

Tipo de artículo: Artículo original
Temática: Inteligencia artificial
Recibido: 2/04/2013 | Aceptado: 18/11/2013 | Publicado: 10/12/2013

Propuesta de técnicas evolutivas para la confección automática de tribunales de trabajos de diploma

A proposal of evolutionary techniques for automatic building of thesis evaluation teams

Pavel Novoa Hernández ^{1*}, Milvio A. Novoa Hernández ², Yasnalla Rivero Peña ³

^{1*} Departamento de Matemática, Universidad de Holguín. Ave XX Aniversario S/N. Holguín, Cuba. Correo-e: pnovoa@facinf.uho.edu.cu

² Oficina Nacional de Estadística e Información. Libertad No. 48 esq Línea, Reparto Loma. Holguín, Cuba. Correo-e: milvio@otehg.co.cu

³ Departamento de Redes, Universidad de Holguín. Ave XX Aniversario S/N. Holguín, Cuba. Correo-e: yasnalla@facinf.uho.edu.cu

Resumen

El Ministerio de Educación Superior de Cuba contempla, como uno de los ejercicios de evaluación final en las carreras universitarias, la defensa de un trabajo de diploma ante un tribunal conformado por profesionales. La confección de estos tribunales es una tarea desarrollada comúnmente por el coordinador de año basado en su intuición y experiencia previa. Sin embargo, esta tarea se torna compleja cuando el número de tesis y profesionales aumentan, y si además se tienen que cumplir con criterios como: la relación del profesional con el tema de la tesis, su disponibilidad, su grado científico, etc. Esta situación puede ser modelada como un problema de optimización combinatorio multicriterio que puede ser resuelto por métodos clásicos. Sin embargo, estos métodos necesitan, por lo general, explorar exhaustivamente el espacio de búsqueda que por su naturaleza combinatoria puede ser muy grande. En ese sentido, el presente trabajo propone un algoritmo evolutivo simple que resuelve de manera eficiente este problema. Como validación, se han seleccionado cuatro casos de estudio relacionados con la especialidad de Informática de la Universidad de Holguín.

Palabras clave: Computación Evolutiva, Confección de equipos, Optimización multiobjetivo, Trabajo de diploma

Abstract

The education system of Cuban states as final evaluation to obtain the bachelor degree the presentation of a thesis, which must be presented to an evaluation team of professionals. The task of building these teams is commonly developed by human experts based on its intuition and previous experience. However, this task usually becomes complex if the number of thesis and professionals increases, and if one must to fulfill several criteria like: the professional knowledge on the thesis topic, its availability, its scientific level, etc. This situation can be modeled as an optimization problem that can be solved by classic methods. However, classic methods need in general to explore the search space which can be considerably huge because its combinatorial nature. In that sense, the present work proposes a simple evolutionary algorithm that solves in an efficient way this problem. As validation, we have selected four cases of study related with the Informatics Engineering career at University of Holguin.

Keywords: Evolutionary Computation, Multiobjective optimization, Team Building, Undergraduated thesis

Introducción

Desde hace varios años el Ministerio de Educación Superior de Cuba estableció como evaluación final para la mayoría de las carreras el desarrollo de trabajos de diploma. Este trabajo tiene por característica que debe ser defendido ante un tribunal conformado por profesionales. Estos profesionales, son en su mayoría profesores del mismo centro universitario, específicamente de la misma facultad a la que pertenece el estudiante involucrado.

La confección de estos tribunales es desarrollada comúnmente por el coordinador de 5to año, el cual lo hace por lo general, auxiliándose de su experiencia e intuición. Sin embargo, se trata de una tarea que se torna compleja cuando el número de tribunales y profesionales aumentan debido al número de posibles variantes (equipos) que se pueden formar combinando los profesionales disponibles. Además, cuando se desea asignar un profesional a un tribunal, en la mayoría de los casos hay que cumplir con determinados criterios como el conocimiento de éste sobre el tema de la tesis y su disponibilidad de tiempo. A estos pueden sumarse otros criterios como la asignación de al menos un profesional con grado científico a cada tribunal, entre otros.

Dadas sus características, esta situación puede modelarse sin mucha dificultad como un problema de optimización, específicamente del tipo combinatorio multicriterio (multiobjetivo). Este problema puede ser resuelto mediante técnicas clásicas pertenecientes del campo de la Investigación de Operaciones (Kaufmann y Henry-Labordere, 1978; Griva et al., 2009; Rao, 2009). Sin embargo, la mayoría de estos métodos requieren de una exploración exhaustiva del espacio de búsqueda y necesitan de un tiempo considerable para obtener la solución óptima. Es aquí donde la aplicación de los métodos heurísticos y en especial de metaheurísticas (Glover, 1986; Melián, 2003), resulta adecuado con la intención de lograr un equilibrio entre complejidad computacional y la calidad de la solución.

Dentro de la amplia variedad de metaheurísticas que existen en la actualidad, los Algoritmos Evolutivos (Bäck et al., 1997; Bäck et al., 2000), han mostrado un gran desarrollo en los últimos 20 años. Estos algoritmos están inspirados en los principios de selección natural de Darwin que rigen la evolución de los seres vivos en la naturaleza en su adaptación al medio ambiente. Son algoritmos fáciles de implementar y con una alta efectividad en contextos diferentes (Burke 2005; Talbi, 2009), entre los cuales se encuentran precisamente problemas combinatorios (ej. el Problema del Viajante Vendedor).

En ese sentido, la presente investigación propone, como solución al problema de confección de tribunales, un algoritmo evolutivo simple y eficiente que obtiene buenas soluciones en un tiempo aceptable. Aunque existen diferentes métodos y enfoques para lidiar con problemas de este tipo, véase por ejemplo (Zitzler, 1999; Deb, 2005), el objetivo de este trabajo es presentar los resultados preliminares de esta investigación que se resumen en: 1) modelación del problema de confección de tribunales, 2) análisis de la complejidad del problema, 3) aplicación de un algoritmo evolutivo para resolverlo de manera aproximada y 4) validación empírica del algoritmo implementado mediante experimentos computacionales.

Metodología computacional

A continuación se exponen los principales pasos seguidos en la metodología computacional empleada para darle solución al problema de la confección de tribunales de trabajos de diploma.

Modelación del problema

Como se ha mencionado anteriormente, la tarea de la confección de tribunales puede ser modelada como un problema de optimización combinatorio multicriterio, el cual puede ser definido informalmente de manera siguiente: *encontrar las mejores asignaciones (tribunales) de cuatro profesionales (presidente, secretario, vocal, y oponente) para τ trabajos de diploma, a partir π profesionales, cumpliendo con los siguientes criterios: el uso del profesional debe estar en correspondencia con su disponibilidad, el conocimiento del tribunal en su conjunto tiene que ser el máximo posible sobre el tema del trabajo de diploma, y por último la distribución de profesionales con grado científico debe ser lo más uniforme posible, en particular de los doctores.*

La disponibilidad del profesor es una medida del número de trabajos en las que éste podrá participar como miembro del tribunal, por lo que es evidente que esta condición puede ser tratada como una restricción a la hora de asignar a algún profesor a un tribunal determinado. Sin embargo, en la práctica se puede comprobar que esta medida es una estimación, siendo posible en algunos casos violarla sin que por ello deje de ser factible la solución final. Por tal motivo, nos pareció conveniente incluirla como un elemento adicional en la función objetivo del modelo. De manera similar, los criterios: conocimiento del tribunal sobre el tema del trabajo de diploma, y asignación uniforme de másteres y doctores, han sido considerados como elementos de la función objetivo sin que por ello se perjudique el modelo matemático en cuestión.

Formalmente, cada tribunal puede representarse como una 5-upla $T_k = \langle t_k, p_1, p_2, p_3, p_4 \rangle$ (donde t_k trabajos de diploma, y p_i , profesionales del tribunal). Similarmente, una solución es una τ -upla formada por τ tribunales: $x = \langle T_1, T_2, \dots, T_\tau \rangle$, donde $x \in \Omega$ (espacio de búsqueda, posibles combinaciones de tribunales). De manera que el problema puede ser definido mediante el modelo siguiente:

$$\text{Min}Z(x) = w_1 e_{con}(x) + w_2 e_{disp}(x) + w_3 e_{grad}(x)$$

donde:

$$e_{con}(x) = \sum_{i=1}^{\tau} \sum_{j=1}^4 (1.0 - c_{ij})^2 \quad (\text{I})$$

$$e_{disp}(x) = \sum_{i=1}^{\|P_x\|} (d_i - u_{ix})^2 \quad (\text{II})$$

$$e_{grad}(x) = \sum_{i=1}^{\tau} (\phi_i - \phi^o)^2 \quad (\text{III})$$

Aquí τ es el número de tribunales, c_{ij} es el conocimiento que tiene el profesor j sobre el trabajo i , $\|P_x\|$ es el número de profesionales utilizados por la solución x . d_i es la disponibilidad del profesor i , y u_{ix} es el número de tribunales que tienen al profesor i en la solución x . ϕ_i es el número de profesionales con grado científico que forman parte del tribunal i , y ϕ^o el número ideal de estos. Los valores w_1, w_2, w_3 son pesos definidos por el experto (ej. el coordinador de año).

Es importante aclarar la forma en que se propone obtener el conocimiento (relación) de un profesor en un trabajo determinado. En este caso, se ha considerado que cada profesor y cada trabajo se relacionan con un conjunto de palabras claves no necesariamente iguales entre sí, de manera que para obtener el conocimiento c_{ij} se aplica la siguiente fórmula:

$$c_{ij} = \frac{\pi_{ji}}{\pi_i} \quad (IV)$$

Donde π_i es el número de palabras claves relacionadas con la trabajo i , y π_{ji} es el número de palabras claves del profesor j que coinciden con las palabras claves de la trabajo i ($\pi_i \neq 0$). Como $\pi_{ji} \leq \pi_i$, c_{ij} es un valor real en el intervalo $[0,1]$, donde un valor cercano a 0 indica poca relación o conocimiento del profesor en el tema de la trabajo, y un valor cercano a 1 lo contrario. En particular cuando $c_{ij} = 1.0$, el profesor domina totalmente el tema, por tal motivo se ha empleado la diferencia $(1.0 - c_{ij})$, que expresa el error respecto al valor ideal (ej. 1.0).

Por otro lado, la obtención de ϕ^o (asignación ideal de profesores con grado científico) es una estimación basada en el total de profesionales con grado y el número de tribunales. En efecto,

$$\phi^o = \text{ceil}\left(\frac{\phi_T}{\tau}\right)$$

donde ϕ_T es el total de doctores o másteres dentro del conjunto de profesionales disponibles, $\text{ceil}: R \rightarrow N$ es una función que devuelve la parte entera superior de un número real (ej. $\text{ceil}(0.14) = \text{ceil}(0.94) = 1.0$).

Se puede notar que la función Z es una suma ponderada de errores cuadráticos relacionados a los criterios: conocimiento, disponibilidad, y grado científico de los profesionales asignados a los tribunales. Al tratarse de una suma de errores, es evidente que el objetivo es minimizar a Z .

Complejidad del problema

Como se puede inferir de la sección anterior, se trata de un problema de optimización combinatorio. En la literatura existe una clase de problemas conocidos como *problemas de construcción de equipos (team building problems)* (Wegener, 2005; Hlaioittinun et al., 2008) que se asemeja al que motiva este trabajo. Sin embargo, existe una característica única en nuestro problema que lo diferencia de la clase *construcción de equipos*, y es que las personas pueden formar parte de varios equipos (tribunales) al mismo tiempo.

Matemáticamente, considere que cada tribunal k tendrá, dentro del conjunto de profesionales P , un subconjunto P_k de posibles candidatos, ($P_k \subseteq P$). Entonces, dado que cada equipo tiene que estar conformado por 4 profesionales, es fácil deducir que el número máximo de combinaciones posibles para crear el tribunal k viene dado por la siguiente fórmula:

$$C_k = \frac{\|P_k\|!}{4! (\|P_k\| - 4)!} \quad (V)$$

Mediante (V) es posible determinar el tamaño del espacio de búsqueda Ω (número de posibles soluciones) como:

$$\|\Omega\| = \prod_{i=1}^{\tau} C_i \quad (VI)$$

De (VI) se puede ver que esta función crece rápidamente conforme P y τ aumentan. Por ejemplo, supongamos que se desea confeccionar 40 tribunales, donde cada tribunal tiene a 30 profesionales como candidatos. Por (II) se tiene que

$\|\Omega\| = \left(\frac{30!}{4!26!}\right)^{40} = 27405^{40} \approx 3.25^{177}$, lo cual es una cantidad importante de posibles soluciones. Téngase en cuenta, que este conjunto también contiene soluciones que son por lo general indeseables, como las que emplean al mismo equipo de trabajo para todos los tribunales y que provocan la máxima sobreutilización de solo cuatro profesionales, y la subutilización del resto. Sin dudas, se trata de un problema complejo no solo para el ser humano, sino también para un ordenador convencional.

Algoritmo propuesto

Los algoritmos evolutivos (EA) se caracterizan por emplear una población de individuos (soluciones) que evolucionan en cada iteración (o generación) del algoritmo mediante operadores de *reproducción*. También pueden usar otros operadores de variación como la *mutación*, los cuales influyen en la adaptación continua de los individuos (Smith, 2008). Al final de cada iteración se aplica el operador de *selección* que es el encargado de seleccionar los mejores individuos de acuerdo a su *aptitud (fitness)*. Esta *aptitud* es, por lo general, el valor (o costo) de la función objetivo del problema de optimización en cuestión.

El algoritmo propuesto incluye algunos de los operadores evolutivos (mutación, cruzamiento, y selección) presentes en los paradigmas evolutivos Algoritmos Genéticos (GA), y Estrategias Evolutivas (ES). La principal dificultad que aparece en la adaptación de estos operadores a este problema, lo constituye la representación de los individuos de la población de soluciones (Bäck et al., 2000). Téngase en cuenta que los GA emplean codificación binaria, y en algunos casos codificación real, mientras que el paradigma ES emplea esta última. Sin embargo, en nuestro caso se necesita una codificación discreta basado en el modelo del problema descrito anteriormente. En lo que sigue se describirán los principales elementos del algoritmo propuesto.

En esencia, el algoritmo cuenta con los siguientes parámetros:

- μ : número de padres, empleados en el cruzamiento (crossover)
- λ : número de hijos
- p_m : probabilidad de mutación
- p_c : probabilidad de cruzamiento
- *maxEval*: número máximo de evaluaciones de la función objetivo en cada ejecución (condición de parada).

Por otro lado, los operadores se definieron de la siguiente forma:

- **Selección**: la selección es *elitista*, esto es, se escogen los mejores μ individuos de una población de $\lambda + \mu$ (hijos y padres), como criterio de ordenación se toman sus respectivos costos o *fitness* (evaluación en la función objetivo Z).
- **Cruzamiento**: este operador genera λ nuevos individuos a partir de los μ padres obtenidos en el operador anterior, siempre y cuando se cumpla que $rand < p_c$ ($rand \in R$, es un número aleatorio generado uniformemente en el intervalo $[0,1]$, $rand = U(0,1)$). Este cruzamiento se realiza estableciendo en cada componente del vector solución (tribunales) del nuevo individuo, la componente correspondiente de uno de los padres seleccionado aleatoriamente. En caso de que la condición booleana no se cumpla, entonces el individuo se genera totalmente de forma aleatoria en el espacio de búsqueda.

- **Mutación:** la mutación se aplica siempre que se cumpla que $rand < p_m$, $rand = U(0,1)$, y se lleva a cabo a nivel de tribunales. En efecto, suponga que se cumple la condición anterior, entonces a la solución x_i se le inserta un tribunal nuevo creado aleatoriamente con el conjunto de profesionales disponibles para el tribunal i . En caso contrario, se deja invariable la componente del vector solución.

Es necesario destacar que el operador de *Cruzamiento* es, junto con al conocido *cruzamiento intermedio* para problemas de optimización continuos, uno de los más comunes en los algoritmos basados en el paradigma ES (Beyer y Schwefel, 2002). Finalmente, los pasos principales del algoritmo propuesto, se muestran en la Figura 1.

```

1:  $P \leftarrow GenerarPoblaciónAleatoriamente(\mu)$ ;
2:  $Evaluar(P)$ ;
3: while  $eval < maxEvaldo$ 
4:    $P' \leftarrow Cruzamiento(\lambda, p_c, P)$ 
5:    $P'' \leftarrow Mutación(p_m, P')$ 
6:    $Evaluar(P'')$ ;
7:    $P \leftarrow Selección(\mu, P \cup P'')$ ;
8: end while
    
```

Figura 1. Algoritmo propuesto.

Resultados y discusión

Los experimentos se desarrollaron sobre cuatro casos de estudios elaborados según la experiencia en la confección de tribunales de la especialidad Ingeniería Informática de la Universidad de Holguín. Los factores que se variaron para la obtención de los casos de estudio fueron los pesos de los criterios, el número de tribunales y de profesionales (los cuales fueron establecidos 20 en ambos casos, por lo que $\|\Omega\| = 5.08^{73}$). En particular se consideraron los cuatro escenarios mostrados en la Tabla 1.

Tabla 1. Casos de estudios seleccionados.

Caso de estudio	w_1	w_2	w_3
CE_1	0.33	0.33	0.33
CE_2	0.1	0.3	0.6
CE_3	0.1	0.6	0.3
CE_4	0.2	0.4	0.4

La distribución de la disponibilidad y grados científicos de los profesionales se muestran en la Figura 2. Como información complementaria, en Tabla 2 se muestra la matriz de relación de conocimiento entre trabajos de diploma y profesionales, donde cada celda es calculada según la expresión (IV) (ver más arriba).

Tabla 2. Matriz de relación de conocimiento entre trabajos de diploma (TD) y profesionales (P)

T.D. P.	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18	p19	p20
t1	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t2	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83

t3	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t4	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t1	0,00	0,50	0,75	0,50	0,75	0,75	0,50	0,75	0,75	0,75	0,50	0,75	0,50	0,25	0,00	0,50	0,75	0,50	0,50	0,75
t2	0,44	0,89	1,00	0,67	1,00	1,00	0,67	1,00	0,78	1,00	0,89	1,00	0,67	0,44	0,00	0,67	1,00	0,67	0,67	0,89
t3	0,38	0,63	0,75	0,38	0,75	0,75	0,38	0,75	0,50	0,75	0,63	0,75	0,63	0,63	0,00	0,38	0,75	0,38	0,63	0,63
t4	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t1	0,67	1,00	1,00	0,67	1,00	1,00	0,67	1,00	0,67	1,00	1,00	1,00	0,67	0,67	0,00	0,67	1,00	0,67	0,67	1,00
t2	0,33	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,33	0,00	1,00	1,00	1,00	1,00	1,00
t3	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t4	0,25	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,50	0,00	0,75	0,75	0,75	0,75	1,00
t1	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t2	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t3	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t4	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t1	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t2	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t3	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83
t4	0,50	0,83	1,00	0,50	1,00	1,00	0,50	1,00	0,67	1,00	0,83	1,00	0,50	0,50	0,00	0,50	1,00	0,50	0,50	0,83

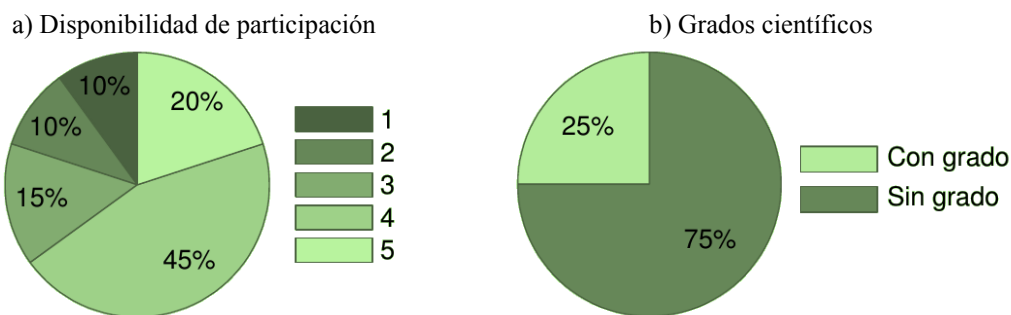


Figura 2. Distribución de la disponibilidad y grados científicos en los profesionales.

En cuanto al algoritmo, se consideró como factor más importante la probabilidad de cruzamiento, dejándose el resto de los parámetros establecidos de la forma siguiente: $\mu = 4, \lambda = 50, p_m = 0.1$. Para identificar de manera intuitiva las instancias del algoritmo, éstas serán denotadas de la siguiente forma: EAp_c . Por ejemplo, $EA80$ es un algoritmo en el que 4 padres generan 20 individuos en cada iteración, con probabilidades de cruzamiento y mutación 0.8 y 0.1, respectivamente.

Como medida de rendimiento se utilizó el propio costo (*fitness*) de la mejor solución en cada ejecución. De manera general, se realizaron 30 ejecuciones por cada par problema-algoritmo con semillas aleatorias diferentes. El número máximo de evaluaciones de la función objetivo por cada ejecución se estableció en 10000, la cual es una cantidad respetable si se tiene en cuenta la alta complejidad computacional de la función objetivo. Además, nuestro objetivo al establecer esta cantidad de evaluaciones, es simular un escenario real en el que el decisor necesita de una buena solución en un tiempo razonablemente corto. Como tecnología para la implementación del problema y algoritmos se empleó el lenguaje de programación Java™, el equipamiento utilizado para realizar los experimentos fue un ordenador PC Intel® Core i7 2600 (4 núcleos, 8 subprocesos) con 8GB de RAM.

Tabla 3. Media y desviación estándar del costo de la mejor solución.

	EA_{20}	EA_{50}	EA_{80}
CE1	7,28 (0,79)	6,74(0,68)	6,32(0,65)

CE2	4,46(0,87)	3,15(0,55)	3,03(0,73)
CE3	6,14(1,66)	5,20(0,96)	4,79(0,79)
CE4	6,43(0,74)	5,80(0,81)	5,32(0,75)

Los valores en negrita corresponden a los mejores.

De los resultados mostrados en la Tabla 3 es posible ver que el aumento de la probabilidad de cruzamiento resulta favorable para el algoritmo. Obsérvese que en todos los casos de estudio la mejor variante del algoritmo es EA_{80} , la cual utiliza una probabilidad de cruzamiento de 0.8. Aunque estos resultados sirven para comparar los algoritmos entre sí, poco pueden ayudar al experto o usuario final en lo relacionado con la calidad de las soluciones obtenidas en función de los criterios. Por tal motivo, a continuación se muestran gráficos de barras que indican la influencia de los errores de cada criterio en el valor de Z (Figura 3). Estos errores (e_{con} , e_{disp} , y e_{grad}) se obtienen de manera directa como aparece en las expresiones I, II, y III.

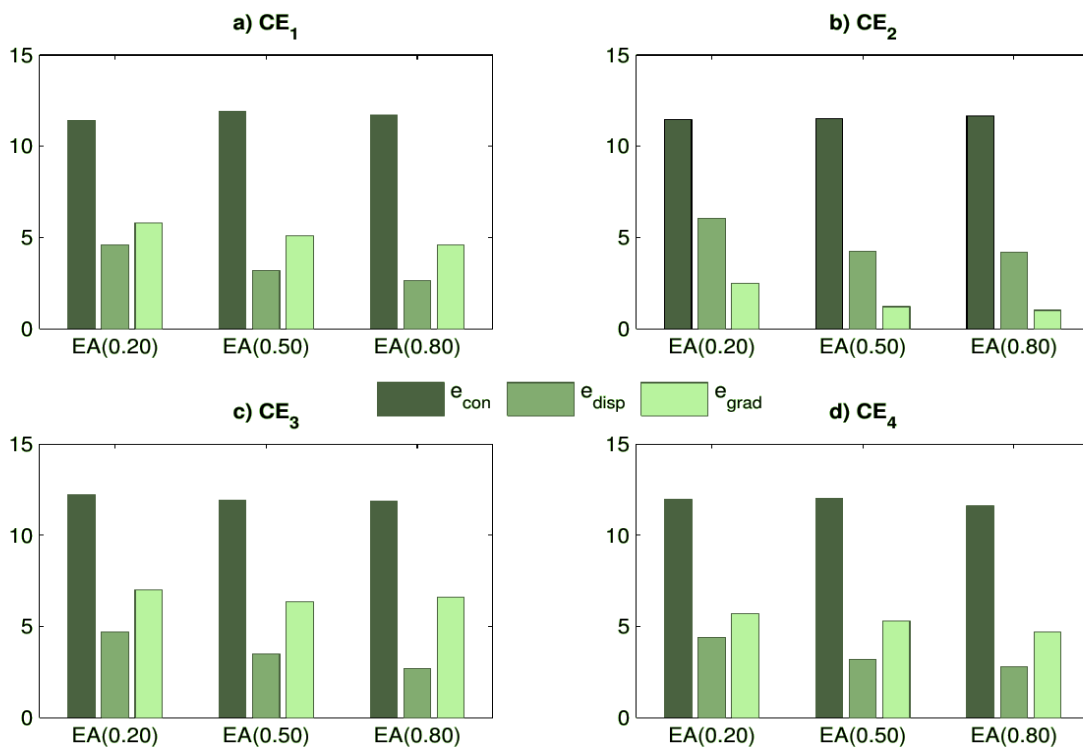


Figura 3. Gráficos de errores por criterios y algoritmos.

Como era de esperar, el algoritmo EA_{80} fue el que mejor rendimiento mostró también a nivel de criterios. Sin embargo, además del análisis del comportamiento de los algoritmos, en estos gráficos es posible apreciar que, para distintas configuraciones de pesos (w_1 , w_2 , y w_3) se obtienen resultados distintos como establece la teoría relacionada con Optimización Multiobjetivo (véase por ejemplo Burke, 2005). Los pesos guían al algoritmo hacia soluciones que en las que unos criterios tienen más importancia que el resto. Obsérvese la variación que sufren en los gráficos de CE1 y CE2 los criterios disponibilidad y grado científico. En CE2, el aumento de w_3 permite que el algoritmo realice una mejor distribución del personal con grado científico, observe que el error relacionado con este criterio e_{grad} disminuye. Por ejemplo, en el caso de EA_{80} el valor $e_{grad} = 1,0$ indica que como promedio se deja solo un trabajo de diploma sin ningún profesional con grado científico.

De manera general, se puede apreciar que para las instancias de problema consideradas, el algoritmo evolutivo aplicado resulta eficiente, pues minimiza aproximadamente la función objetivo del problema con un gasto bajo de tiempo.

Conclusiones

En este trabajo se han mostrado los resultados preliminares relacionados con la automatización de un problema común en las universidades cubanas: la confección de tribunales de trabajos de diploma. A partir de lo analizado, se puede concluir que el problema posee una alta complejidad debido al tamaño del espacio de búsqueda (posibles soluciones), el cual aumenta conforme el número de trabajos de diploma y profesionales crecen. El método propuesto para su solución, un algoritmo evolutivo simple, resuelve el problema de manera aproximada y con un gasto de recursos bajo, en especial de tiempo.

Como trabajo futuro se considera importante el estudio de las siguientes cuestiones:

- la complejidad del problema, tomando como partida la teoría de la complejidad computacional (Wegener, 2005) analizar aspectos como su *resolución algorítmica*.
- probar otros enfoques más sofisticados del área de la Computación Inteligente orientados a la optimización multiobjetivo (Deb et al., 2002; Zitzler et al., 2002), los cuales están basados en conceptos de dominancia y frentes de Pareto.
- la inclusión de este y otros algoritmos en una herramienta informática que gestione no solo el proceso de creación de los tribunales, sino también el sistema de evaluación que se lleva a cabo durante la investigación desarrollada por el estudiante.

Agradecimientos

Los autores desean agradecer a la dirección de la Facultad de Informática y Matemática de la Universidad de Holguín por la colaboración prestada durante el desarrollo de la presente investigación.

Además, los autores agradecen los comentarios y recomendaciones realizados por los revisores, los cuales contribuyeron en la calidad y presentación de este trabajo.

Referencias

- BURKE, E. K.; KENDALL, G. Search methodologies - Introductory Tutorials in Optimization and Decision Support Techniques. Springer Science + Business Media, LLC, 2005. 620 p.
- BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z. Evolutionary Computation 1. Basic Algorithms and Operators. Institute of Physics Publishing, Bristol and Philadelphia, 2000. 339 p.
- BÄCK, T.; HAMMEL, U.; SCHWEFEL, H.-P. Evolutionary computation: Comments on the history and current state. IEEE Transactions on Evolutionary Computation, 1997, 1(1): p. 3-17.
- BEYER, H.-G.; SCHWEFEL, H.-P. Evolution strategies - A comprehensive introduction. Natural Computing. Kluwer Academic Publishers, 2002. 1: p. 3-52.
- DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002. 6(2) : p. 182-197.
- DEB, K. Multi-objective Optimization. En: Search methodologies - Introductory Tutorials in Optimization and Decision Support Techniques. Springer Science - Business Media, LLC, 2005. p. 273-316.
- TALBI, E.-G. Metaheuristics: from design to implementation. John Wiley and Sons, 2009. 593 p.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 1986. 13(5): p. 533-549.

GRIVA, I.; NASH, S. G.; SOFER, A. Linear and nonlinear optimization. Society for Industrial and Applied Mathematics, 2009. 742 p.

HLAOITTINUN, O., BONJOUR, E.; DULMET, M. Multidisciplinary team building method based on competency modelling in design project management. *International Journal of Management Science and Engineering Management*, 2008. 3: p. 163-175.

KAUFAMNN, A.; HENRY-LABORDERE, A. Métodos y Modelos de la Investigación Operativa, Tomo III, Compañía Editorial Continental, Barcelona, España. 1978. 407 p.

MELIÁN, B.; MORENO PÉREZ, J.; MORENO VEGA, J. Metaheurísticas: Una visión global. *Revista Iberoamericana de Inteligencia Artificial*, 2003, 19: p. 7-28.

RAO, S. S. Engineering optimization : theory and practice - 4th edition, John Wiley & Sons, Inc., Hoboken, New Jersey, 2009. 813 p.

SMITH, J. Self-adaptation in evolutionary algorithms for combinatorial optimisation. *Studies in Computational Intelligence*, 2008, 136: p. 31-57.

WEGENER, I. Complexity Theory. Exploring the Limits of Efficient Algorithms. Springer-Verlag, Berlin Heidelberg, 2005. 308 p.

ZITZLER, E. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. thesis, Shaker Verlag, Aachen, Germany, 1999.

ZITZLER, E.; LAUMANN, M.; THIELE, L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. En: *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems*, Barcelona, Spain, 2002. p. 95-100.