Algoritmos Genéticos Basados en Criterios para rompecabezas NP-Completos (•••)

Francisco Gindre•, Nicolas Jafelle•, David Kotlirevsky•, Santiago Scaine•, David Trejo Pizzo•, Gabriel Barrera•, M. Daniela López De Luise••

Resumen

El objetivo principal de esta investigación es desarrollar un Sistema Inteligente Robotizado (SIR) resuelva un rompecabezas desconocido en un lapso de tiempo reducido. SIR aplica técnicas de reconocimiento de patrones junto con Algoritmos Genéticos. Investigando el estado del arte, la naturaleza NP-Completa del problema aparece como el común denominador. SIR se basa en esas experiencias para aportar una nueva aproximación al problema. Este enfoque involucra la conversión del puzzle a un modelo de Teoría de Grafos. El estudio de este modelo sumado a la inclusión de una analogía y enfoque de solución diferentes son los principales aportes de este trabajo. A su vez describe marcos teóricos y prácticos y el estado actual del proyecto y el trabajo a futuro.

Palabras clave: robótica, rompecabezas, Algoritmo Genético

Abstract

The primary goal of this research is to develop an Intelligent Robotic System (SIR in Spanish) that solves unknown jigsaw puzzles in a reduced amount of time. This model applies pattern recognition techniques and feature detection in conjunction with Genetic Algorithms. It relies on the knowledge of authors as a working base for a new approach to the problem; involving a conversion of the puzzle model to a Graph Theory model.

Keywords: Robotic, puzzle, Genetic Algorithms, Graph Theory

Fecha de Recepción: agosto 2012 | Fecha de aceptación: octubre 2012

• Investigadores del AIGroup, Universidad de Palermo

•• Directora del AIGroup, Universidad de Palermo.

••• Este trabajo fue aprobado y presentado en el Congreso TRIC V organizado por IEEE CIS (Computational Intelligence Society) Argentina que se desarrolló en el marco del IEEE Argencon 2012 del 13 al 15 de junio de 2012 en la UNC.

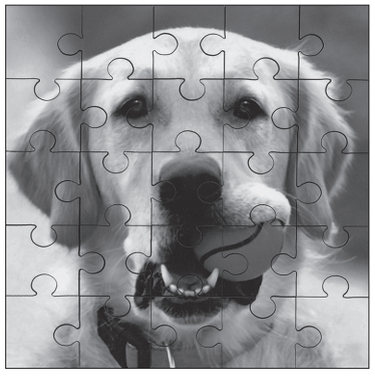
Introducción

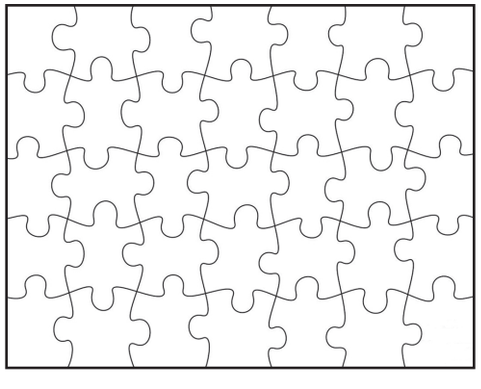
Fabricantes de todo el mundo manufacturan rompecabezas con variedad de formas, tamaños y tipos de piezas. Existe una falta de clasificación común o terminología que identifique unívocamente cada tipo de rompecabezas. La investigación preliminar para este proyecto se encontró con este inconveniente. Por ello se definen algunos tipos. El término *Rompecabezas estándar* (ver figura 1.1) es usado en [2] para describir rompecabezas hechos mediante el corte de una figura impresa en una superficie firme en piezas encastrables. Nótese que este tipo de rompecabezas solo denota la naturaleza encastrable del mismo. No se refiere al tamaño o a la forma de las piezas o del rompecabezas mismo.

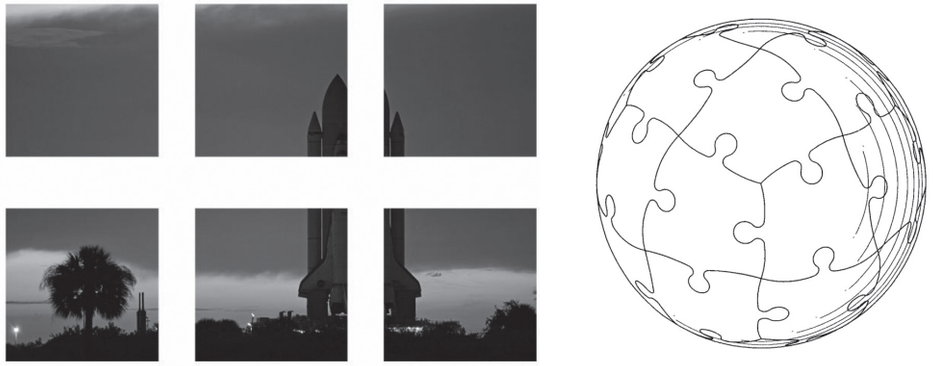
Un *rompecabezas cuadrado* (figura 1.2) refiere a aquellos que tienen piezas cuadradas y tamaño uniforme, que conforman una grilla que completa la imagen solución. Al carecer de una forma curvilínea es posible encontrar múltiples soluciones para ellos, dependiendo de la imagen.

Existe un super-conjunto de rompecabezas llamados *"rompecabezas canónicos",* definidos en [11]. Incluye todos aquellos que tienen piezas de cuatro bordes con cuatro rotaciones posibles que, cuando son ensambladas de forma contigua, forman una grilla rectangular que conforma la imagen resultado. Esta última definición no hace distinción sobre si el borde de las piezas es curvo o no. Se encontraron otros enfoques como el de los *rompecabezas apictóricos* (figuras 1.3 y 1.4)[9], que no contienen una foto o gráfico característico que permita llegar a su resolución, haciendo que ésta dependa exclusivamente de cómo se encastran las piezas.

Muchos autores tocaron el tema durante las últimas cuatro décadas. En 1968, Freeman y Garder[9] trataron el ensamblaje de rompecabezas apictóricos al reconocer las limitaciones técnicas del momento para reunir otra información que no fuese la forma de las piezas. Su trabajo sentó un precedente para muchos otros [1][2][3][5][7][8][11][12], incluido este trabajo.







La complejidad del problema nace de sus propias características: es compatible al *Wang's tiling problema*[6] y *NP-Completo*[2][13][7]. También puede aplicarse a diversos dominios. La reconstrucción de documentos rotos, la restauración arqueológica y el ensamblado de máquinas son tareas complejas que insumen tiempo, lo cual hace evidente la necesidad de automatizar la solución de dichos problemas [12][20]. Como respuesta a esta necesidad el objetivo de SIR es construir un sistema robótico con la habilidad de construir un rompecabezas cuadrado desconocido en un lapso de tiempo reducido. Este trabajo describe el diseño y el desarrollo del Módulo Lógico usando un nuevo enfoque basado en la Teoría de Grafos [25], su sustento y su aporte a la materia.

Desde [9], muchos otros autores sugirieron y desarrollaron distintos enfoques para alcanzar la solución de rompecabezas. Los rompecabezas apictóricos [8] son usados como una forma de resolver el problema de ensamblaje. Su objetivo es desarrollar software capaz de reconstruir un objeto comparando las características físicas de las partes que lo componen. Burdea y Wolfson hacen uso de un brazo robótico para ensamblar un puzle blanco luego de que las piezas son analizadas y se ha alcanzado una solución. El trabajo introduce el concepto de solución global. Su algoritmo resolutivo, llamado *KBEST*, recrea el proceso de ensamblaje hecho por humanos, donde el primer paso es identificar el marco exterior del rompecabezas mediante el uso de heurísticas y un enfoque similar al problema del viajante de comercio (*TSP* [4]). Este proceso itera sobre los marcos internos del puzle usando las *K* mejores soluciones encontradas en las distintas iteraciones hasta que una solución global es alcanzada. Este enfoque puede ensamblar hasta 208 piezas. Otra solución iterativa que también descarta información pictórica es propuesta en [14]. Aborda el problema de encontrar los bordes correctos mediante la búsqueda de los puntos críticos en los *istmos* de las piezas para luego probar iterativamente todos los bordes posibles. Este enfoque es capaz de ensamblar rompecabezas estándar

de hasta 24 piezas. El algoritmo *KBEST* visto en [8] es revisado en [5]. En ese trabajo, Chung propone ensamblar un *rompecabezas estándar* usando su forma y color, proclamando que enfocarse en una sola característica derivaría a una solución incorrecta debido a la similitud entre piezas. Para resolver el problema diseñaron 3 algoritmos: *AP*, que recrea el enfoque humano visto anteriormente; *TSP/AP* que combina el enfoque humano con la estrategia considerar las piezas como si fuesen ciudades del problema del viajante de comercio; y el tercero, combina el enfoque *KBEST* con el de *TSP*. El desempeño de este último es el mejor de los tres, pero solo fue capaz de resolver ciertos puzles con un límite de 54 piezas. La principal contribución fue aportar el enfoque *TSP* a la materia. Dicho enfoque [4] en conjunto con la *solución global* [8], lleva al uso de algoritmos genéticos [15][16].

Esta técnica es usada en [3] para resolver un *rompecabezas cuadrado* de m por n piezas usando el color como información. La composición del borde es la única información requerida en este enfoque. Cada pieza es capturada en una paleta blanco y negro de 1 bit, donde cada borde está compuesto por los píxeles externos de una pieza. Durante este proceso la información interna de cada pieza es descartada, pues no es parte de la comparación. Este enfoque pudo encontrar la solución global de ciertas imágenes segmentadas hasta una dimensión de 8x8 piezas. Años después esta técnica es reusada en [1] con una perspectiva diferente. Las piezas son capturadas en *RGB*, conteniendo así más información sobre cada una de ellas. A su vez el ensamblado de las piezas no sería a partir de las piezas individuales, en cambio, las primeras se ensamblarían en bloque con distancias óptimas entre si. Luego esos bloques serán comparados como si fuesen piezas. El Algoritmo genético (GA) optimiza la distancia entre piezas para construir bloques óptimos que son comparados con otros bloques y piezas individuales. La principal ventaja de este método es que los bloques óptimos no son afectados por operaciones futuras con los cromosomas, reduciendo así la naturaleza aleatoria de la evolución del GA. Este enfoque es capaz de ensamblar puzles de 16x12 piezas. La tesis de Tybon [2] presenta un relevamiento interesante sobre el estado del arte y los diferentes métodos que pueden ser usados para resolver rompecabezas cuadrados. Presenta una galería de posibles enfoques al problema del rompecabezas que incluye programación lineal[21], Evolutiva y GA[15][16], *Simulated Annealing (SA), Tabu Search* [18][19] y programación basada en restricciones. Tybon implementó estas distintas técnicas y las probó utilizando rompecabezas de 2x2 hasta 4x5 piezas con soluciones únicas. Su tesis concluye que GA y SA son las técnicas más adecuadas. [12] Utiliza programación dinámica y *Hungarian Procedure* como alternativa superadora del enfoque "pieza por pieza". Estas técnicas son usadas para resolver un problema de *image-scrambling* con mosaicos de 10x10 considerando la textura de los mosaicos contiguos.

El análisis del estado del arte describe diferentes enfoques para resolver rompecabezas. De acuerdo a [5][9][11][20] es posible inferir que la extracción

de piezas materiales y su subsecuente inclusión en el modelo computarizado no es el factor clave que limita la cantidad de piezas que dichas soluciones pueden tolerar. Por el contrario, los medios para manejar la naturaleza NP-Completa del problema determina la escalabilidad de esos enfoques. Basado en esta conclusión, el módulo lógico de SIR tienen la mayor prioridad de implementación. Por ello esta publicación se concentra en dicho subsistema. La siguiente sección describe brevemente la arquitectura de SIR contextualizando su ubicación en el flujo de trabajo general. La sección 3 discute el nuevo el diseño del nuevo enfoque, sus ventajas y desventajas. Por último la sección 4 ilustra el trabajo a futuro.

Arquitectura, flujo y alcance

SIR está diseñado como un sistema con tres módulos independientes. El primero es el *Módulo de Captura,* compuesto por una cámara web corriente y un módulo de software que articula un *framework* para controlar la cámara y activar la captura. La salida de este módulo es un conjunto de piezas identificadas por su orden de aparición y ubicación que servirá de entrada al módulo siguiente. Los datos del rompecabezas son convertidos a una abstracción de la imagen capturada, cambiando el foco de una matriz de píxeles a piezas bien formadas e identificadas que constituyen un rompecabezas. Cada pieza es identificada y sus bordes son considerados la única región de interés requerida para resolver el problema [1][2][3]. La dimensión del puzle es todavía desconocida hasta que la salida del módulo. El *Módulo Lógico* utiliza un GA para encontrar la solución del rompecabezas dado mediante la abstracción del problema original a uno de optimización de grafos (ver sección 3). La salida de este proceso produce la solución concreta y la dimensión del puzle. El *Módulo Robótico* es el paso final del flujo de trabajo. Está diseñado para consumir la salida del módulo anterior y ejecutar el ensamblado de un puzle real. Está compuesto por un kit de robótica Lego MindStorms NXT 2.0®[22] que se conecta mediante Bluetooth®[23] a una laptop que corre el software SIR. El resultado del módulo previo es interpretado y ejecutado, guiando al robot por los pasos necesarios para completar el rompecabezas. SIR asume las siguientes restricciones para controlar el alcance del proyecto: 1) el sistema se limita a *rompecabezas cuadrados*, 2) las piezas del puzle están ubicadas sobre una superficie plana con una cromática distintiva para facilitar su extracción, 3) el color de dicha superficie es el único parámetro que recibe el sistema, 4) el sistema admite solo rotaciones de 90 grados, 5) la ejecución se asume en un ambiente controlado que cumple con lo anterior.

Un nuevo enfoque de Algoritmos Genéticos

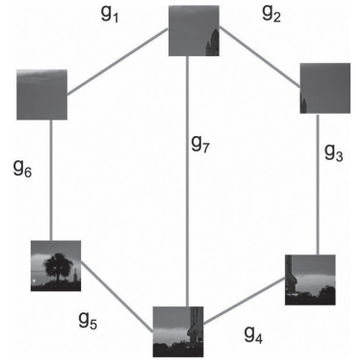
Resolver un rompecabezas puede ser definido como el descubrimientos de una solución global que minimice todas las distancias comparadas entre piezas [1][2][3][4][8][11][12]. De hecho, un *rompecabezas cuadrado* que contenga gradientes o grandes porciones de color uniforme puede tener varias soluciones que lleven a máximos locales [2]. Una coincidencia perfecta en la mayoría de las piezas, sumada a las diferentes rotaciones de las piezas en el área gradiente o de color uniforme pude generar dichos múltiples máximos globales. Dichas áreas tienden a que algunos algoritmos resolutivos converjan a resultados falsos positivos (o a ninguno). Las soluciones sub-óptimas y el crecimiento factorial-exponencial[2] son las principales motivaciones para aplicar GA[15][26]. Las siguientes sub-secciones explican algunos contrapuntos de esta técnica.

La distancia global entre piezas es la función objetivo, donde una solución global debe optimizar la distancia borde a borde. [2] muestra la influencia que tienen las restricciones en el resultado final, mientras que [3] prueba la efectividad en resolver algunos tipos de rompecabezas a pesar de las limitaciones impuestas por la analogía escogida para determinar que es un cromosoma y una población. Sumado al criterio anterior, las características NP-Completas y de optimización de distancias derivaron en una abstracción del problema que presenta similitudes al *TSP*[4]. El enfoque *TSP* fue destacado entre los mas efectivos para resolver este problema para puzles grandes[5]. En el nuevo enfoque propuesto el puzle es representado como un multigrafo [25] (sin ciclos), donde cada *nodo* es una *pieza*. Cada *arista* establece una adyacencia *borde a borde* entre dos piezas. La representación del GA fue rediseñada para adecuarse a este enfoque. La arquitectura del algoritmo debe soportar la incertidumbre y las restricciones presentes en el problema original. La incertidumbre sobre la cantidad de piezas involucradas en el puzle ayudó a la idea de que el cromosoma representa una solución candidata para un rompecabezas sobre otras consideraciones. Por lo tanto, el tamaño del cromosoma está atado a la dimensión del rompecabezas. El modelo presenta un cromosoma de longitud fija. Cada gen representa una *arista* del grafo y los lados de las piezas involucrados en esa adyacencia. Los componentes son discutidos en detalle luego de abordar las restricciones comunes que aplican al rompecabezas y su versión en el modelo de grafos.

Los rompecabezas cuadrados tienen reglas estrictas en cuanto a las adyacencias. Estas son detalladas en la subsección 1. Cuando las reglas no son validadas correctamente, las operaciones de mutación y cruza pueden causar adyacencias que no son física o lógicamente posibles. Las soluciones incorrectas se generan cuando estos errores no son considerados. Se propagan a lo largo del modelo en las distintas iteraciones sacando al algoritmo del espectro de búsqueda correcto.

De aquí se concluye que una de las desventajas mas importantes de este modelo es la necesidad de validar cada una de las operaciones realizadas.

Para evitar posibles sesgos en el espectro de búsqueda, los GA son inicializados con poblaciones aleatorias [26][27]. El presente enfoque hace una excepción a esa regla. Las poblaciones iniciales, deben ser creadas en un proceso de aleatoriedad controlada. Es decir, que los cromosomas creados deben ser aleatorios, pero aún así cumpliendo con las restricciones del problema. Otras operaciones como la mutación y la cruza requieren un control estricto para asegurar que el GA busque más allá del problema dado.



Esta validación se hace durante la selección y otras operaciones que crean nueva descendencia para garantizar la consistencia de todo el proceso. Las experiencias en [1][2][3] conducen al diseño de la abstracción multigrafo (fig. 3.1). El modelo de teoría de grafos ofrece versatilidad de aplicación junto con un espectro de búsqueda mayor. Es así posible representar cualquier disposición para el puzle. Aún aquellas que son inválidas.

Restricciones del modelo de grafos

Las adyacencias inválidas deben ser detectadas y descartadas. Dado un puzle de m x n piezas el número de genes corresponde a:

$$a(m,n) = 2mn - n - m . \quad (1)$$

Los nodos de 4 aristas son:

$$j(m,n)= (m- 2)(n-2) . \quad (2)$$

Los nodos de 3 aristas son:

$$k(m,n) = 2(m-2) + 2(n-2) . \qquad (3)$$

Los nodos de 2 aristas (esquinas) son siempre cuatro. Las validaciones de cromosomas y genes deben realizarse continuamente para asegurar la consistencia del proceso. La complejidad del código para realizar dichas operaciones se reduce gracias a conceptos de Teoría de Grafos. Un cromosoma es evaluado al representarlo como una matriz de adyacencia[25]. El grado de los vértices determina el tipo de adyacencia que representa (2, 3, o 4 aristas). Estas son contabilizadas y comparadas contra los valores esperados de las funciones (1),(2) y (3). Cualquier cromosoma que no cumpla con las restricciones considerado *infértil*. La abstracción propuesta incluye métodos de validación mencionados en [2]. Es así como se evita la evaluación compleja de individuos y soluciones como procesos separados y define mas parámetros para minimizar el espectro de búsqueda del GA.

Elementos del algoritmo genético

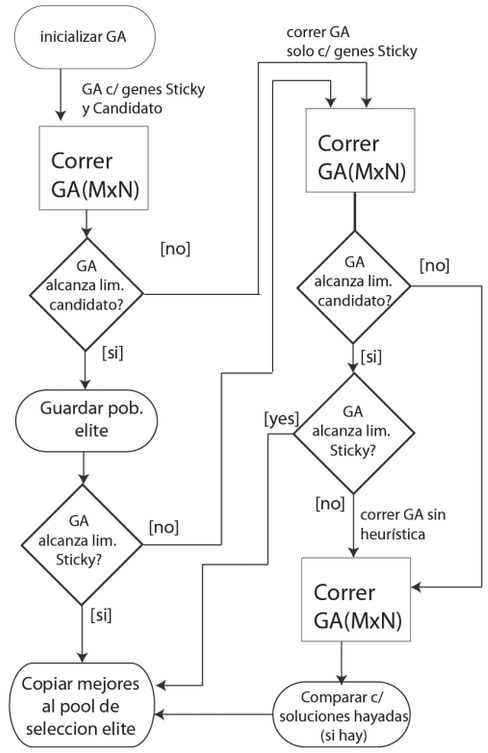
La función de ajuste (fitness) se define como

$$F(x) = \sum_{i=1}^{n} [1 - P(f(S_i))] \qquad (4)$$

Donde P(x) es el porcentaje de coincidencia borde a borde entre dos piezas y $f(s_i)$ es una función que retorna dos vértices (piezas) involucradas en la arista S_i del grafo. El parámetro x representa la entrada de un cromosoma.

Una *población* se define como un conjunto de *cromosomas fértiles*. Es evolucionada hasta que se detecta una convergencia. Un cromosoma representa un grafo que ilustra una posible solución al puzle. Estos son analizados con los criterios mencionados. Si alguno no es alcanzado, el individuo es considerado infértil asignándole cero a su valor de *fitness*, lo que causa su supresión de la población. Un *gen* representa una arista que conecta dos piezas del puzle en el grafo. Recursivamente, los genes son evaluados con diferentes criterios, donde la falla de alguno de ellos vuelve al cromosoma infértil. Los genes están compuestos por dos *alelos*. Cada uno de ellos representa un borde de una pieza del rompecabezas.

Se ejecuta un GA por cada disposición MxN posible para el total de las piezas. Los cromosomas más aptos de cada instancia de ejecución son guardados como una población de élite hasta que todo el proceso concluye. Luego, se selecciona el mejor de ellos como solución del rompecabezas.



Heuristic fitness classification

Abstraer el puzzle a un modelo de grafos presenta sustanciales ventajas respecto del modelo "pieza a pieza" utilizado en trabajos anteriores [1] [2] [3] [8] [11] [12] [13] [20]. El modelo de grafos aumenta el espectro de búsqueda del GA, haciendo al algoritmo menos sesgado, lo que minimiza la probabilidad de llegar resultados sub-óptimos. También provee una configuración flexible que soporta topologías distintas a las elegidas para este proyecto. Dicha capacidad se logra cambiando los criterios de validez de un puzle. Estas características tienen consecuencias. Al salir del representación como matriz de piezas, es posible que las operaciones de cruza y mutación desmantelen una población de cromosomas apta, bajando su aptitud o bien tornándola infértil. Dada la naturaleza del problema, los genes que tienen una distancia borde a borde mínima, tienen mayor probabilidad de estar presentes en una solución óptima. Para evitar que dichos genes sean separados por las operaciones, el flujo de trabajo del GA incluye una forma de etiquetar aquellos genes que superen un porcentaje de similitud entre bordes definida. SIR contempla dos límites que pueden ajustarse a distintos valores de similitud entre bordes. Esto crea dos clases nuevas de genes: *genes candidato* y *genes Sticky* (pegajosos). *Candidato* refiere a aquellos

genes que tienen un porcentaje de similitud aceptable (ej. 70%) como parte de un cromosoma apto pero no considerado "óptimo". *Sticky* se refiere a aquellos genes que tienen una distancia "óptima" (>=90%). Los genes *Sticky* o *Candidatos* tienen menor probabilidad de ser afectados por las operaciones genéticas. Esta heurística permite que las operaciones no desmantelen genes *aptos* durante la mutación o la cruza. El proceso de etiquetado es llevado a cabo por la función de ajuste. Esta capacidad puede ser activada o no durante la ejecución del GA dependiendo de la situación de convergencia del algoritmo.

Estado actual y trabajo a futuro

El proyecto descripto en este trabajo se encuentra bajo desarrollo. La aplicación robótica es el último de los pasos. El equipamiento necesario ha sido adquirido y probado. El kit Lego ha sido convertido a una máquina virtual Java soportada por el *framework* de código abierto LEjOS[24]. El módulo robótico será desarrollado en el tercer cuatrimestre de 2010.

Bibliografia

[1] Murakami T., Toyama F. Shoji K. , Miyamichi J. : "Assembly of Puzzles by Connecting Between Blocks", (2008).

[2] Tybon R.: "Generating solutions to the Jigsaw problem." Griffith University, (2004).

[3] Toyama F., Fujiki Y. , Shoji K. , Miyamichi J. : "Assembly of puzzles using a genetic algorithm." 16th International Conference on Pattern Recognition, (2002).

[4] Papadimitriou, C.H., Yannakakis M. : "Shortest paths without a map". Proc. 16th ICALP. Lecture notes in computer science (Springer-Verlag.), (1989).

[5] Chung M. G, Fleck, M. M, Forsyth D. A.: "Jigsaw puzzle solver using shape and color." Proceedings of ICSP (IEEE), 877-880, (1998).

[6] Wang J.: Introduction to NP-Completeness, 91.502 Foundations of Computer Science, (2006).

[7] Ville Lukkaril: The square tiling problem is NP- complete for deterministic tile sets. TUCS Technical Report No 754, March, (2006).

[8] Burdea G. D., Wolfson H. J.: Solving jigsaw puzzles by a robot. IEEE Trans. Robotic. Autom. v5 i6. 752-764, (1989).

[9] Freeman H. and Garder L.: Apictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern Recognition, IEEE Trans, on Electronic Comp., Vol EC-13, No. 2, 118-127, April (1964).

[10] McAdam D.: History of Jigsaw puzzles. American Jigsaw Puzzle Society. http://www.jigsaw-puzzle.org/jigsaw-puzzle-history.html.

[11] Yao F.H., Shao G.F.: A shape and image merging technique to solve jigsaw puzzles, Pattern Recognition Letters 24 (2003) 1819-1835. (2003).

[12] Alajlan, N.: Solving Square Jigsaw Puzzles Using Dynamic Programming and the Hungarian Procedure. American Journal of Applied Sciences 6(11): 1942-1948, (2009).

[13] Altman T.: Solving the jigsaw puzzle problem in linear time. Applied Artificial Intelligence, 3,(1989).

[14] Webster R. W., LaFollete P. S., Stafford R. L.: Isthmus Critical Points for Solving Jigsaw Puzzles in Computer Vision. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. 21, NO. 5, 1271-1278. SEPTEMBER, (1991).

[15] Holland, J. H.: Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor,(1975).

[16] Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press. ISBN 0-262-11170-5, (1975).

[17] Kirkpatrik S., Gelatt C. D., Vecchi M. P.: Optimization by simulated annealing, Science, (1983).

[18] Glover F.: Tabu search- Part I ORSA Journal on computing,(1990).

[19] Glover F.: Tabu search- Part II ORSA Journal on computing,(1990).

[20] Sagiroglu M. S., Ercil A.: A texture based matching approach for automated assembly of puzzles. Proceedings of 18th International Conference on Pattern Recognition, (PR'06), IEEE Xplore Press, Hong Kong, pp: 1036-1041. DOI: 10.1109/ICPR.2006.184, (2006).

[21] Dredsner E. C.: Técnicas cuantitativas: el management científico aplicado a las decisiones en la economía de empresas. 3ra edición, Ed. Universo, Bs. As, (1998).

[22] Lego®http://mindstorms.lego.com/eng/default.aspx .

[23] Bluetooth Technology. http://www.bluetooth.com

[24] LeJOS, Java for Lego MindStorms. http://lejos.sourceforge.net/

[25] Balakrishnan, V. K.: Graph Theory, McGraw-Hill; 1st edition. (1997).

[26] Bigus J. P., Bigus J.: Constructing Intelligent Agents with JavaTM: A Programmer's Guide to Smarter Applications, John Wiley & Sons, (1999).

[27] Amit K.: Artificial Intelligence and Soft Computing Behavioral and cognitive modeling of the human brain, CRC Press, (2000).