

Inteligencia Artificial

Informe Final: Balanced Academic Curriculum Problem

Juan Pablo Escalona G.

19 de junio de 2015

Evaluación

Mejoras 1ra Entrega (10 %):	_____
Código Fuente (10 %):	_____
Representación (15 %):	_____
Descripción del algoritmo (20 %):	_____
Experimentos (10 %):	_____
Resultados (10 %):	_____
Conclusiones (20 %):	_____
Bibliografía (5 %):	_____

Nota Final (100):	_____
--------------------------	-------

Resumen

Balanced Academic Curriculum Problem, también llamado BACP, es un problema que busca asignar los ramos de una malla curricular en diferentes periodos de manera balanceada para que los alumnos puedan cursar exitosamente los ramos dependiendo de la carga (créditos) de estos, cumpliendo restricciones asociadas a la cantidad de ramos y carga en cada período. El BACP es un problema recurrente en universidades de todo el mundo, pero también puede ser aplicado en otras áreas como la asignación de carga de trabajo para los empleados de una empresa. En este informe se presenta el estado del arte del BACP resumiendo los diferentes enfoques que se han publicado para resolver este problema desde que fue publicado, comparando las técnicas utilizadas y los resultados.

1. Introducción

Hoy en día las entidades universitarias ofrecen distintas carreras a sus alumnos, estas carreras comprenden una malla curricular con ciertos ramos, que llevan asociado una dificultad. La correcta distribución de los ramos en los periodos académicos puede influir en el éxito o fracaso de los alumnos para terminar su carrera satisfactoriamente. Los ramos pueden comprender de ciertos prerrequisitos para ser cursados (e.g., matemáticas 2 necesita matemáticas 1). Además los ramos pueden cambiar

su nivel de exigencia para cursarlo exitosamente, lo que se le denominará créditos de ahora en adelante.

Un problema natural que surge es como asignar los ramos de la malla curricular de manera balanceada en su carga, i.e., que los créditos de los ramos de un periodo específico no sean desmedidos, pero que a su vez termine la carrera en un tiempo razonable estableciendo una cantidad máxima y mínima de ramos y créditos por periodo. A esto se le conoce como el Balanced Academic Curriculum Problem (BACP).

El BACP puede ser aplicado en otras áreas como es la asignación de carga de trabajo para los empleados de una empresa basada en turnos, e.g., carga y horarios de las enfermeras de un hospital [8]

En este informe se presentan en forma resumida los distintos métodos, técnicas y enfoques que han publicado los investigadores para representar y resolver el BACP comparando el rendimiento y la calidad de las soluciones. El informe esta organizado de la siguiente manera: En la sección 2 se define el problema y se muestran variantes, en la sección 3 se presentan los estudios realizados anteriormente haciendo comparaciones, luego en la sección 4 se establecen distintos modelos matemáticos y por último en la sección 5 se concluye y plantean futuras líneas de investigación.

2. Definición del Problema

2.1. BACP: el problema

El BACP definido originalmente en [5] busca resolver un problema recurrente para las universidades, asignación los ramos de una malla curricular en distintos periodos, balanceando la carga académica (i.e., similitud entre la carga de todos los periodos), pero a la vez cumpliendo con restricciones de precedencia y límites de exigencias y cantidad de ramos por periodo.

Ha sido demostrado que el BACP es un problema NP-complete [2, 7]

2.2. Definición del Modelo

El BACP definido en [1] comprende los siguientes elementos:

Una malla curricular es un conjunto de ramos que al completarlos se obtiene un título universitario. Cada malla curricular tiene definida una cantidad de periodos en que los ramos deben ser completados.

Los ramos pueden tener distintas dificultades que se relaciona al esfuerzo que un alumno debe aplicar para completar satisfactoriamente dicho ramo, es por esto que se le asocia a los ramos un número de créditos que representa el esfuerzo necesario para completar satisfactoriamente el ramo.

Por otra parte algunos ramos pueden tener prerequisites que definen una precedencia al ser cursados, es decir, si el ramo β tiene como prerequisite al ramo α , entonces β debe estar en un periodo académico posterior al periodo en que se encuentra asignado α .

La carga académica de un periodo es la suma de los créditos de los ramos que están presentes en dicho periodo. Esta carga académica debe cumplir un máximo para prevenir sobrecarga y un mínimo para ser considerado un estudiante de tiempo completo. Así mismo se establece un máximo y un mínimo de ramos por periodo.

El problema consiste en encontrar una asignación de los ramos en los distintos periodos. Dicha asignación debe satisfacer las restricciones de prerequisites, máximo y mínimo de créditos por

periodo, y máximo y mínimo de ramos por periodo. El óptimo busca minimizar la máxima carga académica de los periodos [1] o minimizar la diferencias de carga entre los periodos [7].

2.3. Problemas similares y variantes

El BACP es muy similar a problemas como el bin-packing, scheduling y balancing problems [7], e.g., los requisitos del BACP lo hacen similar al scheduling problem asignando precedencia en intervalos de tiempo.

La principal variante a este problema es la GBACP presentada en [3]. Esta variante agrega nuevas formulaciones que representan mejor el problema real de las universidades, e.g., se da la posibilidad de compartir ramos de mallas curriculares diferentes, incluir la disponibilidad y preferencia de los profesores al momento de asignar los ramos en periodos específicos y le da la posibilidad a los alumnos de elegir ciertos ramos.

3. Estado del Arte

3.1. Origen del problema

El BACP fue propuesto en 1990 por Hnich et al., [5] y publicado en CSPLib¹ Los primeros en abordar el problema fueron Castro y Manzano [1]. Ellos presentan dos métodos para resolver el problema, el primero consiste en un modelo de programación entera que resuelven utilizando `lp_solve`². El segundo en un modelo basado en restricciones utilizando heurísticas de orden de instanciación de variables y asignación de valores utilizando el lenguaje Oz³. Al ser los primeros en abordar el problema, hicieron público en CSPLib tres mallas curriculares las cuales utilizaron en su experimentación. `bacp8` una instancia de 8 periodos, `bacp10` una instancia de 10 periodos y `bacp12` una instancia de 12 periodos, las cuales corresponden a las tres carreras de informática que se imparten en la Universidad Técnica Federico Santa María. Estas instancias se convierten en las utilizadas por la mayoría de los investigadores posteriores para tener resultados comparables.

3.2. Primeros acercamientos: ILP y CP

Los primeros acercamientos realizados por Castro y Manzano, [1] se basaron en la comparación de programación entera en contraste con programación de restricciones utilizando heurísticas de orden de instanciación de variables y asignación de valores. Los tiempos de ejecución para los datasets entregados `bacp8`, `bacp10` y `bacp12` fueron los siguientes.

Los resultados se observan en la siguiente tabla muestran el tiempo que tomo a cada método en las distintas instancias de CSPLib, comparando el tiempo que tomo en encontrar la solución óptima.

¹CSPLib es una librería con múltiples problemas de prueba para solvers de satisfacción de restricciones.

²Solver para programación lineal entera mixta. <http://lpsolve.sourceforge.net/5.5/>

³Lenguaje de programación Mozart <http://mozart.github.io/>

		lp_solve	Oz			
		agrupado por ramo			agrupado por periodo	
			ingenuo	inverso	ingenuo	inverso
8 PERIODOS						
	Primera solución	1.7 [s]	0.1 [s]	0.1 [s]	0.2 [s]	0.1 [s]
	Mejor solución	1479.7 [s]	∞	0.1 [s]	∞	0.1 [s]
10 PERIODOS						
	Primera solución	5.9 [s]	0.2 [s]	0.1 [s]	0.2 [s]	0.1 [s]
	Mejor solución	∞	3.6 [s]	0.1 [s]	∞	0.1 [s]
12 PERIODOS						
	Primera solución	∞	0.4 [s]	0.5 [s]	∞	0.3 [s]
	Mejor solución	∞	∞	0.3 [s]	∞	0.3 [s]

Cuadro 1: Sumario comparativo del rendimiento entre lp_solve y Oz [1]

Notaron de inmediato que las heurísticas utilizadas en Oz eran muy superiores a los resultados que lograba lp_solve. Se observa que el modelo de programación entera no logró obtener resultados para el problema de 12 periodos. lp_solve no pudo encontrar el resultado óptimo, pues solo llegó a 24 créditos en 1626.84 segundos.

3.3. Modelos Híbridos

El año 2002 Hnich et al. [4] publican una nueva técnica para resolver el problema. Presentan un nuevo modelo de CP, pero no se quedan ahí, lo que hacen son experimentos híbridos mezclando las técnicas definidas en [1] con nuevos modelos de CP que definen nuevos espacios de búsqueda.

El nuevo modelo (CP_2) modifica la representación de los periodos en el que se le asigna al ramo i -ésimo un periodo específico utilizando un arreglo (1d), a diferencia del modelo presentado en [1] (CP_1) en que se representaba una matriz (2d) binaria. Lo que se logró fue un modelo mas eficiente al eliminar variables binarias necesarias en las restricciones dada la nueva representación de un arreglo unidimensional disminuyendo el espacio de búsqueda.

Las posibles mezclas de algoritmos híbridos incluyen: $ILP + CP_2$ y $CP_1 + CP_2$. La idea detrás de estos algoritmos híbridos era la de lograr reducir el espacio de búsqueda podando soluciones infactibles. Lo ideal es que el algoritmo híbrido utilice lo mejor de cada uno de los algoritmos que lo componen.

Los resultados que obtuvieron fueron los siguientes:

Modelo	8 periodos		10 periodos		12 periodos	
	tiempo	fallas	tiempo	fallas	tiempo	fallas
ILP	3.45	N/A	4.23	N/A	131.30	N/A
CP_1	58.52	499336	-	-	-	-
CP_2	45.10	56766	-	-	-	-
$ILP + CP_2$	0.81	183	8.44	4445	3.05	525
$CP_1 + CP_2$	0.29	651	0.59	1736	1.09	1539

Cuadro 2: Encontrando una solución óptima

El modelo mas rápido en encontrar una solución optima fue el híbrido entre CP_1 y CP_2 , este modelo logró reducir el espacio de búsqueda que compensó el incremento del uso de variables y restricciones debido al uso híbrido de ambos algoritmos.

Este trabajo dio pie a seguir investigando en la linea de modelos híbridos. Uno de ellos es el uso de Algoritmos Genéticos y Propagación de restricciones [6]. En esta investigación utilizan un framework genérico de hibridación. El framework desarrollado les permite especificar la proporción de los métodos, ajustando cuanto tiempo se asigna al GA o al CP. De esa forma pudieron realizar diferentes pruebas cambiando el tiempo que GA o CP tiene destinado en el algoritmo híbrido. De todas formas el principal aporte de esta investigación fue el framework utilizado que les permitió controlar la proporción entre los algoritmos que componen al híbrido.

Dentro de sus resultados en la investigación de GA + CP se observa que logra calcular el óptimo para las instancias bacp8, bacp10 y bacp12 en 15.05, 34.84 y 35.20 segundos respectivamente, mucho mas rápido de lo que logra `lp_solve`.

3.4. Función objetivo alternativa

El 2007 aparece un nuevo ángulo para atacar el problema consiste en minimizar la suma de las distancias medias de la carga de cada periodo y la carga promedio total, i.e.,

$$\min C(p) = \sum_{i=1}^m |m \alpha_i - w|^p$$

donde α_i es la carga del ramo i , y w es la carga total [7] que se obtiene como:

$$w_i = \sum_{i \in \text{ramos}} \alpha_i$$

En esta investigación comparan 4 casos particulares: $C(1)$, $C(2)$, $C(\infty)$ y el método clásico de [1] C^{max} . Junto con esta nueva función objetivo los autores introducen un método híbrido entre branch-and-bound y Tabu Search como método de búsqueda local. Esto permite encontrar soluciones factibles rápidamente pero no garantizan el óptimo. Además sugieren que dada la naturaleza de las instancias (bacp8, bacp10 y bacp12) las restricciones de máximos y mínimos para la carga por periodo y cantidad de ramos por periodo no es necesario considerarlas, pues las soluciones balanceadas de estas instancias siempre respetan dichas restricciones.

Monette et al., aportan con un generador de instancias, el cual utilizan para generar 100 nuevas instancias mas grandes y complejas que las bacp8, bacp10 y bacp12. En sus experimentos concluyen que el método híbrido de branch-and-bound y Tabu search es mas eficiente para el modelo C^{max} , ya que tabu search logra encontrar óptimos locales rápidamente, para que luego branch-and-bound pueda proseguir mejorando soluciones alternativas. Para el caso de minimizar $C(1)$ proponen el método iterativo como mejor opción. Es por esto que plantean como futuras líneas de investigación el uso de portafolio de diferentes métodos de búsqueda específicos para cada instancia del BACP.

Dentro de la experimentación que realizan con las instancias generadas muestran que las instancias con mayor cantidad de requisitos tienden a ser mas fáciles de responder, más aún, los problemas que tienen entre un 0 % y un 25 % de prerequisites tienden a ser difíciles. La dificultad disminuye considerablemente cuando se pasa el umbral del 25 % [7]. Esto se traduce en cantidad de restricciones, el espacio de búsqueda del problema inicialmente es muy grande y mientras mas prerequisites, mas restricciones aparecen. Es fácil entonces reducir drásticamente el espacio de búsqueda del problema y escapar del estado de transición⁴.

3.5. Búsquedas locales híbridas

El año 2008 Di Gaspero y Schaerf [3] en donde proponen el GBACP, una variantes más compleja del BACP que busca acercar el problema a la realidad, donde hay ciertos aspectos que BACP no considera, algunos de estos son: los estudiantes pueden elegir alternativas de ramos e.g., ramos de especialidad, las mallas curriculares pueden compartir ciertos ramos, e.g., matemáticas y físicas, los profesores pueden dar su disponibilidad y preferencia para dictar un ramo en periodos específicos.

Para resolver esta variante deciden utilizar métodos de búsqueda local (Tabu Search, Simulated Annealing, Steepest Descent, entre otros), los cuales aplican como primera instancia a los problemas de CSPLib (bacp8, bacp10 y bacp12), los resultados que obtienen son en general muy buenos, con una tasa de éxito de 66.23 % promedio para el bacp8. Los métodos Simulated Annealing, Tabu Search y Dynamic Tabu Search obtuvieron una tasa de exito de 100 % para el bacp8 y bacp10, por los que parecen ser los mas apropiados para abordar este problema.

3.6. Hormigas y rastros de feromonas

Uno de los trabajos mas reciente (2013) sobre el BACP utiliza metaheurísticas de Ant Colony Optimization (AOC) en específico el método BWAS (Best-Worst Ant System). Esta investigación muestra como se puede simular el comportamiento de hormigas para lograr satisfacer las restricciones del problema. Para esto se crea un algoritmo modificado de BWAS para actualizar y mutar rastros de feromonas, así como también reiniciar el proceso de búsqueda cuando este se estanca [8]. Fueron capaces de encontrar óptimos para los problemas bacp8, bacp10 y bacp12, luego de 100 iteraciones el algoritmo no logra encontrar mejores soluciones, pues ya logró el óptimo.

Este método resulto ser muy rápido para hallar las soluciones óptimas, tardando 1.25, 1.92 y 6.37 segundos para el bacp8, bacp10 y bacp12 respectivamente. En su estudio también aplican el BACP a el caso real de su propia casa de estudios, la Universidad Católica de Valparaíso y Universidad de Playa Ancha.

⁴El estado de transición es el rango en que un problema es mas difícil dada la cantidad de restricciones que el problema presenta.

3.7. Comparación de métodos

En la siguiente tabla se muestra un análisis comparativo entre el tiempo que tomo a cada método encontrar una solución óptima (o lo mas cercano)⁵.

	bacp8	bacp10	bacp12
CASTRO Y MANZANO (2001) [1]			
lp_solve	1459.73 [s]	1626.84 [s] (no óptimo)	∞
HINCH ET AL. (2002) [4]			
<i>ILP</i>	3.45 [s]	4.23 [s]	131.30 [s]
<i>CLP</i> ₁	58.52 [s]	∞	∞
<i>CLP</i> ₂	45.10 [s]	∞	∞
<i>ILP</i> + <i>CLP</i> ₂	0.81 [s]	8.44 [s]	3.05 [s]
<i>CP</i> ₁ + <i>CLP</i> ₂	0.29 [s]	0.59 [s]	1.09 [s]
LAMBERT ET AL. (2006) [6]			
<i>GA</i> + <i>CP</i>	15.05 [s]	34.84 [s]	35.20 [s]
DI GASPERO Y SCHAEF (2008) [3]			
Simulated Annealing	0.0042 [s]	0.0429 [s]	0.1764 [s]
Tabu Search	0.0023 [s]	0.0046 [s]	0.0459 [s]
Dynamic Tabu Search	0.0026 [s]	0.0060 [s]	0.0843 [s]
RUBIO ET AL. (2013) [8]			
Best Worst Ant System	1.25 [s]	1.25 [s]	6.37 [s]

Cuadro 3: Comparación de resultados entre todos los métodos estudiados

Es claro entonces que los mejores resultados son los logrados por Di Gaspero y Schaefer [3] utilizando métodos de búsqueda local híbridos. Muy de cerca están los resultados de Hinch et al. Pareciera entonces que los métodos híbridos son el método preferido para encontrar soluciones al BACP.

4. Modelo Matemático

4.1. Representación en matriz binaria

El primer modelo clásico es el presentado en [1], en su representación utiliza una matriz binaria para establecer x_{ij} si un ramo i pertenece a un periodo j . De esta manera se puede obtener la carga total de el periodo k -ésimo como la suma de la columna k de la matriz de pertenencia.

⁵Es necesario destacar que hay una diferencia de hasta 12 años entre el primer método y el mas reciente estudiado en este informe, por lo que parte de la mejora en el rendimiento puede deberse a la evolución del poder de computación en el tiempo.

Parámetros

m: número de ramos

n: número de periodos académicos

α_i : número de créditos del ramo i $\forall i = 1..m$

β : carga académica mínima permitida por periodo

γ : carga académica máxima permitida por periodo

δ : cantidad mínima de ramos por periodo

ϵ : cantidad máxima de ramos por periodo

Variables

$$x_{ij} = \begin{cases} 1 & \text{si ramo } i \text{ es asignado al periodo } j \\ 0 & \text{en otro caso} \end{cases} \quad \forall i = 1..m, \forall j = 1..n$$

Función Objetivo

$$\text{Min } c = \text{Max}\{c_1, \dots, c_n\}$$

Restricciones

Carga académica del periodo j

$$c_j = \sum_{i=1}^m \alpha_i x_{ij} \quad \forall j = 1 \dots n$$

Todos los ramos i deben ser asignados a algún periodo j

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i = 1 \dots m$$

El ramo b tiene como prerequisite al a

$$x_{bj} \leq \sum_{r=1}^{j-1} x_{ar} = 1 \quad \forall j = 2 \dots n$$

La carga académica máxima esta definida como

$$\begin{aligned} c &= \text{Max} \{c_1, \dots, c_n\} \\ c_j &\leq c \quad \forall j = 1 \dots n \end{aligned}$$

La carga académica de un periodo no debe pasar los límites

$$\beta \leq c_j \leq \gamma \quad \forall j = 1 \dots n$$

La cantidad de ramos de un periodo no debe pasar los límites

$$\delta \leq \sum_{i=1}^m x_{ij} \leq \epsilon \quad \forall j = 1 \dots n$$

El espacio de búsqueda para este modelo es una matriz, en que cada elemento (i, j) puede tomar dos valores posibles. Si el tamaño de la matriz es de $m \times n$ entonces el espacio de búsqueda es de: $2^{m \times n}$. Para los casos particulares de bacp8, bacp10 y bacp12 se tienen los siguientes espacios de búsqueda:

Instancia	n	m	Espacio de búsqueda
bacp8	8	46	2^{368}
bacp10	10	42	2^{420}
bacp12	12	66	2^{792}

Cuadro 4: Espacio de búsqueda para el modelo clásico

4.2. Representación en un arreglo

Esta representación disminuye considerablemente el espacio de búsqueda, ya que utiliza solo un arreglo unidimensional para la asignación de ramos a los periodos. Si bien reduce el espacio de búsqueda, la restricción de la carga académica del periodo j es mas difícil de calcular [4]. El modelo es el siguiente:

Parámetros

- m:** número de ramos
- n:** número de periodos académicos
- α_i :** número de créditos del ramo i $\forall i = 1..m$
- β :** carga académica mínima permitida por periodo
- γ :** carga académica máxima permitida por periodo
- δ :** cantidad mínima de ramos por periodo
- ϵ :** cantidad máxima de ramos por periodo

Variables

- x_i :** periodo del ramo i $\forall i = 1 \dots m, x_i \in \{1 \dots n\}$

Función Objetivo

$$\text{Min } c = \text{Max}\{c_1, \dots, c_n\}$$

Restricciones

Carga académica del periodo j

$$c_j = \sum_{i=1}^m \alpha_i \mid x_i = j \quad \forall j = 1 \dots n$$

El ramo b tiene como requisito al a

$$x_a < x_b$$

La carga académica máxima esta definida como

$$c = \text{Max} \{c_1, \dots, c_n\}$$

$$c_j \leq c \quad \forall j = 1 \dots n$$

La carga académica de un periodo no debe pasar los límites

$$\beta \leq c_j \leq \gamma \quad \forall j = 1 \dots n$$

La cantidad de ramos de un periodo no debe pasar los límites

$$\delta \leq \sum_{i=1}^m 1 \mid x_i = j \leq \epsilon \quad \forall j = 1 \dots n$$

La única diferencia es la variable que se utiliza, esto modifica el espacio de búsqueda. existirán m variables, cada una podrá tomar n valores, por lo que el espacio de búsqueda es de n^m

Instancia	n	m	Espacio de búsqueda
bacp8	8	46	8^{46}
bacp10	10	42	10^{42}
bacp12	12	66	12^{66}

Cuadro 5: Espacio de búsqueda para el modelo de un arreglo 2d

En todos los casos, el espacio de búsqueda es menor que en el modelo clásico.

4.3. Minimizar distancias

Este modelo cambia la función objetivo, en este caso se minimiza la suma de las distancias medias entre la carga del periodo y la carga promedio total [7]. En este modelo deciden eliminar las restricciones de número de ramos máximo y mínimo por periodo, así como también la carga máxima y mínima por periodo argumentando que la naturaleza de las soluciones balanceadas de los problemas (bacp8, bacp10 y bacp12) siempre cumplen con dichas restricciones.

Parámetros

m: número de ramos

n: número de periodos académicos

α_i : número de créditos del ramo $i \quad \forall i = 1..m$

w : carga total $w = \sum_{i=1}^m \alpha_i$

Variables

x_i : periodo del ramo $i \quad \forall i = 1 \dots m, x_i \in \{1 \dots n\}$

Función Objetivo

$$\text{Min } c(p) = \sum_{i=1}^m |m \alpha_i - w|^p \quad p > 0$$

Restricciones

Carga académica del periodo j

$$c_j = \sum_{i=1}^m \alpha_i \mid x_i = j \quad \forall j = 1 \dots n$$

El ramo b tiene como prerequisite al a

$$x_a < x_b$$

En este modelo, se busca minimizar una función de distancia media, esta distancia viene dada por la expresión $|m \alpha_i - w|^p$ que depende de p y representa la distancia de la carga de el periodo i con la carga promedio. En el modelo no existe una preferencia a algún p [7]. En sus estudios comparan para los casos de $p = 1$, $p = 2$ y $p = \infty$ pero no encuentran una relación directa entre p y los resultados obtenidos.

El espacio de búsqueda es el mismo que en el modelo anterior. En la investigación [7] muestran ambas representaciones de variables pero no hacen mención al espacio de búsqueda.

5. Representación

Representación matemática y estructura de datos que se usa (arreglos, matrices, etc.), por qué se usa, la relación entre la representación matemática y la estructura. **Utilizar figuras, tablas, etc. en lo posible, de modo que sea fácil entender las estructuras utilizadas, además describa lo más sencillo y claro que pueda.**

6. Descripción del algoritmo

Cómo fue implementando, interesa la implementación más que el algoritmo genérico, es decir, si se tiene que implementar SA, lo que se espera es que se explique en pseudo código la estructura general y en párrafo explicativo cada parte cómo fue implementada para su caso particular, si se utilizan operadores se debe explicar por qué se utilizó ese operador, si fuera el caso de una técnica completa, si se utiliza recursión o no, etc. **Recuerde utilizar pseudocódigo para mostrar cada algoritmo y separar la explicación de su algoritmo en secciones para que se logre un mejor entendimiento. Además, en este punto no se espera que se incluya código, eso va aparte.**

7. Experimentos

Se necesita saber cómo experimentaron, cómo definieron parámetros, cómo se fueron modificando, cuáles problemas se trataron, cuáles fueron las instancias utilizadas, cuáles fueron las características de las instancias, cuáles fueron los criterios de término del algoritmo. **En el caso de las técnicas incompletas, es importante que ejecute su programa varias veces con distintas semillas aleatorias, para obtener valores estadísticos de los resultados.**

8. Resultados

Que fue lo que se logró con la experimentación, incluir tablas y parámetros, gráficos, etc. lo más explicativo posible. Además destacar puntos importantes del problema, características de las instancias que influyeron en los resultados, valores de parámetros que influyeron en los resultados, análisis de calidad de las soluciones encontradas a través del tiempo, análisis en profundidad de la técnica vs los resultados obtenidos, valores promedio, desviaciones, **comparaciones con resultados de la literatura**, etc. También debería discutir qué cosas podría haber agregado o quedan como desafíos para trabajo futuro en su algoritmo.

9. Conclusiones

Conclusiones RELEVANTES del estudio realizado. Debería contener conclusiones del informe 1 junto con conclusiones obtenidas del desarrollo del software, análisis de resultados de experimentos y trabajo futuro.

10. Bibliografía

Indicando toda la información necesaria de acuerdo al tipo de documento revisado. Las referencias deben ser citadas en el documento.

Referencias

- [1] Carlos Castro and Sebastian Manzano. Variable and value ordering when solving balanced academic curriculum problems. *CoRR*, cs.PL/0110007, 2001.

- [2] Marco Chiarandini, Luca Di Gaspero, Stefano Gualandi, and Andrea Schaerf. The balanced academic curriculum problem revisited. *Journal of Heuristics*, 18(1):119–148, 2012.
- [3] Luca Di Gaspero and Andrea Schaerf. Hybrid local search techniques for the generalized balanced academic curriculum problem. In MarÅaJ. Blesa, Christian Blum, Carlos Cotta, AntonioJ. FernÃndez, JosÃ©E. Gallardo, Andrea Roli, and Michael Sampels, editors, *Hybrid Metaheuristics*, volume 5296 of *Lecture Notes in Computer Science*, pages 146–157. Springer Berlin Heidelberg, 2008.
- [4] Brahim Hnich and Zeynep Kiziltan. Modelling a balanced academic curriculum problem. In *Proceedings of CP-AI-OR-2002*, pages 121–131, 2002.
- [5] Brahim Hnich, Zeynep Kiziltan, and Toby Walsh. CSPLib problem 030: Balanced academic curriculum problem (bacp). <http://www.csplib.org/Problems/prob030>.
- [6] T. Lambert, C. Castro, E. Monfroy, and F. Saubion. Solving the balanced academic curriculum problem with an hybridization of genetic algorithm and constraint propagation. In Leszek Rutkowski, Ryszard Tadeusiewicz, LotfiA. Zadeh, and JacekM. Åurada, editors, *Artificial Intelligence and Soft Computing â ICAISC 2006*, volume 4029 of *Lecture Notes in Computer Science*, pages 410–419. Springer Berlin Heidelberg, 2006.
- [7] Jean noÃl Monette, Pierre Schaus, StÃ©phane Zampelli, Yves Deville, and Pierre Dupont. A cp approach to the balanced academic curriculum problem, 2007.
- [8] JosÃ©-Miguel Rubio, Wenceslao Palma, Nibaldo Rodriguez, Ricardo Soto, Broderick Crawford, Fernando Paredes, and Guillermo Cabrera. Solving the balanced academic curriculum problem using the aco metaheuristic. *Mathematical Problems in Engineering*, 2013:Article ID 793671, 8 p., 2013.