"A hybrid genetic algorithm for a Hamiltonian path problem"

Eunice Ponce de León
Roberto Santana
eunice@cidet.icmf.inf.cu, rsantana@cidet.icmf.inf.cu
Centro de Inteligencia Artificial
Instituto de Cibernética, Matemática y Física
Calle 15, #551, entre C y D, Vedado
CP 10 400, C. Habana, CUBA

Abstract

This paper deals with Hável and Erdös conjecture that asserts that all bipartite graph formed with the middle levels of n-dimensional cube is Hamiltonian, when n is odd. This means that at least one Hamiltonian cycle should be found in the graph.

We use a circular representation of the vertices of n-dimensional cube which makes it possible to introduce two group actions in order to reduce the problem to find out a Hamiltonian path in multi - level quotient graphs.

We report on a hybrid genetic algorithm for the Hável and Erdös conjecture problem. We present interesting results which show that this GA approach gives optimal solutions in multi-level quotient graph. We use bitstring representation, an evolutionary fitness function, restricted edge crossover operator, intelligent mutation, seeding and adaptive mutation rate. Three Hamiltonian paths are constructed with this method.

Key words: Genetic algorithms, NP-complete, Hamiltonian circuit, cycle or path, Hável and Erdös conjecture, graph theory, hybrid methods.

1. Introduction

This paper deals with Hável and Erdös Conjecture that asserts that all bipartite graph formed with the middle levels of n-dimensional cube is Hamiltonian, when n is odd. This means that at least one Hamiltonian cycle should be found in the graph.

An approach to Hável and Erdös Conjecture was reported by Dejter [Dejter, 1985; Dejter, 1989; Dejter and Quintana, 1991; Dejter, Cedeño and Jauregui, 1993]. In these papers, he introduced the group action approach.

We use a circular representation of the vertices of n-dimensional cube which make it possible to introduce two group actions in order to reduce the problem to find out a Hamiltonian path in multi - level quotient graphs.

The *Hamiltonian path problem* (HPP) is one of the classic problems of combinatorial optimization. It consists of finding a path through a directed graph that touches all nodes exactly once. Clearly, if a graph is fully connected this is an easy task. However, as edges are removed the problem becomes much more difficult, and the general problem is known as NP- complete [Garay and Johnson,1977].

Attempting to solve the Hamiltonian path problem directly with GAs rises many of same representation issues as in the case of traveling salesman problems (TSP) [De John, 1985], [Grenfenstette et. al.,1985]. They use the permutation representation. For example, in a 5-city TSP the string (b a c e d) means that city b is visited first, then city a and so on. However, the genetic operators used in the classical GA do not give valid offspring tours so new operators must be designed.

Crossover operators which have been suggested in the literature include Partially Mapped Crossover (PMX) [Goldberg and Lingle, 1985], Order Crossover (OX) [Oliver et. al., 1977], Order Crossover # 2 (OX2) [Syswerda, 1991], Position Based Crossover (PBX) [Syswerda, 1991], and Cycle Crossover (CX) [Oliver et. al., 1977] . In our approach we use standard representation using the advantage of the little proportion between edges and vertices that exist in the graph subjected to study.

The contribution of this paper lies in two significant aspects:

1) We use the circular representation of the vertices of n-dimensional cube which makes it possible to introduce two group actions. This result is discussed in a paper related to our research [Ponce de León, 1997].
1)
1) We introduce a hybrid genetic algorithm to solve the problem. It is possible to use this algorithm in any graph of the conjecture. An earlier version of this approach is presented in [Ponce de León et al, 1995].

In this paper we report on a hybrid genetic algorithm for the Hável and Erdös conjecture

problem. We present interesting results which show that this GA approach gives optimal solutions in multi-level quotient graph. We use bitstring representation, an evolutionary fitness function, restricted edge crossover operator, intelligent mutation, seeding and adaptive mutation rate. Three Hamiltonian paths are constructed with this method.

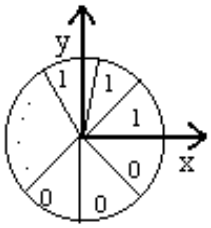
2. Problem definition and notation

Let $i,j \in Z$ with $0 < i < j$ and let $n = i + j$. Let V_{ij} be the family of (i+j) - tuple over $Z_2 = \{0,1\}$ whose Hamming weight is either i or j. Let RG_{ij} be the bipartite graph whose set of vertices is V_{ij} with adjacency given by changing exactly $j - i$ coordinates of value 0 to value 1 or viceversa. The *Hável and Erdös conjecture* asserts that the graphs RG_{ij} are Hamiltonian.

In this paper we have treated the Hamiltonicity problem on $RG_{i,i+1}$. That is $j = i + 1$. Graph $RG_{i,i+1}$ is the bipartite graph formed with the middle level of n - dimensional cube, when $n = 2\,i + 1$.

Circular representation: Each (2i+1) - tuple in $V_{i,i+1}$ is represented in circular form by the following agreement: Let the circle in the cartesian plane with 2i+1 concentric angles of the same degree. The first angle is formed with the X- axis. We associate to each angle to one coordinate of the (2i+1) - tuple starting with the first angle formed on the X- axis, we set the coordinates values in every division of the circle formed by this angles, in a counter clockwise sense.

Example: if a = (1 1 1 … 0 0 0)



Let $V_{i,i+1} = V_i \cup V_{i+1}$ in such away that $V_i \cap V_{i+1} = \varnothing$. Lets make V_{i+1} as formed by all different vectors with i+1 unitary coordinates and i zero's coordinates represented in circular way like the agreement above. In a similar way we can be define V_i.

2.1 Quotient graphs for $RG_{i,i+1}$

If Γ is a group then a *group - action* τ: Γ x G $\rightarrow$ G on a graph G = (V, E) is defined in [Dejter et al, 1988] as a pair (T: Γ x V $\rightarrow$ V, t: Γ x E $\rightarrow$ E) of compatible actions. In fact $RG = RG_{i,i+1}$ admits some group actions that ease the study of their Hamiltonicity. To

describe them it is enough to indicate their vertex set action components:

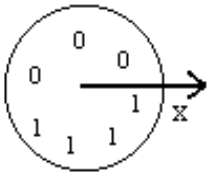
(0 Rotation of coordinate:
T_0: Γ x $V_{i,i+1} \rightarrow V_{i,i+1}$

Let $\Gamma = Z_{2i+1}$ be the group of integer residue classes modulo 2i+1. The residue class of integer $g \in Z$ is denoted g. We identify Z_{2i+1} with the group Γ_{rot} of rotations of circumference so that $1 \in \Gamma$ corresponds to the counterclockwise $2\pi/2i+1$ - radian rotation.

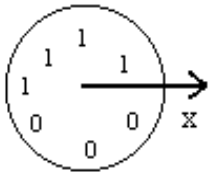
Let T_0: Γ x $V_{i,i+1} \rightarrow V_{i,i+1}$ be the Γ - action induced by Γ_{rot} . In fact, $T_0(g,v) = (g(2\pi/2i+1), v)$ for every $g \in [0,2i+1]$ and a vertex $v \in V_{i,i+1}$.

Example: Let $RG_{3,4}= (V_{3,4},E)$

$T_0(4,v) = (4(2\pi)/7,v)$



v



$T_0(4,v)$

(1 Complementation:
T_1: Z_2x $V_{i,i+1} \rightarrow V_{i,i+1}$

where $T_1(1,v) = \bar{v}$, where $1 \in Z_2$ and v is as before. $\bar{v}$ denote the negation logic of v.

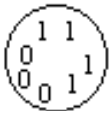
In [Ponce de León, 1997] we showed that there exists such a quotient graph $H_{i+1} = RG_{i,i+1}/ T_0,T_1$ that any Hamilton path bearing a mild condition can always be lifted to a Hamilton cycle in $RG_{i,i+1}$. The mild condition is:
"Each end of the Hamilton path in H_{i+1} has two loops".

H_{i+1} graph can be defined like a multi-level graph as it follows.

2.2 Multi - level graphs (H_{i+1})

Let $H_{i+1} = (V_h E_h)$ be a graph whose vertex set is V_h and whose edge set is E_h.
Each vertex $v \in V_h$ is representing a circular vector of Hamming weight i+1 that has no endpoints [Ponce de León, 1997]. Let $|V_h|$ be a cardinal of set V_h, then $|V_h| = \binom{2i+1}{i} / (2i+1)$, which is a catalan number. This number represents all possible binary strings represented in circular form with exactly i coordinates in 0 and i+1 coordinates in 1.

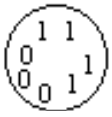
For example: for i = 3, we have five vertices

The adjacency relation in H_{i+1} goes as follows: Consider for example the vertex  for H_4. Select a coordinate with value 1. There is an adjacent vertex to  that can be seen as follows:

(1) write again 1 for the chosen coordinate

(2) write all the other values changed, for the remaining coordinates.

For example:

 is adjacent to  .

Definition.- Let $v \in V_h$ be such that v has *two loops*. Then there exist two different unitary coordinates in v, in such away that by the adjacency relation, v is again obtained.

2.3 Levels of H_{i+1} graph

In [Ponce de León,1997] we introduced a natural classification of the vertices of V_h, according to the number of unitary and zero blocks of the circular vectors that they represent.

Definition.- A *block* is the longest sequence of consecutive coordinates of a circular vector that has the same value. If all the coordinates are 1 the block is said to be a unitary block. The same happens with the zero blocks.

The circular vectors (respectively vertices) with the same number of unitary and zero blocks constitute one level of the graph H_{i+1}. There is an even number of blocks. We consider one level of graph H_{i+1} as a partial subgraph of graph H_{i+1}. We denote it $H_{i+1,b}$. where b is the level. When b=1, we have the first level and this represents that there is one unitary block and one zero block. When b=2 we are in the second level. There are two unitary blocks and two zero blocks, and so on. This multi-level graph is showed with an example of graph H_{i+1}, with i=4 as shown in Fig. 1.

In general, the graph H_{i+1} is a multigraph, but when we make the adjacency matrix, because

we consider only the edges in our problem, it is considered a simple graph. In this graph there are some vertex whose degree is two.

The GA works in each graph $H_{i+1,b}$ at all levels, in a separate way. An algorithm that generates all vertices is given in [Santana and Ponce de León,1997].

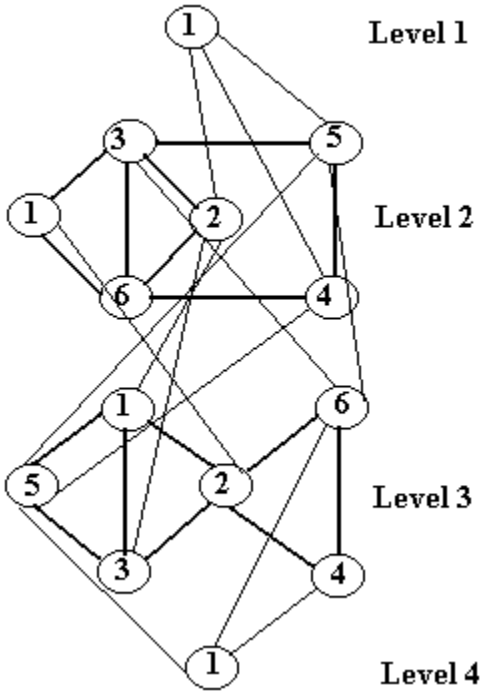


Figure 1.- Multi-level graph H_5. Each vertex is representing a circular vector of Hamming weight 5 that has no endpoint.

3. GA for a Hamiltonian path problem in multi-level graphs

An approach to the search of Hamiltonian paths or cycles using a genetic algorithm with binary representation was reported by De Jong and Spears in [De Jong and Spears, 1989]. In their paper the problem of the search of a Hamiltonian path is addressed for oriented graphs. A transformation is made to the boolean satisfiability problem (SAT). In the binary string of the chosen representation each bit represents the veritable value of the i variable in the boolean SAT expression. This transformation needs an additional preprocessing step. Terms represent the legal combinations of the edges. Nevertheless the big amount of terms in the obtained expressions did not let the authors to investigate graphs with more than 11 nodes. Besides, the hardest problem is the selection of the fitness function [Michalecwicks, 1992].

We have incorporated to our hybrid genetic algorithm new operators that deal with the particular constraints of this problem and improve the efficiency bounds. Nevertheless the GA keeps the basic features of a SGA. Thus, in the remaining of this paper we will use the terminology of Goldberg [Goldberg, 1989].

3.1 Problem Encoding

The graphs $H_{i+1,b}$ are characterized by a little proportion between edges and vertices, which was between 1.5 and 2.5 (See Table #1 in Experimental results). This fact suggested us the possibility of using an edge representation.

Candidate solutions are represented or encoded, as binary string or bitstrings. Each bitstring comprises M bit that mask the M edge in graph $H_{i+1,b}$: bit value 1 stands for "select the corresponding edge", whereas bit value 0 stands for "do not select the corresponding edge". This correspondence is given by adjacency matrix of graph $H_{i+1,b}$. If the graph have N vertices, then the bitstring have N - 1 bit put in 1.

An example of this encoding is the following 8-bit bitstring, when i=4, b=2, N=6.
(1 0 1 0 1 1 0 1)
This bitstring represents a subgraph of the graph $H_{k+1,b}$. Knowing that all the chromosomes have size M adn exactly N - 1 bits with value 1, it is possible to calculate the size of the search space. This corresponds to $\binom{M}{N-1}$ which is smaller than that result of a permutation encoding .

3.2 The Fitness Function

For calculating the objective function, first we count the degree of each vertex of the subgraph generated by the chromosome. This information is entered in the vector W= $(w_1,...,w_k)$. Let 2 = (2,...,2) the vector whose coordinates are all 2, then
D = | 2 - W | is the vector of difference coordinate to coordinate between 2 and W.
Thus, objective function F is define as F = 1000 / D

When two coordinates of vector W are 1 and the rest zeros, F will reach its maximum value, that is 500. This function determine a necessary condition for the existence of the path, but this is not sufficient.

If two vertices have a degree of one and the rest a degree of two in the chromosome, it starts to take on the form of a path. In this moment, the necessary condition can be only satisfied for those chromosomes that have cycles and a subpath or just a Hamiltonian path. We will allow a period between the beginning of the genetic algorithm and the appearance of this type of chromosome as the first evolutionary stage.

Once this quality among individuals is reached in the population it is then convenient to verify by the sufficient condition. This condition is given by a measure of the length of the path and of the cycles that are codified in the chromosome.

We suppose that we have p cycles and each one connects $t_1,...t_p$ vertices respectively and the subpath has the length l, then
$F = 500 + t_1 (t_1 - 1) + ... + t_p (t_p - 1) + l (l - 1)$, as $t_1 + ... + t_p + l = k$, it can be seen that the maximum is reached for $l = k$.

A second evolutionary stage starts when the first chromosome appears that fulfills the necessary condition. This stage is characterized by the both conditions checking simultaneously.

The evolutionary objective function was introduced in a natural way at the start of the GA, the graph generated by chromosomes are very complex. Many vertices could have a degree greater than two. This gave us the possibility of evaluating the first stage with the necessary condition and afterwards a second stage with sufficient condition. In this way the computational cost of the objective function is reduced.

3.3 The Genetic Hamiltonian Path Algorithm (GHPA)

In this section we describe all the components of our hybrid genetic algorithm for the Hamiltonian path problem.

1 Create the first population following one of the two initialization schemes.

2 While (stopping condition is not met)
3 Determine the Fitness of the population's members
4 Select two parents (P1 and P2) using the proportional selection scheme
5 Crossing P1 and P2 the two offspring (H1 and H2) are created and incorporated to the next population.

6 Apply the intelligent mutation operator to H1 and H2. (Local optimization procedure.)
7 Update the stopping condition
8 Return the set of the best members of the population

Preprocessing .- The circular vectors that integrate the vertex set are generated using the algorithm presented in [Santana and Ponce de León,1997]. The adjacency list for each vertex is formed applying the adjacency relation for circular vectors defined in 2.2. These adjacency lists store all the information needed for the evaluation of the fitness function.

Initial population.- We tested two different initialization schemes. In the first, chromosomes are randomly generated without considering any information relating the

graph. In the second, subpaths are randomly constructed using the information about nodes and edges and then seeded in the population .

Selection operator.- In our GA each bitstring is chosen with a probability proportional to its fitness.

Restricted edge crossover operator.- This crossover follows from the fact that offspring should preserve the edges that are in both its parents and they should not have those that are not present in either of its parents. On the sites where the values of both parents coincide the offspring will keep this value. In the other case (When parents sites have different values) one of the offspring is chosen randomly and it is assigned 1 or 0 values alternatively.

Example:

C1 = 1 0 0 1 0 0 1 1 1 0 1 0 1 1 0 1
C2 = 0 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1

The total edges is 16, the path is of 10 vertices, therefore 9 edges are set to 1. H1 was chosen as the starting chromosome and given alternating. The underlined values are the positions where the values do not coincide.

H1 = 1 0 0 1 0 1 1 0 1 0 1 1 1 0 0 1
H2 = 0 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1

It can be seen that this crossover guarantees that the restriction is adhered to, since both offspring have the same number sites for each value.

Intelligent Mutation.- To speed up the convergence a local optimization step is used. We defined it as an Intelligent Mutation. Thus we have a hybrid genetic algorithm .

In a similar way to that of the fitness function, one vector $W=(w_1,...,wn)$ is formed, containing each vertex's degree. For those vectors that don't reach the necessary condition defined in the fitness function, the next steps are followed:

1) One component (wi) is randomly chosen among those wx ($1 \leq x \leq k$) for which $wx<2$.
2) Another component (wj) is randomly chosen among those wx for which $wx>2$.
3) One edge adjacent to i is incorporated to the edges set represented in the chromosome.
4) One edge adjacent to j is extracted from the edges set represented in the chromosome.

In this way, the intelligent mutation makes a local optimization. This helps the chromosomes to reach a configuration nearer to that of the necessary condition. If the chromosome already satisfies the necessary condition but not obey the sufficient condition, the intelligent mutation works the following way: the components wi and wj are chosen such that wi=1, wj=2, and the same previous steps 3 and 4 are executed.

Stopping Condition.- For this problem the optimum fitness value is a priori known. This fact lets the algorithm stop when the optimum value is reached. or when it has been executed a previous defined number of steps.

Software.- All software needed for this study was programmed in the computer language C++. The GA was programmed using the software library GAL [Ochoa et al, 1995], comprising domain-independent routines.

4. Experimental results

Table #1 shows the data of the different graphs $H_{i+1,b}$ in which the algorithm was tested. The search complexity depends not just on the number of edges and vertices but also in the degree of the vertices. In general, vertices with lower degree add constraints to the search problem.

The run times for the graphs $H_{6,2}$, $H_{6,3}$, $H_{6,4}$, $H_{7,2}$, $H_{7,5}$, $H_{8,2}$, $H_{8,6}$ were less than 10 seconds. The program was implemented in C++ language and ran on a GATEWAY 486 DX, 66 mHz.

In every run the GA may find different solutions. If after a certain number of runs the Hamiltonian path has not been found, we suspect the non existence of such a path. In theses cases two neighbors levels of the graph H_{i+1} are joined and the algorithm tried to find a path in this new graph.

GRAPH $H_{I+1,B}$	$\lvert V_{I+1,B} \rvert$	HIGHER DEGREE	VERTEX WITH DEGREE 2	$\lvert E_{I+1,B} \rvert$	HAMILTON PATH
$H_{6,2}$	10	4	5	15	
$H_{6,3}$	20	6	6	35	
$H_{6,4}$	10	4	5	15	
$H_{7,2}$	15	4	5	24	
$H_{7,3}$	50	6	10	99	
$H_{7,4}$	50	6	10	99	
$H_{7,5}$	15	4	5	24	No
$H_{8,2}$	21	4	5	35	
$H_{8,3}$	105	6	15	224	
$H_{8,4}$	175	8	20	399	
$H_{8,5}$	105	6	15	223	
$H_{8,6}$	21	4	5	35	No

Table #1.- Summary of the data of the different graphs $H_{i+1,b}$ in which the algorithm was tested.

Figure 1 shows different run of the GHPA plotting the fitness of the best objects as a function of the generation count. The values shown are based on the average of 40 runs, each with a different seed for random subpaths.

Common parameters for all runs:

- Seeding of the initial population
- Number of generation
- Adaptative mutation rate
- Population size
- Restricted edge crossover operator

In figure 1 the influence of the crossover rate is documented. Table #2 shows the relations between crossover rate, times and percent to reach the optimum in the 40 runs.

The GHPA with the above mentioned parameters and crossover rate of 0.7 obtained the best performance. This combination in GHPA was used to obtain the Hamiltonian paths of the graphs H_6, H_7 and H_8.

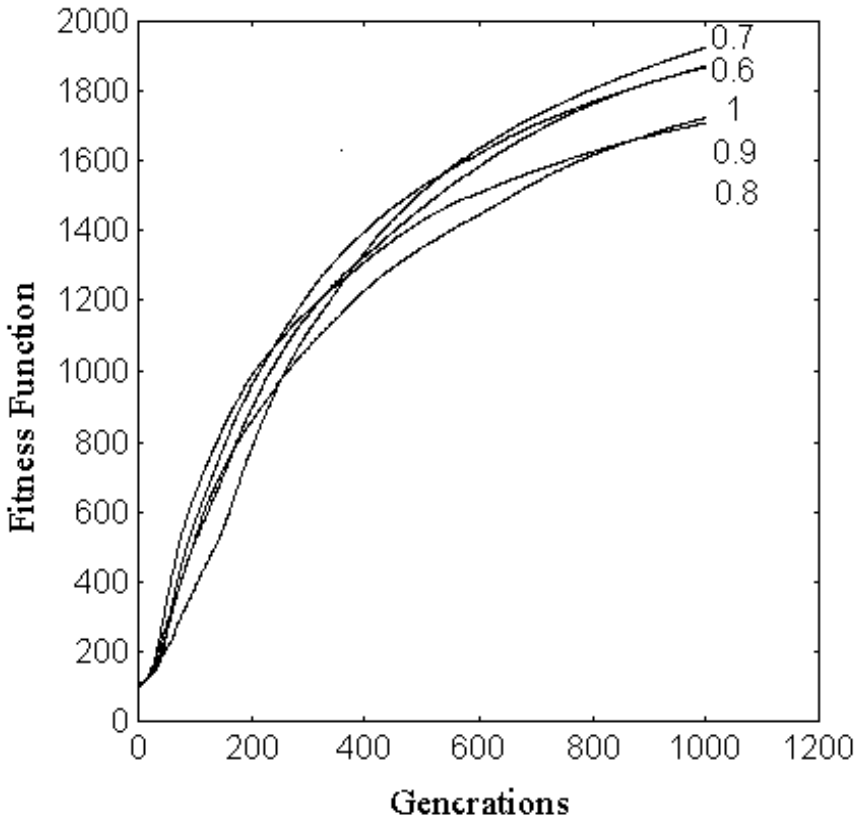


Figure 1.- The influence of the crossover rate in GHPA.

Crossover rate	Times to reach the optimum	Per cent of achievement
0.6	28	70
0.7	30	75
0.8	19	47.5
0.9	21	52.5
1	29	72.5

Table #2.- The relations between crossover rate, times and percent to reach the optimum.

CONCLUSIONS

The formulation of the problem of finding a Hamiltonian path in multi-level graphs is presented. A new mutation operator that improve previous results is described. The GA was used to obtain the Hamiltonian paths of the graphs H_6, H_7 and H_8. The experimental results reveal the efficacy of the GA paradigm for the important problem of finding a Hamiltonian path in very complex graphs.

References

DE JONG, K. A. (1985): Genetic algorithms: a 10 years perspective. **Proc. First Inter. Conf. on Genetic Algorithms and their Applications**. Erlbaum.

DE JONG, K. A. AND SPEARS, W.M. (1989): Using genetic algorithms to solve NP-complete problems**. Proc. Third Inter. Conf. on Genetic Algorithms and their Applications**, pp. 124-132. Kaufmann.

DEJTER, I.J. (1985): Hamilton cycles and quotients of bipartite graphs, in Y. Alavi et al. Eds., ***Graph theory and its applications to algorithms and computer science*** (J. Wiley, New York). 189-199.

DEJTER, I.J. , CORDOVA, J. QUINTANA (1988) : "Two Hamilton cycles in bipartite reflective kneser graphs", ***Discrete Matematics*** *72*. 63-70.

DEJTER, I.J. Y QUINTANA, J. (1991): On an extension of a conjeture of I. Havel en Y. Alavi et al. Eds., ***Graph theory , Combinatorics and Applications,*** Two volume set. John Wiley and Sons, Inc.

DEJTER, I. J.; CEDEÑO, W. Y JAUREGUI, V. (1993): A note Frucht diagrams, Boolean graphs and Hamilton cycles. ***Discrete Mathematics*** *114.* 131-135. North- Holland.

FOX, B.R. AND MCMAHON, M.B. (1991): Genetic operators for sequencing problems. in **Foundations of Genetic Algorithms** (edited By G.J.E. Rawling)., pp. 284-300. Kaufmann.

GARAY AND JONHSON, D.S. (1979): **Computers and intractability. A guide to the theory of NP- completeness.** W. H. Freeman and Company, New york.

GOLDBERG, D. E. (1989): **Genetic algorithms in search, optimization, and machine learning.** Reading, M.A. Addison - Wesley.

GOLDBERG, D. E. AND LINGLE, R. (1985): Alleles, loci and the traveling salesman problem. **Proc. First Int. Conf. Genetic Algorithms and their Applications**, pp.154-159. Erlbaum.

GREFENSTETTE, J.J; GOPAL, R.; ROSMAITA, B. AND VAN GUCHT, D.(1985): Genetic algorithms for the traveling salesman problem**. Proc. First Int. Conf. Genetic Algorithms and their Applications,** pp. 160-168. Erlbaum.

MICHALEWICZ, Z. (1992**): Genetic algorithms + data structure = evolutionary programs**, Springer- Verlag, Berlin.

OCHOA, A.; ROSSETE, A. AND PONCE DE LEÓN, E. (1995): Programación de un algoritmo genetico: un primer paso para la solución del problema. **Revista Ingeniería, Electrónica, Automática y Telecomunicaciones.** No. 2. Facultad Ingeniería Eléctrica. ISPJAE.

OLIVER, I. M., SMITH, D. J. AND HOLLAND, J. R. C. (1987): A study of permutation crossover operators on the traveling salesman problem. **Proc. Second Int. Conf. on Genetic Algorithms and their Applications**, pp 224-230. Erlbaum.

PONCE DE LEÓN, E.; OCHOA, A. AND SANTANA, R. (1995): A genetic algorithm for a Hamiltonian path problem. **Reporte de Investigación ICIMAF**, 95 - 01, CENIA 95 - 1. ISSN 0138 - 8916.

PONCE DE LEÓN, E.(1997): La conjetura de Erdös y Havel: acciones de grupo, vectores circulares y grafos multiniveles. **Reporte de Investigación ICIMAF**, 97 - 39, CENIA 97 - 02. ISSN 0138 - 8916.

POON, P.W. AND CARTER, J.N. (1995): Genetic algorithm crossover operators for ordering applications. **Computers Ops Res.** Vol 22, No.1, pp. 135-147.

SANTANA, R. AND PONCE DE LEÓN, E. (1996): Algoritmos genéticos para un problema de busqueda de caminos Hamiltonianos. **Tesis de diploma**. Facultad de Matemática. Universidad de la Habana.

SYSWERDA, G. (1991): Schedule optimization using genetic algorithms. In **Handbook of Genetic Algorithms** (edited by L. Davis), pp. 332-349. Van Nostrand Reinhold, Amsterdam.

WHITLEY, D., STARKWEATHER, T. AND FUQUAY, D. (1989): Scheduling problems

and the traveling salesman: The genetic edge recombination operator. Proc. Third Int. Conf. on Genetic Algorithms and their Applications, pp. 133-140. Kaufmann.