto altamente restrictivo en cuanto a la posibilidad de especificar todas las necesidades de los grupos familiares, condicionado al conjunto de requerimientos de los diferentes actores.

En este trayecto de investigación y desarrollo, se ha planteado la utilización de una solución metaheurística (Bueno *et al.*, 2011) y se está evaluando la implementación de métodos de análisis multicriterio. En ambos casos la subjetividad del decisor tiene un peso determinante en la asignación de recursos y, del mismo modo, induce un arduo trabajo por parte del decisor en la parametrización del sistema de toma de decisiones.

Continuando estos lineamientos, el trabajo contempla la aplicación de un Algoritmo Genético (AG) ubicado dentro la teoría de computación evolutiva (Schwefel, 1991).

En los antecedentes de la utilización del proceso evolutivo para la resolución de problemas se encuentran los trabajos de Fraser (1957), Friedberg (1958, 1959), Bremermann (1962), Fogel (1966), Holland (1969, 1975), Goldberg (1989), Davis (1991), entre otros, que responden a la selección natural descrita por Darwin (1859).

La elección de la utilización de AG para asistir en la toma de decisiones tiene su justificativo en la baja o nula presencia de subjetividad por parte del decisor en la asignación de los recursos.

El decisor sólo tiene que establecer cuáles son los las necesidades que se atenderán y sus importancias respectivas⁴, y con respecto a los recursos cuál es valor asociado a los mismos. Esta carga de subjetividad en la carga de parámetros es sustancialmente menor que en otros algoritmos utilizados, y es algo que algunos decisores demandan; es decir que el sistema plantee soluciones de asignación con la menor participación posible por parte del decisor.

El desarrollo del presente trabajo propone una variante de la aplicación del algoritmo a la temática planteada. La misma contempla que cada individuo sea una asignación de recursos que se realiza al conjunto de familias, fundamentado en que los bolsones y los recursos que lo integran son demandados por las familias; y cada necesidad que presente el grupo familiar es instrumentada como las aptitudes que tienen los individuos. De esta forma los individuos compiten por los recursos y se resuelve el problema de asignación de los mismos con métodos adaptativos de búsqueda y optimización.

⁴ En el algoritmo implementado esta configuración es opcional ya que se puede tomar como orden de importancia la posición de la opción en la pregunta. Por ejemplo, en las variables cualitativas ordinales como el grado de educación, donde el orden en sus opciones van desde postgrado completo hasta analfabeto (de menor a mayor necesidad).

2. DESARROLLO

2.1 Especificación del Problema

El problema a resolver se puede formalizar de la siguiente manera: Dado un Individuo (P) perteneciente a una Familia (F), el cual presenta 0 o más necesidades (N), para las cuales se demandan 0 o más recursos (R); se define necesidades de una Familia:

$$N_{fi} = \sum_{p=0}^{tmi} N_{pi}$$

Donde tmi es el total de miembros de la familia i , N_{pi} son las necesidades del individuo p perteneciente a la familia i .

Estas necesidades pueden escribirse en función del valor de los recursos necesarios para cubrirlas ponderado por las cantidades, de la siguiente manera:

$$N_{pi} = \sum_{k=0}^t R_{pik} * Cant_{R_{pik}}; \quad R_{pik} \geq 0 \text{ y } Cant_{R_{pik}} \geq 0$$

Donde R_{pik} es el valor del recurso k que satisface la necesidad N_{pi} del individuo p de la familia i , $Cant_{R_{pik}}$ es la cantidad demandada del mismo recurso y t es la máxima cantidad posible de recursos diferentes. El caso $R_{pik} = 0$ corresponde a un recurso sin valor, es decir que no satisface alguna necesidad básica principal.

Luego, el problema consiste en satisfacer la mayor cantidad de necesidades de las familias, sujeto a las restricciones de oferta de recursos, quedando planteado el siguiente funcional:

Maximizar ($\sum_{i=0}^n N_{fi}$); donde n es el total de familias.

2.2 Características de los Algoritmos Genéticos

Al trabajar con un AG se deben definir conceptos que están relacionados con la implementación.

En la codificación de las soluciones se representa a un individuo de la población con una secuencia de genes: el cromosoma. A cada uno de los valores posibles que puede tomar un gen se lo denomina alelo. Si la codificación es binaria, como en el caso desarrollado a continuación, existen dos posibles alelos por cada gen con valor 0 y 1.

Los AG hacen uso de la denominada función de evaluación, función objetivo o “*fitness*” que determina para cada individuo un valor de ajuste o de adaptación proporcional que representa qué tan hábil es dicho individuo para resolver el problema en cuestión, es decir cuál es la cercanía con la mejor solución posible (Rothlauf, 2006).

En su accionar, los AG contemplan la selección de individuos padres que originarán nuevos individuos. La mayoría de los métodos de selección de los AG se basan en los valores de *fitness* y un factor al azar; de tal forma de considerar los individuos con mejor *fitness* de ser elegidos con mayor probabilidad (Mitchell, 1997).

Tras la selección de padres, el proceso de recombinación genética o *crossover* es el intercambio de material genético entre individuos (padres) con el objetivo de dar origen a nuevos individuos.

En la utilización de AG también se considera la mutación genética, a fin de introducir diversidad genética en una población de individuos. La mutación constituye un operador genético que trabaja a nivel de individuos. El operador de mutación evalúa a cada individuo obtenido tras la recombinación y decide si es conveniente realizar alguna mutación o alteración de genes.

Finalmente si el AG se ha planteado correctamente, la media de todos los individuos en cada generación se incrementa hacia el óptimo global. El incremento uniforme del valor de adaptación se denomina convergencia.

Un gen converge a un valor particular cuando el 95% de la población comparte el mismo valor de alelo para ese gen (De Jong, 1975). La población converge cuando todos los genes han convergido.

2.3 Metodología

Se presenta a continuación un pseudocódigo de algoritmo genético simple a aplicar en esta problemática:

```
BEGIN /* Algoritmo Genético Simple */
  Generar una población inicial.

  Computar la función de evaluación de cada individuo.

  WHILE NOT Terminado DO
    BEGIN /* Producir nueva generación */
      FOR Tamaño población/2 DO
        BEGIN /*Ciclo Reproductivo */
```

- Seleccionar dos individuos de la anterior generación para el cruce (probabilidad de selección proporcional a la función de evaluación del individuo).
- Cruzar con cierta probabilidad los dos individuos obteniendo dos descendientes.
- Mutar los dos descendientes con cierta probabilidad.
- Computar la función de evaluación de los dos descendientes mutados.
- Insertar los dos descendientes mutados en la nueva generación.

END

IF la población ha convergido THEN Terminado:= TRUE

END

END

Conforme a las características de los AG se deben definir con precisión los conceptos asociados a la problemática o caso de estudio.

2.3.1 Implementación preliminar. Dificultades

En un principio se planteó una correspondencia entre la población inicial y el conjunto de familias a ser asistidas, representando a cada individuo (familia) como un vector de necesidades (cromosoma) que presenta o no una determinada necesidad (gen del cromosoma); codificada la presencia o ausencia con un 1 o un 0 (los valores de alelos) según corresponda.

En el contexto del problema analizado, el cromosoma es de longitud estática, es decir que en todas las ejecuciones del algoritmo genético realizadas en cada evento, los n individuos de la población presentan la misma cantidad m de genes; siendo n la cantidad de familias y m las necesidades evaluadas por familia.

A modo de ejemplo considerar dos familias $F1$ y $F2$ con ocho necesidades evaluadas en cada familia, las cuales corresponden a cada columna del vector. La representación de cada familia sería similar a la que se puede apreciar en la FIGURA 1:

$$F1 = 10100010 \quad y \quad F2 = 11010110$$

FIGURA 1: Representación de familias

El siguiente paso es computar la función de evaluación en cada individuo. En este punto y, conforme al objeto del Proyecto Koinonía, lo que se pretende es atender la mayor cantidad de necesidades de los individuos.

Sin embargo, no todas las necesidades tienen la misma implicancia y relevancia en todos los casos.

No es lo mismo que una familia presente como necesidad la presencia de un niño desnutrido en el hogar, que la necesidad de abastecimiento de alimentos para diabéticos.

A tal efecto, cada necesidad evaluada o gen, se pondera de acuerdo al criterio del decisor o en base a una escala preestablecida a tal fin, cuyos resultados se efectivizan en el vector de prioridades de necesidades a satisfacer. Luego la función de evaluación para cada individuo queda conformada:

$$f(x_i) = \sum_{j=1}^m x_{ij} * p_j$$

Donde:

m = cantidad de necesidades evaluadas por familia;

x_{ij} = necesidad j de la familia i , puede tomar valores 0 o 1;

p_j = ponderación de la necesidad j ;

Restricción de no negatividad de las ponderaciones: $p_j \geq 0$

Considerando que existan n familias a ser asistidas, la función de optimización para toda la población que se toma como objetivo de análisis en el AG se formula de la siguiente manera:

$$\text{MINIMIZAR } N(f(x_i)) = \sum_{i=1}^n f(x_i) = \sum_{i=1}^n \sum_{j=1}^m x_{ij} * p_j$$

El algoritmo genético comprende un ciclo en el cual se crean nuevas generaciones de individuos combinando dos padres y luego de esta recombinación, se aplica opcionalmente mutación y se vuelve a evaluar la función objetivo sobre los individuos mutados, pudiendo ser incorporados en la nueva generación.

Para seleccionar los padres a combinar, se tienen en cuenta diferentes técnicas de muestreo, las cuales pueden utilizarse solas o de forma combinada, las mismas se describen a continuación:

Selección elitista:

- Se calcula para cada familia la función de evaluación $f(x_i)$.
- Se ordenan en forma decreciente (de menor a mayor necesidad presente) las familias representadas por los cromosomas.
- El decisor establece un parámetro z que determina un número de familias que serán pasadas directamente a la solución.

$$F1_0 = 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0$$

FIGURA 3: Configuración inicial

y suponiendo que en el cruce número nueve (9) del ciclo se producen las siguientes configuraciones:

$$x_i \quad x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8$$

$$F1_8 = 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0$$

$$F1_9 = 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0$$

FIGURA 4: Configuraciones con recombinación

Se puede apreciar que los genes x_4 y x_6 pasaron de un estado de ausencia a presencia de necesidad, y considerando $F1_0$ como referencia de configuración inicial, se establece que es inadmisibile el cambio para x_4 pero si es permitido para x_6 . En definitiva, el resultado de la iteración en el ciclo número nueve (9) debiera ser:

$$F1_9 = 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0$$

FIGURA 5: Configuración resultante

El cambio de valor exigido en el gen x_4 constituye una mutación obligada en el algoritmo planteado. De la misma forma, si el decisor considera que algún gen particular de algún individuo debe ser modificado, contemplando lo expuesto anteriormente, podría hacerlo.

Analizando el proceso de división a través del punto de corte se presenta la dificultad que el gen x_1 de cada familia puede quedar invariable a través del proceso de combinación. Para evitar este inconveniente, de la misma forma que aleatoriamente se establece el punto de corte, las secciones a intercambiar se deben alternar a derecha e izquierda del punto de corte en forma aleatoria.

Bajo este esquema de trabajo, al someter esta implementación a pruebas se evidencian grandes dificultades.

Suponiendo que la familia $F1$ tiene la necesidad x_4 y x_6 insatisfecha, la representación sería 10000001 y la familia $F2$ tiene la necesidad x_2 y x_7 , y se representaría 01000010.

Cualquier entrecruzamiento entre estos individuos resultaría en puntos por fuera del dominio, por ejemplo 10000010 no es admisible al igual que 01000001.

Como éste hay otros ejemplos de valores cuyo entrecruzamiento produce individuos que no pueden ser evaluados, esto produce que el algoritmo malgaste recursos generando valores que en definitiva no exploran nuevo espacio de soluciones.

Los mecanismos de entrecruzamiento, en algoritmos genéticos, están indicados para implementaciones en donde las combinaciones de dos o más individuos son parte también del dominio de soluciones⁵.

2.3.2 Implementación realizada

Ante este conjunto de dificultades observadas en el punto anterior, se optó por establecer que un individuo no representaría una familia, sino una posible asignación de recursos, esto es:

$$A = (RF1, RF2, \dots, RFn)$$

RFx = recursos asignados a la familia x
x está comprendido entre 1 y n donde n es el total de familias.

La forma de evaluar la aptitud de una solución particular (A_i) es sumando las aptitudes individuales de la mismas, es decir, qué tan buena es una asignación particular a un individuo particular.

Para calcular una aptitud individual se multiplican dos valores indicadores correspondientes a esa asignación, conocidos dentro del algoritmo como Necesidad y Asignación. Por lo tanto la aptitud de una solución queda representada por la fórmula:

$$\alpha = \sum_{i=0}^n Asignacion(i) * Necesidad(i)$$

El indicador de Asignación se calcula en base a los recursos que se otorgan de forma que se asignen en forma variada y se prioricen los recursos de alta necesidad antes que los recursos de baja necesidad.

Para lograr esto, cada tipo de recurso cuenta con un valor que indica la importancia o ponderación de este tipo de recurso sobre los demás, por ejemplo el valor que podría tener un paquete de fideos es mayor que el valor que puede llegar a tener un paquete de mayonesa.

⁵ Algunas de estas cuestiones surgen en el proceso de revisión y evaluación del artículo, agradecemos las sugerencias realizadas.

Estos valores son asignados por los usuarios encargados de cargar los recursos a la base de datos.

Para una asignación particular (A_i) se define un indicador de aptitud correspondiente a la asignación realizada:

$$\phi = \sum_{i=0}^{\text{cant recursos diferentes}} VR(i) * \left(\frac{1}{e^{(CR(i))}} \right)$$

Donde $VR(i)$ es el valor del recurso i ; $CR(i)$ es la cantidad del recurso i asignado.

Esta fórmula garantiza que las mejores asignaciones sean las que tienen recursos de necesidad primaria (mayor VR) y que no se asignen muchos recursos de un mismo tipo a una mismo individuo (el valor "marginal" del recurso disminuye en función de su cantidad).

El indicador de necesidad de un individuo se calcula haciendo una suma ponderada de diferentes indicadores de nivel social que se encuentran en la base de datos acerca de la familia completa. Estas ponderaciones están asociadas a las encuestas que se realizan a las familias.

Entonces se puede decir que el AG propuesto tiene como objetivo lograr:

$$MAX(a) = MAX \left(\sum_{i=0}^n Asignacion(i) * Necesidad(i) \right)$$

Dentro de los requerimientos a la solución planteada, existe la asignación de los denominados kit básicos, que consisten en un conjunto de productos que se asignan a razón de uno o más a cada familia, en la medida de las posibilidades dadas por la oferta.

Para entregar estos kits, las familias se ordenan de mayor a menor necesidad presente, y se van asignando los kits hasta agotar el stock de los mismos.

Como corolario se puede decir que cada iteración consta de los siguientes pasos:

1. Se ordena la población anterior (en la primer iteración sería la aleatoria) de acuerdo a la función de aptitud explicada anteriormente, de esta manera se consigue tener ordenadas las soluciones generadas por mejor aptitud y nos permite poder realizar el paso tres.

2. Antes de realizar la siguiente generación de soluciones se comprueba que el algoritmo todavía no haya convergido, esta comprobación consta en comparar la aptitud de la mejor solución de la población actual con la de las anteriores. Si la diferencia entre la nueva mejor aptitud es despreciable en relación a las mejores aptitudes anteriores, entonces el algoritmo converge y se dejan de calcular nuevas generaciones pues se entiende que alcanzó una solución aceptable.
3. Este paso consta en realizar el operador de elitismo sobre la población existente de forma tal de dejar solo las mejores soluciones para poder seguir operando con el algoritmo, y dado que las soluciones ya están ordenadas por su aptitud, este paso resulta trivial.
4. Una vez realizado el elitismo el algoritmo procede a realizar la recombinación genética entre las mejores soluciones de la población actual para poder generar la nueva generación, en este paso se seleccionan dos soluciones al azar de las que forman parte del elitismo y se cruzan de manera tal que la nueva solución posee parte de la solución del primer padre y parte del segundo de forma tal que sea válida, es decir, que la nueva solución no exceda los recursos que se encuentren en el stock.
5. Generada la nueva solución, ésta tiene una probabilidad de mutar. La mutación de una solución consiste en quitarle recursos aleatorios a un individuo aleatorio y dárselos a otro individuo elegido al azar.
6. Cuando se han generado todas las nuevas soluciones, se juntan de forma tal que formen una nueva población junto con las mejores soluciones de la población anterior. Esto posibilita mantener la mejor aptitud generada anteriormente en caso de que toda la nueva generación creada tenga una menor aptitud que la anterior. La población creada será la entrada en la siguiente iteración del algoritmo (paso uno).

En el AG planteado se cargan los parámetros de convergencia a través de dos variables. La convergencia puede presentarse por la profundidad del AG, es decir el número de asignaciones posibles. También puede converger cuando al transcurrir una cantidad X de iteraciones, las asignaciones logradas no logran mejorar la asignación $A(i - X)$.

2.3.2.1 Ejemplificación del AG propuesto

Conforme a la representación de las asignaciones se puede generar una recombinación como, dadas dos asignaciones padres, se elige un lugar aleatorio por el que cortar los padres, la mitad derecha del primero más la mitad izquierda del segundo forman al hijo.

Por ejemplo si se tienen 10 recursos asignados de dos formas distintas (padres): (1, 2, 3, 4) y (4, 3, 2, 1)

Suponiendo que el aleatorio es 2, de esta forma el hijo sería: (4, 3, 3, 4). La primera variación que se presenta es que el hijo generado sin ningún control puede exceder la cantidad de recursos existentes en el stock, como sucede en el ejemplo anterior, que solo hay 10 recursos, pero el hijo tiene 14.

Para solucionar este inconveniente lo que se hace es asignar de a uno por vez intercalando desde el punto de partida, por ejemplo, si el hijo en primera instancia está vacío sería:

(0, 0, 0, 0) -> (0, 0, 3, 0) -> (0, 3, 3, 0) -> (0, 3, 3, 4) -> (4, 3, 3, 4)

Cada vez que se hace una asignación se verifica el stock para ver si existen los productos que se van a asignar. En el caso de arriba la última asignación no se hubiera hecho dado que (0, 3, 3, 4) ya alcanzó los 10 recursos asignados y terminaría el algoritmo.

El problema planteado bajo este esquema es que trata los recursos como todos diferentes pero no lo son. El algoritmo anterior asignó 10 recursos, pero puede haber asignado 10 paquetes de harina cuando solo había 5 y lo restante era azúcar, por ejemplo.

Para solucionar este problema se utiliza una función que obtiene el máximo por cada recurso disponible de cada asignación individual dado el stock remanente. En el ejemplo siguiente supongamos que la composición de la demanda de recursos de la asignación es: (4, 3, 3, 4), detallada de la siguiente forma: 3 azúcar y 1 de harina, 1 azúcar y 2 harina, 3 azúcar, 1 azúcar y 3 harina. Y se tiene como stock remanente 3 de azúcar y 7 de harina.

Los pasos de la asignación serían:

(0, 0, 0, 0) -> (0, 0, 3, 0) -> (0, 2, 3, 0) -> (0, 2, 3, 3) -> (1, 2, 3, 3)

En este caso la primer asignación se hizo bien (siempre es así ya que el padre es correcto) y se asignaron 3 azúcar agotando el stock de azúcar.

La segunda asignación perdió un elemento porque con haber asignado los primeros 3 ya se agotó en el stock, y se asigna 2 de harina.

En la tercera también se pierde un elemento al igual que en la cuarta que solo se puede asignar uno de harina. De esta forma los recursos son tratados como diferentes pero se evidencia un nuevo problema. Pueden quedar recursos sin asignar.

Ante este inconveniente, una vez terminada de hacer una asignación, se recorre el excedente de los recursos y se asignan aleatoriamente a las familias hasta agotar el stock.

Esta última distribución aleatoria bien podría ser siguiendo un seguimiento de las necesidades de las familias comenzando por las más necesitadas y siguiendo por las menos o, de las menos a las más ya que éstas seguramente recibirán menos recursos. La forma de tratamiento de estos recursos será tratada en trabajos posteriores.

2.4. Caso de estudio

El sistema desarrollado se aplica en forma de prueba piloto en una Parroquia de la Ciudad de Tandil, que asiste a través de la entrega de bolsones de alimentos a unas 66 familias.

A los fines del sistema, se define familia como el conjunto de personas que habitan bajo el mismo techo. En promedio, el número de integrantes por familias es de 11 personas.

Las personas asistidas integran mayoritariamente edades comprendidas entre 10 y 20 años y existen relevadas 136 personas menores de 10 años y 5 personas mayores de 60 años. La disparidad de edades presenta la obligación de atender necesidades diferentes.

El relevamiento de las necesidades se realiza a través de dos encuestas. La primera encuesta se hace al titular o responsable del grupo familiar, está integrada por 75 preguntas y contempla aspectos de datos personales tales como educación, salud y trabajo; datos de la vivienda como material de construcción, servicios, etc.; y datos de situación social que se traducen en una evaluación que realiza un profesional en la asistencia social del grupo familiar en su conjunto. La segunda encuesta se completa con cada uno de los integrantes del grupo familiar (a excepción del titular) y contempla solo datos personales. Está compuesta por 42 preguntas.

Los bolsones que se entregan están afectados por el caudal de donaciones principalmente y por la atención de problemáticas particulares y estacionarias en segundo término.

En general se incluye 1 kg. de azúcar, 1 kg. de fideos guiseros, 1 salsa de tomates, 1 kg. de harina, 1 botella de aceite o 1 paquete de yerba, 1 kg. de arroz o polenta.

De acuerdo a las existencias se refuerzan con alimentos como: leche en caja, cacao, arvejas, lentejas, paté, té y mate cocido.

Por donaciones directas de tipo “a granel”, y en general de carácter estacional, se pueden entregar en forma fraccionada papa, zapallo y zanahoria.

En cuanto a los productos navideños, se suele hacer un brindis en diciembre (cuando se entregan los bolsones) y se dan productos en febrero (turrones, budines, pan dulce, garrapiñadas).

Ocasionalmente y dependiendo de lo disponible en el Banco de Alimentos⁶, se pueden incluir alguno de los siguientes productos: galletitas, mayonesa, tallarines, aguas saborizadas, agua mineral, caldos, duraznos en lata, mermelada, golosinas.

La información relevada en las encuestas, como así también la disponibilidad de recursos, se almacenan en una base de datos POSTGRESQL.

En cuanto a las necesidades definidas en la implementación del caso de estudio se consideraron cantidad de integrantes del grupo familiar, escolaridad, trabajo y situación social.

La razón por la que se consideran estos factores es porque propone asistir a la mayor cantidad de personas posibles, con bajas oportunidades laborales de conseguir trabajo por su grado de escolaridad y que remiten una evaluación negativa en el contexto del grupo familiar.

2.4.1 Desarrollo del sistema

EL sistema se desarrolló con lenguaje de programación .php, en un entorno web y con base de datos Postgresql. En el momento de ejecución de las corridas había 18 recursos distintos con stock (de un total de 31), con un volumen total de 1201 unidades de recursos.

El kit básico está integrado por: 1 kg. de harina, 1 kg. de azúcar, 1 kg. de fideos, 1 salsa de tomates, 1 botella de aceite y 1 kg. de arroz. Por disponibilidad temporal, al momento de las corridas, con los recursos del kit básico se pueden armar un total de 70 kits.

⁶ Banco de Alimentos es una ONG que recibe donaciones de mayoristas o grandes cadenas de supermercados y vende las mismas a un valor simbólico (muy por debajo del valor de mercado); lo recaudado a instituciones sin fines de lucros que ayudan en la alimentación de los más necesitados.

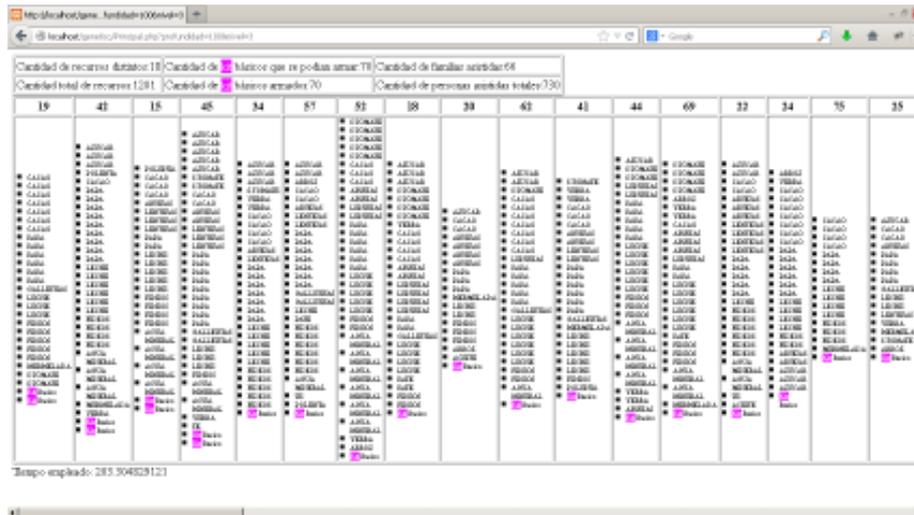


FIGURA 6: Grado de profundidad 100 – Nivel de convergencia 3

En la primera figura se muestra, con un grado de profundidad en el AG de 100 y un nivel de convergencia de 3, cómo se distribuyen los recursos empleando un tiempo de 203.30 segundos para realizar la asignación.

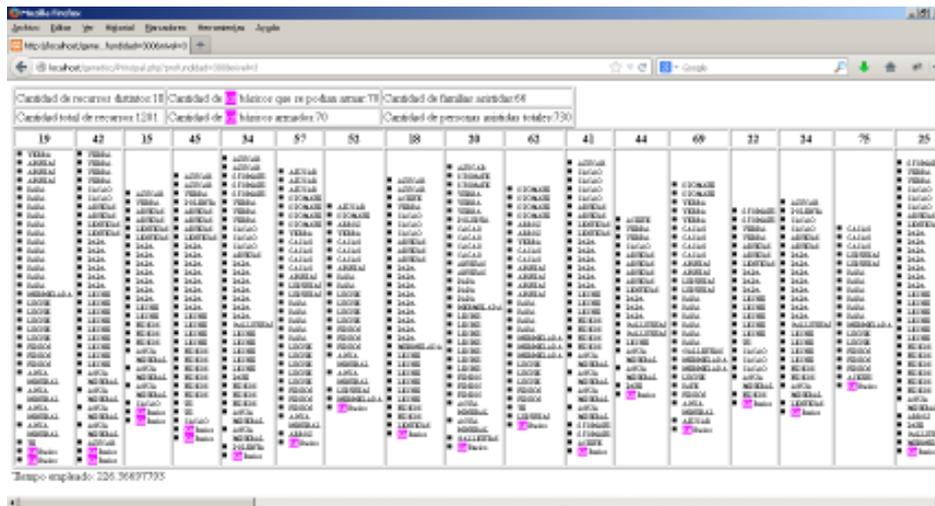


FIGURA 7: Grado de profundidad 300 – Nivel de convergencia 3

En la segunda figura el grado de profundidad se elevó a 300 manteniendo el nivel de convergencia en 3 y se consigue una mejor distribución de los recursos, aunque con detrimento del tiempo empleado que ahora es de 226.36 segundos.

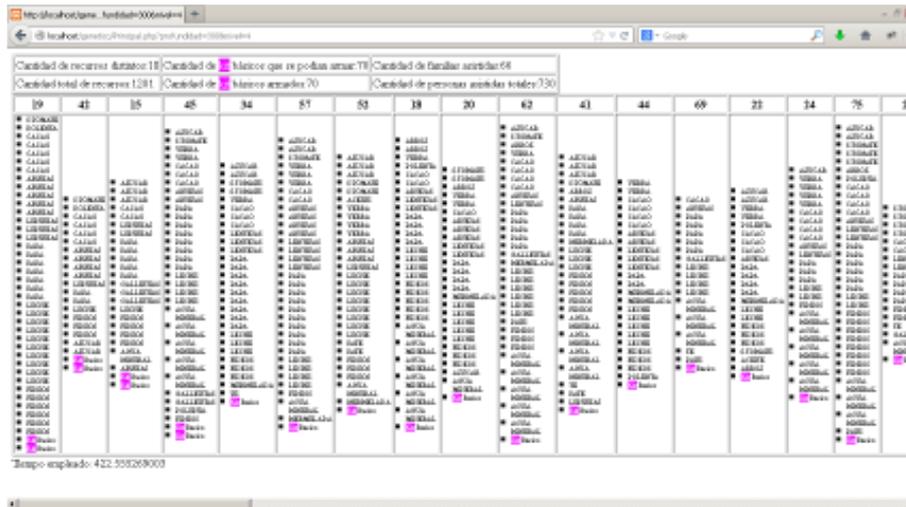


FIGURA 8: Grado de profundidad 300 – Nivel de convergencia 4

En esta tercera figura el grado de profundidad se mantiene en 300 pero se fija en 4 el nivel de convergencia. El tiempo empleado se eleva a 422.52 segundos logrando una asignación similar a la anterior.

En las gráficas presentadas, las familias están ordenadas de acuerdo al grado de necesidad presente en las mismas. Esto es de mayor a menor necesidad.

Desde el punto de vista computacional, el tiempo demandado por el AG puede ser considerado alto, aunque en la práctica no es un factor crítico ya que la asignación se produce una vez al mes y además el decisor no tiene gran carga de parámetros en el sistema. También se observó que el aumento de nivel de convergencia afecta en mayor medida el tiempo de respuesta del AG que el aumento de profundidad⁷.

3. CONCLUSIONES

La propuesta presentada en este trabajo expone la aplicación de un algoritmo genético simple con algunas variantes, contempla la optimización en la asignación de recursos alimenticios a familias con necesidades básicas insatisfechas, en el marco del Proyecto Koinonía.

⁷ En las pruebas realizadas, un nivel de convergencia de 6 no pudo efectuarse por el excesivo tiempo de procesamiento demandado.

El desafío presente en el trabajo radica en el contraste filosófico de pensar en un algoritmo genético de selección natural (Darwin, 1859) y la supervivencia del más apto; aplicado en un sistema de asignación de recursos a los más necesitados, es decir priorizar la debilidad sobre la fortaleza.

El objetivo planteado obliga a reformular el algoritmo por medio del diseño de equivalencia entre necesidades básicas insatisfechas con aptitudes o genes presentes en los individuos o cromosomas; siendo estos cromosomas las familias que presentan necesidades y en definitiva demandan recursos que son asignados a fin de maximizar la ayuda recibida. Lo que se evalúa en definitiva no es directamente la aptitud o debilidad de las familias, sino la asignación de recursos que maximice el beneficio y la ayuda brindada.

La existencia de recursos valuados en asociación a la ponderación de las necesidades, sea por importancia predefinida dada por el orden de opción en el relevamiento de encuesta, o por la carga de parámetros del sistema por parte del decisor; en conjunto con el grado de profundidad y nivel de convergencia deseado en la aplicación del AG constituyen el grado de subjetividad del sistema y condicionan la convergencia.

La aplicación del AG, conjuntamente con la implementación de otros métodos de decisión evaluados en esta problemática (Bueno *et al.*, 2010; Bueno *et al.*, 2011; Illescás *et al.*, 2013) permite brindar al centro decisor, quien debe armar los bolsones de alimentos o distribuir los recursos, herramientas de gestión del conocimiento y apoyo a las decisiones que le permitan evaluar la mejor alternativa.

A futuro se prevé una comparación de las metodologías aplicadas que posibilite sustentar la elección de las mismas de acuerdo a cada situación, así como la implementación de diferentes técnicas de muestreo que permita evaluar la precisión y eficacia del algoritmo en términos de sesgo⁸ y complejidad computacional.

REFERENCIAS

- BUENO, M.; DOS REIS, M.; ILLESCAS, G.; TRIPODI, G.; VALLEJOS I.; MÉNDEZ CASARIEGO I. (2010): "CONOCIMIENTO EN ACCIÓN: RANKING DE FAMILIAS SEGÚN NECESIDADES BÁSICAS INSATISFECHAS – PROYECTO KOINONÍA". II ERABIO, XXIII ENDIO, XXI EPIO. Septiembre de 2010, UNCPBA, Tandil, Buenos Aires, Argentina.

⁸ Sesgo: diferencia en valor absoluto entre el número real y el número esperado de descendientes de un individuo. Es deseable un sesgo de 0.

- BUENO, M.; DOS REIS, M.; ILLESCAS, G.; TRIPODI, G.; VALLEJOS, I.; MÉNDEZ CASARIEGO I. (2011): "CONOCIMIENTO EN ACCIÓN: MÉTODOS DE ASIGNACIÓN DE ALIMENTOS A GRUPOS FAMILIARES. PROYECTO KOINONÍA". Revista EPIO N° 32. Marzo 2011.
- BREMERMAN, M. F. (1962): "OPTIMIZATION THROUGH EVOLUTION AND RECOMBINATION". Self Organizing Systems. M. C. Yovits, G. T. Jacobi, and G. D. Goldstein (eds). Spartan Books, Washington D. C.
- CARITAS. SITIO WEB: <http://www.caritas.org.ar>
- CARITAS KOINONIA. SITIO WEB: <http://www.caritaskoinonia.org.ar>
- DAVIS, L. (1991): "THE HANDBOOK OF GENETIC ALGORITHMS". Van Nostrand Reinhold, New York.
- DARWIN, C. R. (1859): "ON THE ORIGIN OF SPECIES BY MEANS OF NATURAL SELECTION OR THE PRESERVATION OF FAVOURED RACES IN THE STRUGGLE FOR LIFE". London. J. Murray.
- DE JONG, K. A. (1975): "ANALYSIS OF BEHAVIOUR OF A CLASS OF GENETIC ADAPTIVE SYSTEMS". Tesis Doctoral. The University of Michigan. Technical Report N° 185, pp. 48-53.
- FOGEL, L. J., OWENS, A. J.; WALSH, M. J. (1966): "ARTIFICIAL INTELLIGENCE THROUGH SIMULATED EVOLUTION". John Wiley.
- FRASER, A. S. (1957): "SIMULATION OF GENETIC SYSTEMS BY AUTOMATIC DIGITAL COMPUTERS". Australian Journal Biological Sciences (10), pp. 484-499.
- FRIEDBERG, R. M. (1958): "A LEARNING MACHINE: PART I". IBM Journal (2), pp. 2-13.
- FRIEDBERG, R. M.; DUNHAM, B.; NORTH, J. H. (1959): "A LEARNING MACHINE: PART II". IBM Journal (2), pp. 282-287.
- GOLDBERG, D. E. (1989): "ZEN AND THE ART OF GENETIC ALGORITHMS". Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 80-85.
- HOLLAND, J. H. (1969): "ADAPTATIVE PLANS OPTIMAL FOR PLAYOFF-ONLY ENVIRONMENTS". Proceedings of the 2nd Hawai International Conference on System Sciences, pp. 917-920.

- HOLLAND, J. H. (1975): "ADAPTATION IN NATURAL AND ARTIFICIAL SYSTEMS". University of Michigan. Ann Arbor.
- ILLESCAS, G.; BUENO, M. E.; DOS REIS, M. R.; XODO, D.; PÉREZ, C. D.; RECOFSKY, E.; WEIMANN, R. H. (2013): "OPTIMIZACIÓN EN LA ASIGNACIÓN DE RECURSOS. INDICADORES DE TRANSICIÓN DE LA DECISIÓN A LA ACCIÓN". Anales XXVI ENDIO – XXIV EPIO. ISBN en trámite. Mayo 2013. Córdoba. Argentina.
- MITCHELL, T. (1997): "MACHINE LEARNING". McGraw-Hill. New York. USA.
- ROTHLAUF, F. (2006): "REPRESENTATIONS FOR GENETIC AND EVOLUTIONARY ALGORITHMS". Springer-Verlag Berlin Heidelberg, pp. 9-31.
- SCHWEFEL, H. P.; MÄNNER, R. (1991): "PARALLEL PROBLEM SOLVING FROM NATURE". Proceedings of the 1st Workshop PPSN I. Springer, Berlin, pp. 307-313.