

UNA SOLUCIÓN BASADA EN AGENTES AL PROBLEMA DE GENERACIÓN DE HORARIOS

An Agent-Based Solution to the Timetable Problem

ORLANDO LÓPEZ-CRUZ*

Recibido: 25 de marzo de 2015. Aceptado: 18 de Abril de 2015

RESUMEN

La generación de horarios de clase es una actividad académico-administrativa inevitable en las organizaciones educativas. Además de su carácter periódico, es de alta complejidad computacional: se trata de un problema NP-completo. Las interacciones típicas en una organización educativa tienden a mantener la solución del problema en ese orden de complejidad computacional. Se ha acostumbrado que quien controla los recursos de infraestructura, en una orientación jerárquica, gobierna la generación de horarios. No obstante, en un contexto en el que los proveedores de infraestructura –la organización educativa–, los profesores y los estudiantes actúan como pares, los horarios de clase pueden emerger de las interacciones. Suponiendo que la organización educativa está al mismo nivel que cada docente y que cada estudiante, se tiene un entorno apropiado para un sistema multiagente. Este artículo da cuenta de los resultados de pruebas realizadas a una aplicación basada en agentes para la resolución del problema de generación de horarios.

Palabras clave: sistemas basados en agentes, agente, complejidad computacional, administración de recursos, problema de generación de horarios.

ABSTRACT

Course timetabling generation is an unavoidable academic-administrative activity in educational organizations. Besides its periodicity, it is a high computational complex NP-Complete problem. Typical interactions in an educational organization cause the solution to remain in that complexity order. It is used that those whom hierarchical control resources, governs course timetabling generation. Nevertheless, when infrastructure providers –the educational organization–, teachers and students act as peers, course timetabling may emerge from interactions. When assuming that an educational organization is at the same level that teachers and students, there is an appropriate environment for an agent-based system. This paper shed light on results from tests performed on an agent-based application to solve the timetable problem.

Keywords: agent-based systems, computational complexity, resource management, timetable problem.

* Miembro del grupo de investigación Riesgo en Sistemas Naturales y Antrópicos de la Pontificia Universidad Javeriana. Doctor (c) en Ingeniería, Pontificia Universidad Javeriana, Bogotá D.C., Magister en Administración, Universidad Nacional de Colombia, Especialista en Sistemas de Control Organizacional, Universidad de los Andes, Economista. Docente Universidad El Bosque. Correo electrónico: orlandolopez@uelbosque.edu.co

I. INTRODUCCIÓN

Las organizaciones de educación, especialmente las instituciones de educación superior como institución social que forma fuerza de trabajo altamente calificada e investigadores académicos, pueden ser caracterizadas por la concurrencia de docentes y estudiantes que comparten la actividad de aprendizaje. La coordinación de tal actividad exige un esfuerzo para hacer converger docentes y estudiantes en un mismo entorno materializado en los horarios de clase en los que se asocian estudiantes, docentes e infraestructura, para representar los espacios académicos.

Verificar la coordinación de acciones académicas, curriculares y extracurriculares, de estudiantes, docentes, cursos, tiempos, infraestructura (sedes, aulas con distintas características, laboratorios de experimentación, lugares de práctica clínica, consultorios, salas de cómputo, entre otros) es un problema de optimización de complejidad computacional no polinómica duro (NP-hard), dentro del conjunto de problemas del paradigma del «problema del agente viajero» (TSP) [1].

La creación o generación de horarios de clase ha sido un problema permanentemente abordado en distintos tiempos y distintas ópticas [2-5]. En Colombia se informa la utilización de técnicas de computación evolutiva y bioinspirada [6]-[7] para resolverlo.

Problemas de este tipo, en los que se exige mejorar capacidad de razonamiento enfrentan la disyuntiva entre la velocidad de obtención de la solución o su equivalente medido en tiempo de procesamiento, frente a la calidad de la solución, han sido abordados desde la inteligencia artificial [8]. En las últimas décadas del siglo XX se realizó de forma que la generación de horarios se efectuara de forma «automática» [9]-[10] mediante máquinas que simulan el pensamiento humano.

No obstante, las soluciones propuestas al problema de asignación de horarios de clase típicamente están orientadas hacia la perspectiva de la parte administrativa de la universidad, como controlador de la infraestructura, sin considerar a los directamente interesados: docentes y estudiantes, desde la «parte» administrativa de la Universidad, generando en más de una ocasión ineficiencias y

malestares. Una de las posibles explicaciones a esta situación procede del hecho de que la planeación de horarios de clase ocurre previamente a la inscripción oficial de estudiantes (mediada por el pago de matrícula). Esto hace que el controlador de la infraestructura, sin intervención de estudiantes y algunas veces sin docentes, prepare una oferta de horarios. Los docentes, pueden informar previamente su disponibilidad para el periodo, pero la anticipación puede ser excesiva al punto que la planeación resulte poco realista por situaciones sobrevinientes como cambio de docente o inevitable cambio en la planeación de actividades del docente.

La información de inscripción previa de estudiantes, ha estado orientada a ser una fuente del planificador, usualmente el propietario de la infraestructura, para estimar la posible demanda por cursos y demanda de uso de la infraestructura misma, pero no como instrumento para poner al estudiante y al docente en situación par con los demás agentes. Además, los cambios originados en la disponibilidad de los docentes, puede llegar a generar un esfuerzo inmenso, cuando no es que resulta imposible el cambio. Una forma en la que se puede intentar reducir la complejidad computacional de la asignación de horarios es haciendo concurrir a los agentes en el mismo momento del tiempo, de forma que puedan hacer emerger un acuerdo inmediato que favorezca a los agentes.

Este trabajo presenta los resultados de una solución por computador con un sistema basado en agentes, en la que los estudiantes actúan como agentes colaborativos en cuya interacción emerge un horario de clases. La sección II presenta los resultados de las etapas de análisis y diseño apoyadas con la herramienta CASE (Computer-Aided Software Engineering). La sección III muestra resultados obtenidos con el producto desarrollado y la sección IV los compara con otras aproximaciones a la solución del problema mediante distintas tecnologías. Finalmente, en la sección V, se presentan conclusiones.

II. MATERIALES Y MÉTODO

Considerando que la complejidad en la construcción de sistemas inteligentes crece permanentemente [11], una perspectiva integradora, como

la teoría general de sistemas debería resultar evidente no solo porque «la tecnología ha acabado pensando no ya en términos de máquinas sueltas sino de sistemas» [12], sino porque no se dan las dos condiciones para la aplicación del principio analítico: i) No existencia de interacciones entre partes y ii) linealidad. Este principio epistemológico sistémico valida la aplicación de un sistema basado en agentes para verificar organizaciones emergentes [13], como «arreglo de relaciones entre componentes o individuos que producen una unidad, o sistema, dotada con cualidades que no aparecen en el nivel de los componentes o individuos» [14].

Entendiendo el concepto de complejidad como «la propiedad de un sistema de ser capaz de adoptar un gran número de estados o comportamientos» [15], y la variedad como «el número de estados posibles de cualquier situación cuya complejidad queremos medir» [15], entonces al aplicar la ley de la variedad requerida según la cual «sólo la variedad puede absorber variedad» [15] a los sistemas basados en agentes, la medida de la complejidad, llamada variedad, de una «totalidad» es mayor que la variedad de los subsistemas que la integran. Dicho exceso de complejidad procede de las propiedades emergentes de la «totalidad». Así que si un sistema basado en agentes modela dicha «totalidad», para ser un modelo mínimamente aceptable, debe incluir algunas de las relaciones emergentes que caracterizan la «totalidad» modelada, asunto que se logra precisamente al incluir dentro del modelo las interacciones entre los agentes que conforman una organización, en la que no solamente concurren seres humanos sino otros recursos [16], como es el caso de una organización educativa.

Los sistemas basados en agentes [17]-[18] ocupan un lugar especial en la evolución de metodologías de Ingeniería de Software, que permiten la programación orientada a agentes (POA) [19]. Se reporta la existencia de distintas metodologías de POA como Gaia, Tropos, MAS-CommonKADs, MaSE, Archon y Awic entre otras [19]. La metodología AOPOA (Aproximación Organizacional para la Programación Orientada a Agentes) [20] cuyo proceso se desarrolla en las etapas de análisis, diseño y despliegue, permitió el desarrollo del artefacto que se ejecutó y del que obtuvieron los resultados.

Esta metodología dispone de una herramienta CASE (*Computer Aided Software Engineering*) que asiste al diseñador y que se denomina AOPOA CASE Tool V1.1 [20] que apoya el proceso de diseño del sistema y genera un conjunto de artefactos, una estructura final de programación que permite la implementación y ejecución. La herramienta fue instalada en un computador personal con disco duro de 500 Gbytes, procesador Intel Centrino duo T5250 a 1.5 GHz y 2 Gbytes de memoria física (RAM), sistema operacional MS Windows 7.

Para las pruebas realizadas se dispuso de un entorno de red de dicho computador y una máquina virtual con Windows XP como sistema operativo.

A. Fase de análisis

De acuerdo con la metodología de diseño [20], antes de iniciar la etapa de análisis debe haberse determinado que el problema es adecuado para ser resuelto con una solución basada en agentes. La asignación de horarios de clase, por tratarse de un problema complejo y de acuerdo con la aproximación en la que todos los agentes del sistema concurren en el tiempo, aunque estén geográficamente dispersos, puede considerarse apropiado para ser resuelto mediante esta metodología.

1) *Descripción del problema*: El acto que vincula a un estudiante a un programa académico de una universidad es el pago de matrícula. Este hecho puede ser enriquecido si se entienden más que el hecho de emisión de una cuenta de cobro al cliente estudiante (como cliente), sino como una oportunidad para conocer las circunstancias como el cliente (el estudiante y en programas de pregrado usualmente su familia) está en disposición de apropiarse de los servicios educativos. Desde la contemporánea perspectiva de la gestión organizacional empresarial, es uno de los puntos importantes de la gestión de relación con el cliente (*customer relationship management*). Para la solución planteada, se supone que la matrícula se ha efectuado.

Un estudiante se vincula por lo menos a un plan de estudios de una universidad. El vínculo se materializa con la individualización de su horario de clases de acuerdo con la oferta de cursos de la universidad. Por otra parte, los docentes materializan

la prestación del servicio educativo por mantener una relación directa con el estudiante.

El controlador de la infraestructura se identifica como la Administración Académica que, como órgano administrativo, formula los planes de estudio, respalda operativamente las actividades académicas realizadas por los docentes y son la fuente de datos que permiten establecer una asociación entre docentes y cursos que pueden dictar los docentes, de acuerdo con su formación, especialidad y disponibilidad.

Como la oportuna disponibilidad de la infraestructura adecuada constituye un factor crítico de éxito en la prestación de un servicio que el cliente perciba como de alta calidad, que permite realizar las clases en un lapso que se ubica dentro de un periodo fijo de tiempo (usualmente una semana). El manejo de dichos ítems dependerá de distintas variables como la ubicación de los alumnos, de los profesores, de la planta física de la misma universidad, de los cupos de cada curso a dictar y de la disponibilidad de cada profesor.

Dado el contexto problemático descrito antes, una perspectiva en la que cada uno de los actores en el proceso de «planeación educativa» participa a través de medios computacionales que disponen de una aplicación basada en agentes puede vincular al proceso otros aspectos del entorno universitario y, además, disminuir o, al menos, controlar la complejidad computacional de la coordinación académica incluyendo los procedimientos administrativos de ingreso de estudiantes, selección de cursos y distribución de los mismos en una universidad.

También, debería considerarse la posibilidad de que la Universidad disponga de distintas sedes o edificios distantes entre sí, por lo cual la preparación de un horario debería incluir los tiempos de traslado de los docentes y estudiantes. La Fig. 1 representa esta posibilidad, permitiendo considerar el manejo adecuado de las asignaturas dependiendo no solo de las especificaciones de los estudiantes, sino también de la distribución que la universidad tenga de sus asignaturas en la planta física.

La descomposición recursiva de la organización permite la identificación de roles que, en la meto-

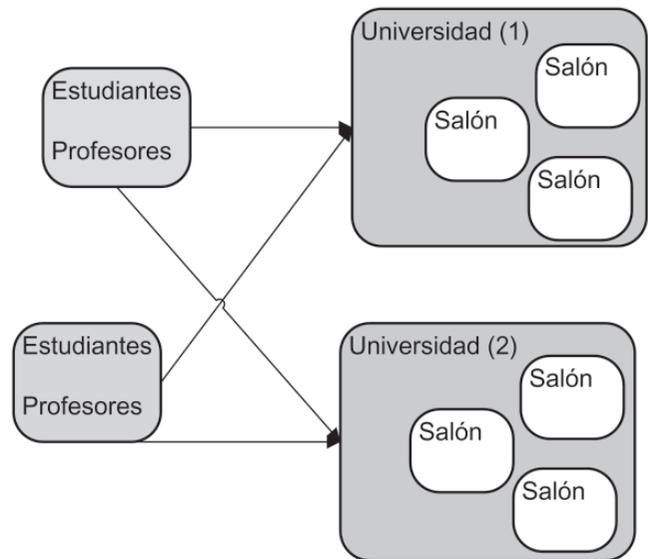


Fig. 1. Entorno general del problema. Los estudiantes y docentes concurren al uso de salones y otra infraestructura que es administrada por un tercer «agente» que materializa la administración de la Universidad.

dología, representan una entidad abstracta encargada del logro de un conjunto de objetivos, aprovechando sus habilidades y usando los recursos disponibles en su entorno.

Desde la perspectiva del estudiante como cliente de los servicios académicos de la Universidad, el sistema debería proveer información sobre los cursos disponibles y permitirle inscribir los cursos que sean de su mejor conveniencia, dentro del plan de estudios. Para esto debe facilitar la inscripción de cursos a los estudiantes, tomando en cuenta las siguientes cuatro restricciones: Cupos, Ubicación, Horario y Disponibilidad de la planta física.

Un sistema basado en agentes ayuda a que las tareas antes mencionadas se manejen de forma real y eventualmente permitan que la interacción con sujetos humanos induzca una reducción de la complejidad computacional.

2) *Objetivos*: De acuerdo con la metodología, en la herramienta CASE se integraron las clases de objetivos y los objetivos generados de dichas clases así:

- Clase 1: Generar horarios
 - Generar horarios de estudiantes en forma dinámica.

- Buscar colaboración para problemas de horarios.
- Asignar inscripción.
- Clase 2: Generar alertas
 - Notificar inscripciones.
- Clase 3: Analizar datos
 - Verificar cruces de horarios.

Dichas clases se identificaron de acuerdo con las necesidades identificadas para especificar los objetivos del sistema y cada una generaliza sus respectivos objetivos.

De acuerdo con estos objetivos, se desarrollaron los respectivos casos de uso, se definieron los recursos necesarios, las tareas a realizar y las habilidades con las que deben contar para desarrollar cada uno.

Cada caso de uso se presenta en forma independiente.

El caso de uso de la generación de horarios (Fig. 2), diagrama las relaciones entre los recursos «información de asignaturas» y «las preferencias de los estudiantes» con las acciones que el sistema debe hacer para poder generar los horarios.

El caso de uso de la generación de alertas de inscripción (Fig. 3), especifica las relaciones existentes para poder generar las alertas de cada inscripción de manera que cada estudiante y el sistema estén al tanto de los cambios en cada asignatura.

Por otra parte, el caso de uso correspondiente a ‘verificar restricciones’ (Fig. 4), diagrama la forma como el sistema debe verificar la información de las asignaturas, de los profesores y de los estudiantes para poder encontrar posibles complicaciones en alguna preferencia de estudiante o asignatura.

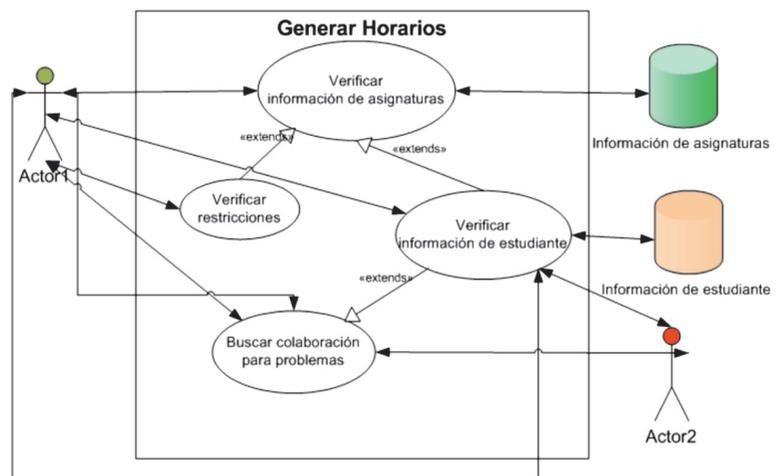


Fig. 2. Caso de uso Generar Horario.

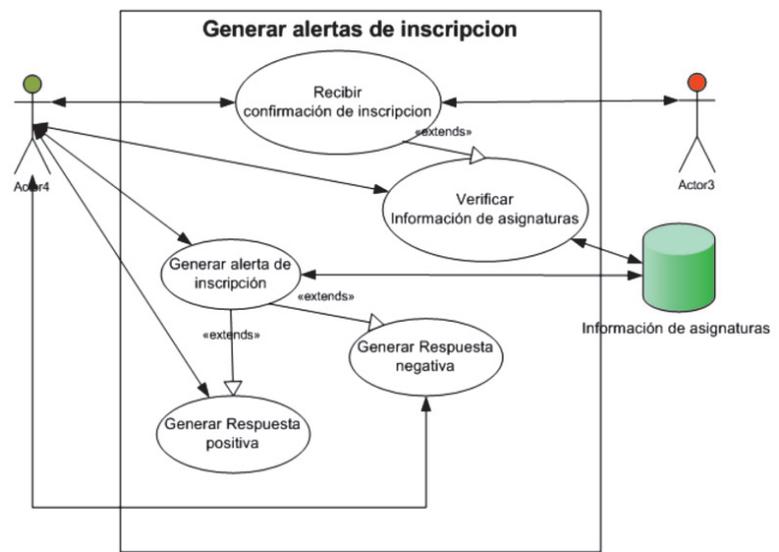


Fig. 3. Caso de uso Generar Alertas de Inscripción.

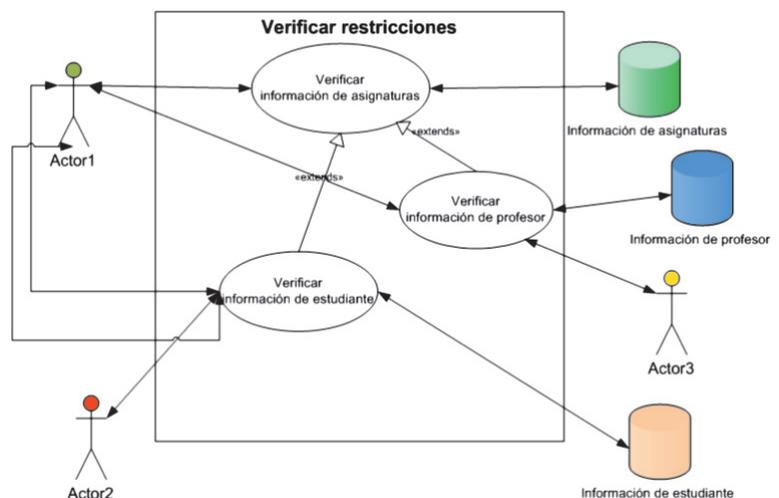


Fig. 4. Caso de uso verificar restricciones.

Con los objetivos ya especificados, se pasa a identificar los recursos necesarios. El sistema debe contar con los recursos identificados en la Tabla I para cumplir con sus metas.

De acuerdo con esto, se especificaron las habilidades para el manejo de los objetivos, de manera que las necesidades encontradas pudieran ser atendidas. Estas son:

- Leer información de asignaturas.
- Leer preferencias de estudiante.
- Analizar conflictos de cruce de horario.
- Enviar alertas de horario.
- Consultar disponibilidad de cupos.
- Consultar disponibilidad de profesores.
- Asignar inscripción a asignatura.
- Asignar cupos.

Dichas habilidades y recursos son asociados en las tareas que especifican y caracterizan cada objetivo con el fin de definir todo lo necesario para alcanzarlos. Las tareas del sistema son las relaciones entre objetivos y habilidades y entre objetivos y recursos y se desarrollan con el fin de encontrar correlación entre objetivos y recursos (Tabla II).

Con base en lo anterior, la herramienta CASE generó tres (3) clases de roles con sus respectivos objetivos (Tabla III). Cada rol resulta de agrupar tanto los recursos como las habilidades necesarias para alcanzar un objetivo según las relaciones dadas. Los artefactos que no generó la herramienta CASE se desarrollaron en forma independiente y se presentan a continuación en el documento.

Tabla I. Recursos del sistema.

Recurso	Tipo	Fuente
Número de asignaturas	Determinado	Rol
Información de entrada de asignaturas	Determinado	Rol
Información de disponibilidad de docentes	Determinado	Actor (Profesor)
Número de docentes	Determinado	Rol
Preferencias de estudiantes	Determinado	Actor (Estudiante)
Información dinámica de horarios establecidos	Infinito	Rol
Información dinámica de asignaturas	Infinito	Rol

Tabla II. Relación entre objetivos y habilidades del sistema.

Objetivo	Habilidad
Generar horarios de estudiantes en forma dinámica.	Leer información de asignaturas.
Generar horarios de estudiantes en forma dinámica.	Leer preferencias de estudiante.
Buscar colaboración para problemas de horarios.	Leer información de asignaturas.
Buscar colaboración para problemas de horarios.	Leer preferencias de estudiante.
Buscar colaboración para problemas de horarios.	Analizar conflictos de cruce de horario.
Asignar inscripción.	Asignar inscripción a asignatura.
Notificar inscripciones.	Leer información de asignaturas.
Verificar cruces de horarios.	Analizar conflictos de cruce de horario.

Objetivo	Recurso
Generar horarios de estudiantes en forma dinámica.	Número de asignaturas.
Generar horarios de estudiantes en forma dinámica.	Información de entrada de asignaturas.
Generar horarios de estudiantes en forma dinámica.	Preferencias de estudiantes.
Buscar colaboración para problemas de horarios.	Preferencias de estudiantes.
Asignar inscripción.	Información dinámica de asignaturas.
Notificar inscripciones.	Información dinámica de asignaturas.
Verificar cruces de horarios.	Preferencias de estudiantes.

Tabla III. Roles generados por la herramienta case y sus respectivos objetivos.

Rol	Objetivos
Rol 0	<ul style="list-style-type: none"> • Generar horarios de estudiantes en forma interactiva. • Asignar inscripción.
Rol 1	<ul style="list-style-type: none"> • Notificar inscripciones.
Rol 2	<ul style="list-style-type: none"> • Verificar cruces de horarios. • Buscar colaboración para problemas de horario.

Con la especificación de cada una de las partes intervinientes en el sistema, se procedió a especificar los vínculos de cooperación y participación de recursos para cada objetivo.

La metodología permite aprovechar la característica implícita de las interacciones consistente en que los vínculos de cooperación emergen cuando hay un objetivo compartido por dos o más roles o cuando hay recursos en común.

Para el objetivo «Generar horarios de estudiantes de forma interactiva» (Fig. 5) se graficaron las relaciones de dicho objetivo con los recursos necesarios y las relaciones entre los roles identificados para lograr el objetivo. El propósito de estos vínculos entre roles es encontrar colaboración entre estos según los recursos necesarios, para lo cual se adoptó el método de la subasta inglesa con algunas variaciones, como se presenta en la fase de diseño.

Para el objetivo «Buscar colaboración para problemas de horario» (Fig. 6), se graficaron las relaciones de dicho objetivo con los recursos necesarios y las relaciones entre los roles identificados para lograr el objetivo. El objetivo de estos vínculos entre roles es encontrar colaboración entre estos según los recursos necesarios, por lo que se definió utilizar el concepto de «vecindad» [21] para poder propagar la información y así lograr colaboración para resolver conflictos de cruces de horario.

Para los objetivos «Asignar inscripción» (Fig.7), «Notificar Inscripción» (Fig. 8) y «Verificar cruces de horario» (Fig. 9), se graficaron las relaciones del respectivo objetivo con los recursos necesarios

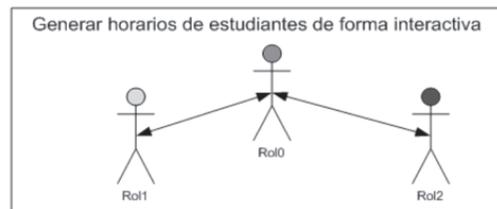
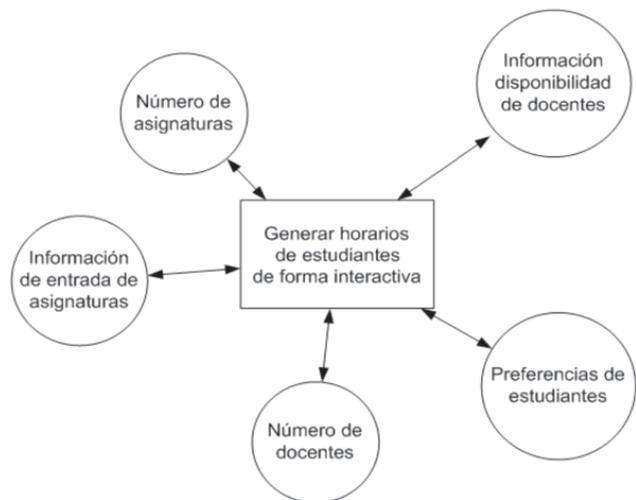


Fig. 5. Diagrama de vínculos y relación de recursos para generación de horarios.

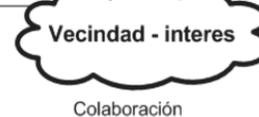
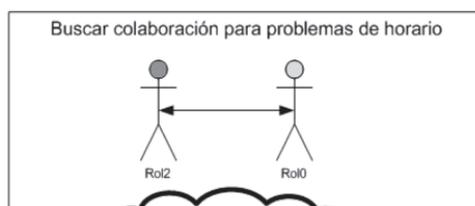
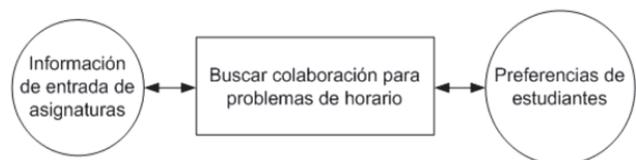


Fig. 6. Diagrama de vínculos y relación de recursos para buscar colaboración para problemas de horario.

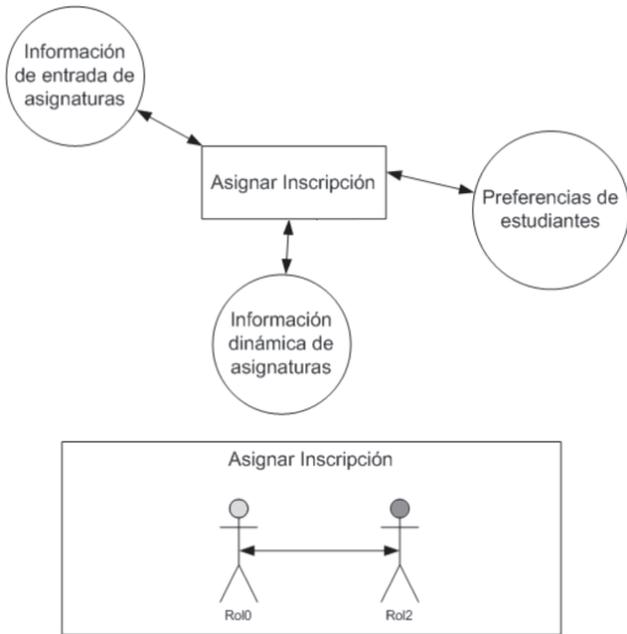


Fig. 7. Diagrama de vínculos y relación de recursos para asignar la inscripción.

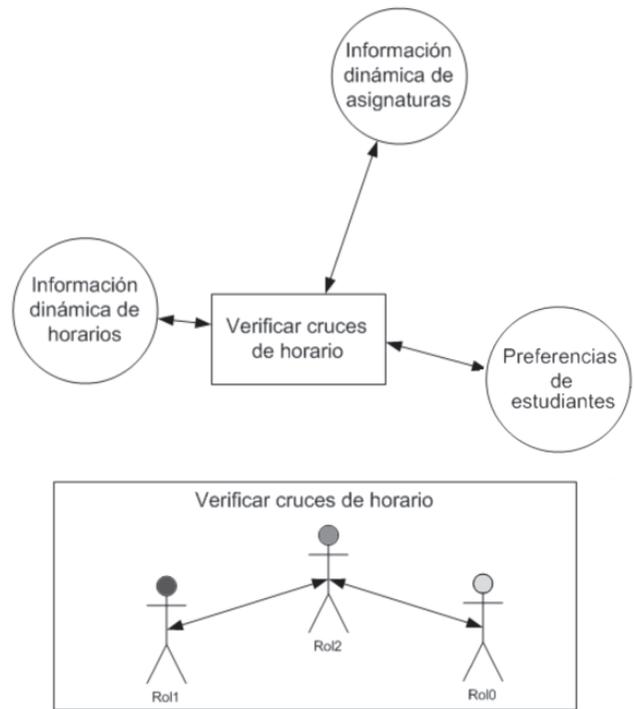


Fig. 9. Diagrama de vínculos y relación de recursos para verificar de cruces de horario.

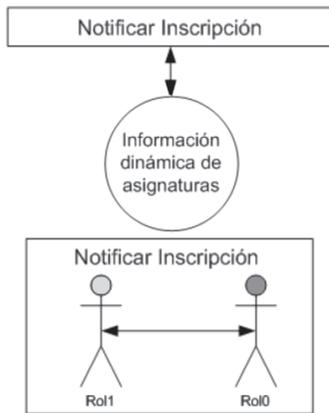


Fig. 8. Diagrama de vínculos y relación de recursos para notificación de inscripción.

y las relaciones entre los roles identificados para lograr cada objetivo. Para estos objetivos no se vio la necesidad de especificar algún método de colaboración, por lo que se puede alcanzar el objetivo con interacciones simples entre agentes. De esta manera se completa la elaboración de un modelo de comportamiento del sistema.

B. Fase de diseño

En la etapa de diseño se realizan los diagramas de vínculos de cooperación y el proceso de diseño de agentes.

Mediante el diagrama de vínculos de cooperación (Fig. 10) se especificaron las relaciones entre los diferentes roles con respecto a los

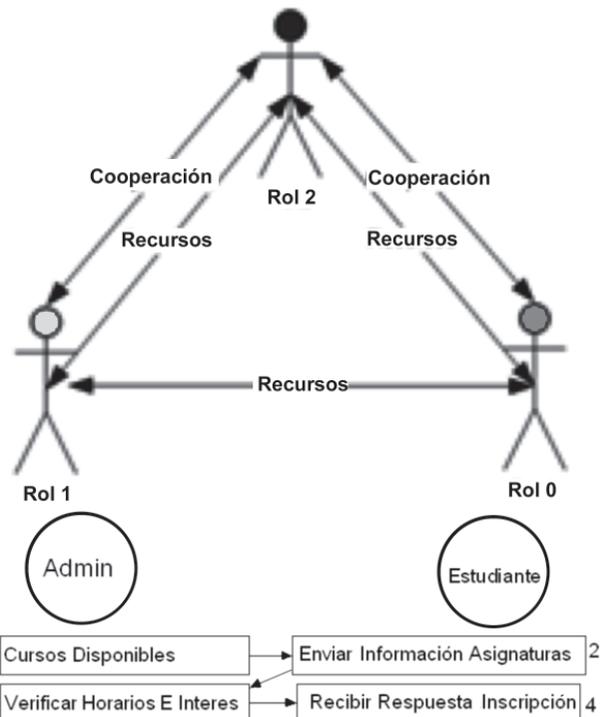


Fig. 10. Diagrama de vínculos de cooperación.

recursos y a la cooperación de acuerdo con lo encontrado en la fase de análisis y en los diagramas de vínculos de cada objetivo, en donde se especifican los métodos de colaboración a desarrollar en los casos necesarios.

Luego, se especificaron los eventos y respuestas de cada uno de ellos en relación con cada rol definido (Tabla IV).

Con estas definiciones, se pasó al diseño de agentes. Para los agentes, se definieron los objetivos

asociados a cada uno y a las relaciones entre estos. Por esta razón, se especificaron dos agentes para el sistema, el agente Admin (Administrador académico) y Sistema Información.

El agente *Admin* toma los roles 0 y 2 especificados (Tabla III), y el agente *Sistema Información* toma el rol 1.

Para cada uno de ellos se declararon las variables más significantes para el estado de los agentes, así:

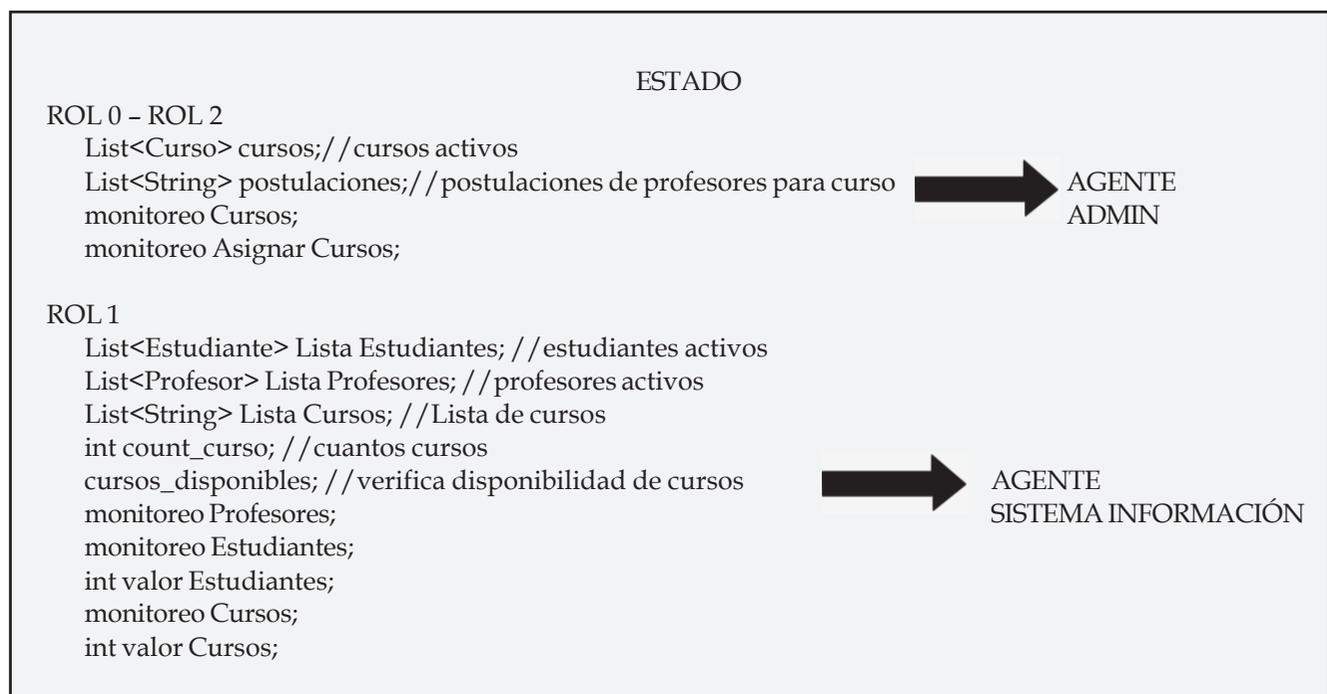


Tabla IV. Eventos de los roles del SMA.

Rol	Evento	Respuesta Evento
Rol 0	Enviar Info Asignaturas Administrar Cupos Buscar Asignatura Manejo Información Asignaturas Manejo Preferencias Profesores	Disposición Asignaturas Responder Inscripción Respuesta Búsqueda Disposición Asignaturas Asignar Profesor Curso
Rol 1	Manejo Información Estudiantes Manejo Notificaciones	Manejo Preferencias Estudiantes Notificar Inscripciones
Rol 2	Manejo Información Asignaturas Manejo Preferencias Estudiantes	Disposición Asignaturas Admin Cupos

Además, según los recursos del sistema, se especificaron los actores externos al sistema como Agente *Profesor* y *Estudiante*, los cuales darán el comportamiento en la puesta en marcha del sistema.

Especificando los métodos de colaboración descritos anteriormente, está la subasta inglesa modificada, porque no se implementa en forma completa (Fig. 11).

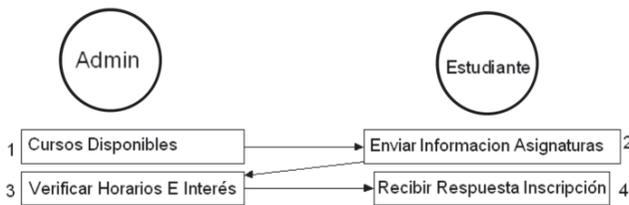


Fig. 11. Subasta inglesa (modificada).

Y la colaboración por vecindad así (Fig. 12).

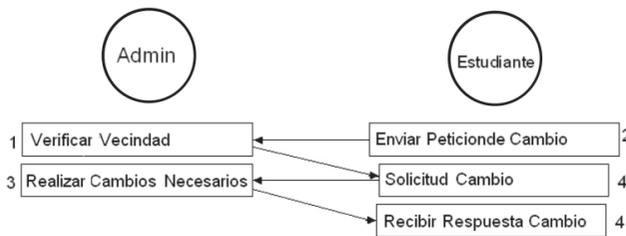


Fig. 12. Colaboración por vecindad

III. RESULTADOS

El diseño para la asignación de horarios se completa permitiendo que en cada contenedor se lean de un archivo de *properties* con las asignaturas, sus horarios correspondientes, número de profesores y de estudiantes de ese contenedor. Cada contenedor representa una sede. Cada contenedor cuenta con un agente *Admin* que asigna los horarios de la sede a los estudiantes.

Cada agente *Estudiante* obtiene las materias que tiene que ver (no necesariamente del contenedor en el que se encuentra) de forma aleatoria al ingresar al sistema. El *Estudiante* solicita inscripción en la materia al agente *Admin* del contenedor, este verifica que no haya cruces de horarios entre dichas materias y si hay cupo lo asigna. Si no, le informa que no es posible, si la materia no es del contenedor envía la petición al agente del conte-

nedor correspondiente. En caso de que el *Estudiante* no se pueda inscribir, el agente *Admin* verifica la inclinación del *Estudiante* por dicha asignatura y si es pertinente envía un mensaje al agente *Estudiante* que se encuentre inscrito en dicha materia y que tenga mayor número de materias inscritas, dicho *Estudiante* estudia la posibilidad de aceptar o rechazar este cambio y envía la respectiva respuesta al *Admin* para que este realice los cambios necesarios en los horarios y asignaturas.

En cuanto al tiempo para generación de horarios, se restringe al tiempo que transcurre desde que el primer agente *Estudiante* inicia su sesión, hasta que el último de los agentes *Estudiante* termina su sesión. Aprovechando los tiempos de concurrencia, a través del paralelismo implícito a un sistema basado en agentes. Es decir, no es la sumatoria de tiempos de las sesiones de cada uno de los agentes *Estudiante*.

El resultado esperado, se ha restringido a la operacionalización de un sistema basado en agentes que permita la generación de horarios en forma interactiva en la que estudiantes «reales» se apoyan en agentes del SMA para lograr beneficios de colaboración para generar los horarios.

Esta etapa funciona con la dinámica de los estudiantes *ceteris paribus*, es decir se suponen dados y fijos los datos de la infraestructura y de los docentes. No hubo claridad respecto de los tiempos de ejecución y el uso de otros recursos computacionales, por lo que parece requerirse de una técnica de medición automática asociada a la ejecución «paralela» de los agentes en un sistema basado en agentes. No obstante lo anterior, no se considera un fracaso en la prueba sino la oportunidad para desarrollar un instrumento de medición apropiado para este tipo de sistemas.

IV. TRABAJOS RELACIONADOS

La búsqueda para una solución al problema de asignación de cursos y generación de horarios, ha sido un actividad históricamente importante y activa al punto de que existe una reunión bienal sobre el tema: PATAT Practice and theory of automated timetabling. El tema está activo tanto en Colombia [7] como en Hispanoamérica [5]-[6] y el resto del mundo [23]-[25].

Tabla V. Comparación de tecnologías de solución de problema de generación de horarios.

Tecnología de solución	Sistemas basados en agentes	Algoritmo genético auto-adaptativo	Very large-scale neighborhood y BWACS)	Computación Evolutiva (optimización search (VLSN)	Colonia de Hormigas (ACS-Ant Colony System estocástica)
1. Permite concurrencia de planificadores de la Universidad, docentes y estudiantes.	Si	No	No	No	Si
2. Permite Interactividad con todos los usuarios	Si	Si	No	No	No
3. Dinámico en manejo de restricciones.	Si	Si	Si	No	No
4. Admite múltiples criterios de restricción	Si	Si	Si	Si	Si
5. Conocimiento previo de la longitud del horario no requerido	Si	Si	No	No	No

La literatura reporta una cantidad importante de abordajes a soluciones al problema con algoritmos genéticos [26]-[28], algoritmos bioinspirados como colonia de hormigas [29]-[30], computación evolutiva [31]-[32] y algoritmos de búsqueda de vecindarios a gran escala [21]. Desde 2004, los sistemas basados en agentes ocupan un lugar importante como método heurístico para resolver el problema de la asignación de horarios [33]-[38]. Frente a los criterios de evaluación presentados en la Tabla V, en la que se presentan los resultados de comparación de cinco tecnologías de solución para el problema de generación de horarios, los sistemas basados en agentes muestran ser una alternativa que facilita la interactividad de los usuarios de un sistema de planificación y generación de horarios y, a la vez incorporando de manera dinámica las restricciones sobre el sistema, lo cual puede disminuir la cantidad de llamadas telefónicas, mensajes y reuniones asociadas al proceso de generación de los horarios, lo cual a su vez –en forma indirecta– reduce el costo del proceso mismo. Todo lo anterior justificó plenamente haber abordado este problema complejo desde una óptica que busca la disminución de la complejidad computacional a través de un sistema basado en agentes.

V. CONCLUSIONES

Mediante la utilización de la herramienta CASE se desarrolló un sistema basado en agentes para la generación cooperada de horarios de clase. Las pruebas efectuadas aún no muestran resultados concluyentes en relación con el mejoramiento de la complejidad computacional inherentes a este tipo de actividad.

No obstante, se esperan mejoramientos marginales que pueden ser más importantes que la disminución de los tiempos computacionales, como la satisfacción de los usuarios/estudiantes.

Una medición rigurosa de la disminución de los tiempos y costos en un entorno de stress permitirá aportar datos adicionales respecto de la conveniencia de la utilización de la tecnología de sistemas basados en agentes para la solución del problema de generación de horarios.

Para la medición rigurosa de esos tiempos con la finalidad de comparar el desempeño de estos sistemas basados en agentes en contraste con otras aproximaciones de solución al problema de los horarios (timetable problem), puede requerirse del desarrollo de un instrumento informático *ad-hoc*.

REFERENCIAS

- [1] R. M Karp, «Reducibility among combinatorial problems» in R.E. Miller and J.W. Thatcher (Eds.), *Complexity of computer computations*. New York. Plenum Press. Pp. 85-103, 1972.
- [2] V. Bardadym, «Computer Aided School and University Timetabling. The New Wave» *Lecture Notes in Computer Science Series*, Vol. 1153, pp. 22-45, 1996.
- [3] E. Burke, K. Jackson, J. Kingston y R. Weare, «Automated University Timetabling: The State of the Art» *The Computer Journal*. 1998; 40(9):565-571.
- [4] M. Carter y G. Laporte, «Recent Developments in Practical Course Timetabling» *Lecture Notes in Computer Science*. Vol. 1408, pp. 3-19, 1998.
- [5] R. Hernández, J. Miranda P., P.A. Rey, «Programación de Horarios de Clases y Asignación de Salas para la Facultad de Ingeniería de la Universidad Diego Portales Mediante un Enfoque de Programación Entera» *Revista Ingeniería de Sistemas*, Vol. XXII, Año 2008. pp. 121-141, 2008.
- [6] J. Gallart Suárez, F. Alva Manchego, N. A. Alama, G. Bejarano Nicho, «Generación Inteligente de Horarios Empleando Heurísticas GRASP con Búsqueda Tabú para la Pontificia Universidad Católica del Perú» *Revista de Ingeniería Informática PUCP*, Vol. 1 No.1, Artículo 2, abril 2010. Pp.15-23, 2010.
- [7] J.M. Mejía Caballero, C. Paternina Arboleda, «Asignación de Horarios de Clases Universitarias Mediante Algoritmos Evolutivos» *Revista Educación en Ingeniería*, Universidad del Norte, Barranquilla (Colombia), ISSN 1900-8260, Junio de 2010 • N°. 9 • Pp 140-149 (2010). • Publicada en línea por la Asociación Colombiana de Facultades de Ingeniería -ACOFI- www.acofi.edu.co, 2010.
- [8] A. Barr, E.A. Feigenbaum, *The Handbook of Artificial Intelligence*. Vol. 1. Addison Wesley. Stanford, CA., 1989.
- [9] N. Graham, *Artificial Intelligence: Making machines «think»*. Tab Books. Blure Ridge Summit, 1979.
- [10] P.H. Winston, *Artificial Intelligence*. Addison Wesley. Phillipines, 1979.
- [11] J. Negrete, *De la filosofía a la inteligencia artificial*. Noriega, España, 1992.
- [12] L.Von Bertalanffy, *Teoría general de los sistemas*. Fondo de Cultura Económica, Bogotá, 1994.
- [13] J. Ferber, *Multiagent systems, an introduction to distributed artificial intelligence*, Addison Wesley, 1999.
- [14] Á. Espinosa, «Una visión cibernética de las organizaciones sociales» in: H. Andrade, I. Dyrer, Á. Espinosa, H. López, R. Sotaquirá, *Pensamiento sistémico: Diversidad en búsqueda de unidad*. Ediciones Universidad Industrial de Santander, Bucaramanga, 2001.
- [15] R. Ashby, *Introducción a la cibernética*. Ediciones Nueva Visión, Buenos Aires, 1976.
- [16] O. López-Cruz, V. Muñoz, «Trabajador, trabajo y sociedad: Una relación que se complejiza en la interacción» In: *Revista de Tecnología*, Vol. 5 No.2. Universidad El Bosque. Bogotá D.C. pp.59-77. Jul-Dic., 2006.
- [17] M.J. Wooldridge, *An introduction to multiagent systems*, 2nd ed. John Wiley & Sons. Great Britain, 2009.
- [18] G. Weiss, *Multiagent systems: a modern approach to distributed artificial intelligence*, MIT Press, 1999.
- [19] C. Iglesias, M. Marijo, J. González «A survey of agent oriented methodologies» in: *ATAL '98 Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages*, 1999.
- [20] J. Rodríguez, «Metodología AOPOA» in E. González, C. Bustacara (Eds): *Desarrollo de aplicaciones basadas en sistemas multiagentes*. Ediciones Pontificia Universidad Javeriana, Bogotá, 2007.
- [21] C. Meyers, «J.B. Orlin «Very large-scale neighborhood search techniques in timetabling problems» in *Proceeding PATAT'06 Proceedings of the 6th international conference on Practice and theory of automated timetabling VI*, 2007.
- [22] G. Veysel «Stability of an Asynchronous Swarm With Time-Dependent Communication Links» *Transactions on Systems, Man, and Cybernetics*, Vol. 38, 2008.
- [23] M. Tuga, R. Berretta, A. Mendes, «A Hybrid Simulated Annealing with Kempe Chain Neighborhood for the University Timetabling Problem» in *Computer and Information Science*, 2007. ICIS 2007. 6th IEEE/ACIS International Conference. Pp. 400-405, 2007.
- [24] M. Davoudzadeh, R. Rafeh, R. Rashidi «Linear Solution for the University Timetabling Problem» in *Computer and Electrical Engineering*, 2009. ICCEE '09. Second International Conference. Vol. 2 Pp. 54-57, 2009.
- [25] K. Nguyen, D. Nguyen, K. Trieu, N. Tran, «Automating a Real-World University Timetabling Problem with Tabu Search Algorithm» in *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, 2010 IEEE RIVF International Conference. Pp. 1-6, 2010.

- [26] R. Perzina, «Solving Multicriteria University Timetabling Problem by a Self-adaptive Genetic Algorithm with Minimal Perturbation Information Reuse and Integration» in 2007. IRI 2007. IEEE International Conference. Pp. 98-103, 2007.
- [27] H. Kanoh, Y. Sakamoto, «Interactive timetabling system using genetic algorithms». Systems, Man and Cybernetics, 2004 IEEE International Conference. Pp. 5852-5857, 2004.
- [28] S. Abdullah, H. Turabieh, «Generating University Course Timetable Using Genetic Algorithms and Local Search. Convergence and Hybrid Information Technology» in 2008. ICCIT '08. Third International Conference. Vol. 1. Pp. 254-260, 2008.
- [29] T. Lutuksin, P. Pongcharoen, «Best-Worst Ant Colony System Parameter Investigation by Using Experimental Design and Analysis for Course Timetabling Problem» Computer and Network Technology (ICCNT), 2010 Second International Conference. Pp. 467-471, 2010.
- [30] M. Ayob, G. Jaradat, «Hybrid Ant Colony systems for course timetabling problems» in Data Mining and Optimization, 2009. DMO '09. 2nd Conference. Pp. 120-126, 2009.
- [31] C.Y. Cheong, K.C. Tan, B. Veeravalli, «Solving the Exam Timetabling Problem via a Multi-Objective Evolutionary Algorithm - A More General Approach» in Computational Intelligence in Scheduling, 2007. In: SCIS '07. IEEE Symposium. pp. 165-172, 2007.
- [32] M. Aldasht, M. Alsaheb, S. Adi, M.A. Qopita «University Course Scheduling Using Evolutionary Algorithms» in Computing in the Global Information Technology, 2009. ICCGI '09. Fourth International Multi-Conference. Pp. 47-51, 2009.
- [33] L. Di Gaspero, S. Mizzaro, A. Schaerf «A MultiAgent Architecture for Distributed Course Timetabling» In Proc. of the 5th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2004), pp. 471-474, 2004.
- [34] D. Strnad, N. Guid «A Multi-Agent System for University Course Timetabling» Journal Applied Artificial Intelligence Vol. 21 Issue 2, pp.137-153, Feb. 2007.
- [35] M. Oprea «Multi-Agent System for University Course Timetable Scheduling» in ICVL Proceedings, pp. 231-238., Bucharest, 2006.
- [36] Y. Yang, R. Paranjape, L. Benedicenti «An Agent Based General Solution Model for the Course Timetabling Problem» in The Fifth International Joint conference on Autonomous Agents, (AAMOS'06), Hakodate, Hokkaido, 2006.
- [37] M.-H. Verrons, P. Mathieu «How to solve a timetabling problem by negotiation?» in Proceedings of 6th International Conference on the Practice and Theory of Automated Timetabling (PATAT), pp. 502-505, 2006.
- [38] E. Kaplansky, A. Meisels «Negotiation among Scheduling Agents for Distributed Timetabling» in Proceedings of 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT) pp. 517-520., Pittsburgh, (2004).

