Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

A Genetic Algorithm Based on Stochastic Crossover for DHCP

Marco Carpentieri

Abstract– We introduce a genetic model based on stochastic crossover to solve the Hamiltonian cycle problem (DHCP) for random digraphs containing a random Hamiltonian cycle. The genetic model represents a new decision computational method inspired by the remark that DHCP can be formulated as determining the compatibility of a quadratic system over the finite field GF(2). A (simple) genetic algorithm based on the stochastic crossover is experimentally compared with a randomized algorithm based on the Angulin and Valiant classic technique designed to find Hamiltonian cycles in random digraphs.

I. Introduction

Genetic algorithms (GAs) are probabilistic search algorithms inspired by mechanisms of natural selection and genetics, introduced by John Holland in the 1970*s*. They have received considerable attention because of their many applications to several research fields such as optimization, adaptive control and others [23], [28], [29], [34].

According to the classic genetic algorithm theory, the fittest individuals chromosomes are formed by merging short definition length and small specificity order allele schemata, whose fitness remains above the average fitness of the populations generated by the genetic cycles (Holland [23], [28], [29]). This central result has introduced the concept of separability of the fitness functions, with respect to short chromosomic traits, that is recognized [14], [42], [46] as a basic property required to justify the application of a genetic algorithm. More recently, a new class of marginal distribution genetic algorithms is appeared in the literature [3], [4], [11], [12], [53]. Such new algorithms, based on models related with the classic ones, consent to perform (state transition) efficient implementation for the associated infinite population genetic systems; the analysis of the marginal distribution genetic systems has shown important analogies with a class of local optimizers in which Hopfield Networks [32] are included. Note it is well known that Hopfield networks can be used to provide approximated solutions for hard optimization problems [31], but there are also algorithmic techniques that exhibit better theoretical error bounds and better experimental performances (for example [25]).

The classic genetic algorithms (and those based on marginal distribution models) do not seem to be competitive with the specialized efficient optimization techniques [2], [6], [20], [25], [52] designed to solve specific hard problems [13], [31]. What seems to make difficult the genetic algorithms approach is that the properties of separability and statistic independence required (the second one in marginal distribution *GAs*) by the classic genetic computation are not satisfied. This problem is also connected with that of efficient implementation in case of infinite populations since exponential complexity seems to be required to represent the states, and consequently the dynamics computation time of the genetic systems. Such remarks motivate the need of designing new genetic models (probably representing approximations of the true ones) aimed to provide efficient computation methods to solve hard optimization problems. Related work, connected with the more recent development of marginal distribution *GAs*, can be found in the literature of the Estimation of Distribution Algorithms; for an introduction the reader is referred to [38], [40], [45], [58], [59].

A different approach adopted to apply genetic algorithms to hard optimization problems has been introduced in the 1980s considering chromosomes encoding permutations of $l-$ary alleles ($l \in \mathbf{N}$) instead of arbitrary binary strings. In this setting, many authors studied the problem of designing effective genetic operators (depending on the considered specific problem) able to exchange chromosomic traits preserving the permutation structure [7], [10], [15], [16], [17], [18], [19], [24], [26], [27], [30], [33], [35], [36], [39], [41], [43], [48], [49], [50], [51], [55], [56]. Reviewing the great amount of work in the literature, the reader shall find that several researchers have recognized that useful operators for problems such as the (Directed) Hamiltonian Cycle Problem (DHCP) [31] or the Traveling Salesman Problem *TSP* [13] do not have to preserve order or positions as in the classic *GAs*, but connections between consecutive alleles in the chromosomes. In spite of all these efforts, as the author is aware, it is not clear neither whether the genetic algorithms could represent (efficient) optimization techniques competitive with the (known) best classic methods whose performance can be theoretically estimated (for example [2], [8], [20], [25], [52]), nor whether the genetic algorithm paradigm could be really used to improve the results obtained by such (more specialized) techniques.

In this paper we introduce a genetic model based on stochastic crossover to solve (the $NP-$complete problem [31]) *DHCP* for $p-$random digraphs [2], [6], [13], [20], [31], [47], [52] containing a (superposed) random Hamiltonian cycle. This class of graphs is considered both since is a natural extension of the random graphs and (mainly) since in case of small edge densities $\left(p = O\left(\frac{\log l}{l}\right)\right)$, the classic techniques, such as those presented in [2], [20], are unsuccessful (other basic motivations can be found in [8]). The genetic model we present is characterized by the non-classic properties that selection and replacement are

Manuscript received October, 29, 2006.

Marco Carpentieri is with the Dipartimento di Fisica E. R. Caianiello (DF), Università degli Studi di Salerno (Sa), Italy

deterministic processes, while the genetic recombination is performed by a stochastic binary operator whose action can also depend on specific properties of the input instances (digraphs). The definition of the (crossover) operator is quite flexible and includes the possibility of incorporating other unary operators such as those used to find as long as possible simple paths. One of the main differences exhibited by the model with respect to the classic genetic ones is that the chromosomes encode permutations (of vertices). In such a context, as we have already discussed above, a (binary) recombination operator exchanging chromosome traits has to complete them preserving the permutation structure. A specific implementation of the offspring chromosomes completion is designed following two main principles:

1) the connections between consecutive vertices in the parent whose first chromosomic segment is superposed by the corresponding trait of the other parent in the child copy have to be preserved;
2) when an allele is set in the chromosome complement of a child and the connections (between consecutive vertices) in the parent cannot be preserved, the next allele is selected according to a stochastic greedy choice that depends on a small depth deterministic simulation of a nondeterministic visit of a suitable subgraph.

The completion process (after the exchange of the two initial segments of the parents chromosomes) is inspired by some remarks about the algebraic nature of $DHCP$. In fact, we show that the Hamiltonian cycle problem can be stated as the problem of determining the compatibility of some quadratic systems over the finite field $GF(2)$. Such formal model can be considered as alternative to the other ones such as Nondeterministic Turing Machines [31], Verification Algorithms [13] and Integer Quadratic Optimization Programs [25] that more frequently can be found in the literature. Although our remark is quite straightforward, formulating $DHCP$ in such a way suggests a new computational scheme to try to solve the problem based on searching for, by subsequent approximations, a permutation transforming the input instances into isomorphic graphs containing a fixed unit Hamiltonian cycle. We shall introduce two distinct genetic algorithms; the first one is based on the previously described implementation of the binary stochastic crossover, while the other on Angulin and Valiant heuristic [2]. Note that the results provided by Angulin and Valiant in [2] regard random digraphs and are theoretical. We provide an experimental evaluation of the performance of their algorithm (even if in the context of a more general schema). An experimental comparison between the two genetic algorithms is performed.

II. Preliminaries

A. Notation

In the rest of the paper we shall adopt the following notation. Let $G = (V, E)$ be a directed graph (digraph) with vertex set $V = \{1, \ldots, l\}$, edge set $E \subseteq \{(\omega, \omega') : \omega \neq \omega' \text{ and } \omega, \omega' \in V\}$ and adjacency matrix $W = [w_{i,j}]_{1, \leq i,j \leq l}$. A simple ([13]) path from vertex ω to vertex ω' is an ordered sequence $(\omega_1, \ldots, \omega_e)$ $(e \geq 2)$ such that $(\omega_i, \omega_{i+1}) \in E$, $\omega_i \neq \omega_j$ for $i \neq j$ $(i, j = 1, \ldots, e)$ and $\omega = \omega_1, \omega' = \omega_e$. The path is Hamiltonian if $e = l$ and, in such a case, if there exists the edge $(\omega_l, \omega_1) \in E$ we shall also say that $\underline{\omega}$ is (or represents) a Hamiltonian cycle. Note that if $\underline{\omega}$ is Hamiltonian $(e = l)$, then such sequence of l vertices denote also permutation $\underline{\omega} : V \rightarrow V$ defined by $\underline{\omega}(i) = \omega_i$ for $i = 1, \ldots, l$. Throughout we shall mean by $\underline{\omega}$, by sake of conciseness and with abuse of notation, paths, cycles and permutations. The meaning of the $\underline{\omega}-$notation shall be clear from the context in which the symbol appears. By $\underline{\omega}(G)$ we shall mean the graph isomorphic $(\underline{\omega}(G) \approx G)$ to G obtained by relabeling the vertices in G as specified by $\underline{\omega}$. The Hamiltonian cycle (or Hamilton circuit) problem [31] consists in, given a digraph G, deciding whether there exists a Hamiltonian path $\underline{\omega}$ that visits each vertex in V exactly once and returns to its starting point.

B. Background

The Hamiltonian Cycle problem for random directed graphs (digraphs) [31], denoted by $DHCP$, is a $NP-$complete (decision) problem that has received considerable attention in the literature due to its practical applications and to its computational structural complexity properties. Recently, a great deal of work has been dedicated to consider the problem of finding efficient solution algorithms in case of random (di)graphs. One of the first results about Hamiltonian cycles in (random) digraphs is due to Perepelica [44] who introduced randomized procedures of time complexity $O(l^2)$ for almost certainly [2] finding Hamiltonian cycles in digraphs with at least $cl^{\frac{3}{2}}\sqrt{\log l}$ edges, where c is a sufficiently large constant (throughout the paper by l we shall denote the number of vertices of the digraphs). Wright [57] gave a non-algorithmic proof of the fact that almost all (random) digraph with $\underline{\mathrm{O}}(l)l^{\frac{3}{2}}$ edges have a Hamiltonian cycle, where by $\underline{\mathrm{O}}(l)$ we mean any function such that $\underline{\mathrm{O}}(l) \rightarrow \infty$ as $l \rightarrow \infty$. Angulin and Valiant [2] proposed a polynomial algorithm, of time complexity $O(l(\log l)^2)$, for almost certainly finding a Hamiltonian cycle in a (random) digraph with (at least) $cl \log l$ edges (in this notation c is also an unspecified sufficiently large constant). Other work about Hamiltonian cycles in digraphs is due to McDiarmid [37] who presented a non-constructive proof that $\lim_{l \rightarrow \infty} Pr\,(DG_L \text{ is Hamiltonian }) = 1$ for digraphs with $L = l \log l + lc_l$ edges and $c_l - \log\log l \rightarrow \infty$. Frieze [20] showed that for digraph in DG_L with $L = l \log l + lc_l$ edges there exists a $O(l^{1.5})$ polynomial randomized algorithm to find Hamiltonian cycles with probability $e^{-2e^{-c}}$ if $c_l \rightarrow c$ and with probability 1 if $c_l \rightarrow \infty$ (as $l \rightarrow \infty$).

C. Angulin and Valiant Heuristic

Angulin and Valiant procedure, denoted DHC, tries to construct as long as possible simple paths $\underline{\omega}$ starting from an initial random vertex ω_s. Let ω' be the endpoint of $\underline{\omega}$; the path is constructed by iteratively adding random vertices adjacent to the endpoint ω'. Each time a new random vertex ω, adjacent to ω', is selected, the corresponding edge (ω', ω) is deleted by the input digraph G. In the case in which ω is already in $\underline{\omega}$ and it is sufficiently far from the endpoint ω', a rotational transformation is applied to transform $\underline{\omega}$ into a new subpath $\underline{\omega}'$ and a cycle $\underline{\omega}''$, otherwise ω is simply discarded. The rotational transformation computes the new path $\underline{\omega}'$ as the subpath in $\underline{\omega}$ starting from ω_s and reaching the predecessor of ω in $\underline{\omega}$, while the cycle $\underline{\omega}''$ is individuated by the subpath (in $\underline{\omega}$) starting from ω and ending in ω'. The endpoint ω' is updated becoming the last vertex of the path $\underline{\omega}'$. After the application of the transformation, new random vertices ω (adjacent to the endpoint ω'), being neither in $\underline{\omega}'$ nor in $\underline{\omega}''$, are appended to $\underline{\omega}'$ (along with a consequent updating of ω'). When a random vertex is found to be in $\underline{\omega}'$, it is simply discarded, while if is in $\underline{\omega}''$ a new unique path is composed from $\underline{\omega}'$ and $\underline{\omega}''$ applying a second rotational transformation. This new transformation is implemented joining the endpoint of $\underline{\omega}'$ to the predecessor of ω in $\underline{\omega}''$ so constructing a new unique path $\underline{\omega}$ (formed by the vertices in $\underline{\omega}'$ and in $\underline{\omega}''$). The path $\underline{\omega}'$ and the cycle $\underline{\omega}''$ are discarded. Thus, the procedure continues to add random vertices ω to the path $\underline{\omega}$ (deleting the edges (ω', ω) from the input graph) alternating the application of the two rotational transformations, when required, until either a Hamiltonian cycle is found or an endpoint ω' is reached that does not have adjacent vertices (in such a case DHC fails).

An example illustrating how the Angulin and Valiant procedure operates is displayed in Figure 1. First, we see a simple path $\underline{\omega}$ from ω_s to ω'. A random vertex ω adjacent to ω' is selected; this vertex ω belongs to $\underline{\omega}$ and is far more than $\frac{l}{2} - 2$ vertices from ω' (suppose $l = 7$). Thus, the path $\underline{\omega}$ is split into a subpath $\underline{\omega}'$ and a cycle $\underline{\omega}''$. After that a new random vertex ω adjacent to the (updated) endpoint ω' of $\underline{\omega}'$ is selected; since ω is neither in $\underline{\omega}'$ nor $\underline{\omega}''$, is appended to $\underline{\omega}'$ (updating the new endpoint). A new random vertex ω (adjacent to ω') is selected, is in $\underline{\omega}''$ and a unique path from ω_s to the predecessor of ω in $\underline{\omega}''$ is composed.

By using the intertranslation conditions provided by Angulin and Valiant, their main result ([2], pag. 156) may be stated as follows.

Theorem 1: [2] If $p \geq c\frac{\log l}{l-1}$, then DHC is a $O(l(\log l)^2)$ polynomial randomized procedure finding a Hamiltonian cycle with probability $1 - O(l^{-a})$ ($a > 0$ constant) in a graph taken from DG_p, for (sufficiently large) $l \in \mathbf{N}$ and where $c \in \mathbf{R}^+$ is a sufficiently large constant.

III. Genetic Model

Denote by

$$\phi_G \quad : \quad \Omega_l \rightarrow [0, 1]$$

a probability distribution over the set $\Omega_l = \{\underline{\omega}_1, \ldots, \underline{\omega}_{l!}\}$ of chromosomes that are permutations of the l vertices in V. We shall adopt notation $\underline{\omega}_i = (\omega_{i,1}, \ldots, \omega_{i,l})$ to represent the chromosome $\underline{\omega}_i \in \Omega_l$ for $i = 1, \ldots, l!$. The fitness function is

$$f \quad : \quad \Omega_l \rightarrow \mathbf{N},$$

defined by

$$f(\underline{\omega}_i) \quad = \quad \sum_{j=1}^{l-1} w_{\omega_{i,j}, \omega_{i,j+1}} + w_{\omega_{i,l}, \omega_{i,1}} \; (1 \leq i \leq l!).$$

A population P is represented by a multi-set $\{\underline{\omega}_{i_1}, \ldots, \underline{\omega}_{i_n}\}$ of n chromosomes in Ω_l. In the following, we shall suppose that the multi-sets $P = \{\underline{\omega}_{i_1}, \ldots, \underline{\omega}_{i_n}\}$ (= is used with abuse of notation) are such that

1) $< \omega_{i_1,1}, \ldots, \omega_{i_1,k} >, \ldots, < \omega_{i_n,1}, \ldots, \omega_{i_n,k} >$ include all distinct subpaths of length k in G for some $k \in \mathbf{N}$ ($k < l$).

Property 1) is intended to let the genetic system take advantage from the information got by an efficient deterministic visit of the paths in the graph G simulating a $k-$time nondeterministic procedure to find a Hamiltonian cycle (by efficient we mean that it is required $n = O(l^c)$ for some suitable constant $c \in \mathbf{R}^+$). The crossover of two chromosomes $\underline{\omega}_i, \underline{\omega}_j$ ($1 \leq i, j \leq l!$) is a (stochastic) binary operator

$$K_{\underline{\omega}_i, \underline{\omega}_j} \quad : \quad \Omega_l^2 \rightarrow [0, 1]$$

whose definition depends on the probability distribution $\phi_G()$. If the population at time t is P, then the next population is generated applying the following sequence of actions:

1) select a pair $(\underline{\omega}_{i_r}, \underline{\omega}_{i_s})$ of chromosomes ($1 \leq r, s \leq n$);
2) perform the crossover producing a pair of children $(\underline{\omega}'_{i_r}, \underline{\omega}'_{i_s})$ with probability $K_{\underline{\omega}_{i_r}, \underline{\omega}_{i_s}}(\underline{\omega}'_{i_r}, \underline{\omega}'_{i_s})$;
3) replace $\underline{\omega}_{i_r}$ with $\underline{\omega}'_{i_r}$ only if it holds that $f(\underline{\omega}'_{i_r}) > f(\underline{\omega}_{i_r})$;
4) replace $\underline{\omega}_{i_s}$ with $\underline{\omega}'_{i_s}$ only if it holds that $f(\underline{\omega}'_{i_s}) > f(\underline{\omega}_{i_s})$.

The selection-crossover-replacement cycle described by the steps $1) - 4)$ for $r, s = 1, \ldots, n$ is repeated until some halting condition is satisfied. Note that the model differs from the classic genetic schemes since selection and replacement are deterministic processes, while the crossover is a stochastic operator.

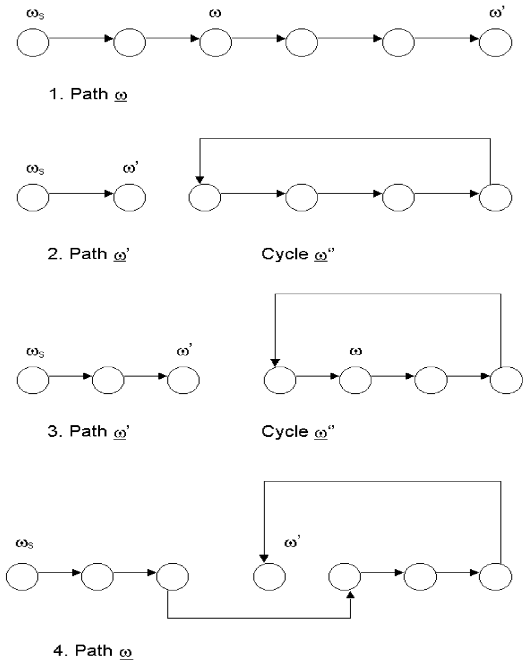

Fig. 1. DHC: Example.

IV. *DHCP*

Let $\underline{\omega} = (\omega_1, \ldots, \omega_l)$ be a Hamiltonian cycle in G. Given any permutation $\underline{\omega}'$ of the l vertices in the vertex set V of G, the isomorphic graph $\underline{\omega}'(G)$, has the cycle $\underline{\omega} \circ \underline{\omega}'$ (where by $\circ$ we denote the usual composition of permutations). Note that there always exists a permutation $\underline{\omega}'$ producing a new graph $\underline{\omega}'(G) \approx G$ in which $\underline{\omega} \circ \underline{\omega}'$ is the unit Hamiltonian cycle $\mathbf{1} = (1, \ldots, l)$. Consequently, the Hamiltonian cycle problem can be reformulated as:

- decide whether there exists a permutation $\underline{\omega}'$ of the vertices in V such that the graph $\underline{\omega}'(G)$ isomorphic to G has the Hamiltonian cycle $\mathbf{1}$.

Any permutation $\underline{\omega}'$ transforming a Hamiltonian cycle $\underline{\omega}$ into the cycle $\underline{\omega} \circ \underline{\omega}' = \mathbf{1}$ generates a new isomorphic graph $\underline{\omega}'(G) \approx G$. Consequently, for each permutation $\underline{\omega}'$ of vertices such that $\underline{\omega} \circ \underline{\omega}' = \mathbf{1}$ we have it holds that:

$$X_{\underline{\omega}'} W X_{\underline{\omega}'}^T \quad = \quad W'_{\underline{\omega}'(G)},$$

where by $X_{\underline{\omega}'}$ we denote the permutation matrix defined by $x_{\omega'_j, j} = 1$ for $1 \leq j \leq l$, zero otherwise and $W_{\underline{\omega}'(G)}$ is the adjacency matrix of the graph $\underline{\omega}'(G)$. Since the adjacency matrix $W = [w_{i,j}]_{1 \leq i,j, \leq l}$ of G is known, each permutation $\underline{\omega}'$ univocally individuates the adjacency matrix $W'_{\underline{\omega}'(G)}$ associated to the isomorphic graph $\underline{\omega}'(G)$; thus, the Hamiltonian cycle problem can be stated as

- determine the compatibility of the quadratic system
  $$XWX^T \quad = \quad W', \tag{1}$$
  with (unknown) solutions (X, W') being $X = [x_{i,j}]_{1 \leq i,j, \leq l}$ a $l-$order permutation matrix and W' a partially specified adjacency matrix having entries $w'_{1,2} = w'_{2,3} = \ldots = w'_{l-1,l} = w'_{l,1} = 1$ and all other entries unknown in $\{0, 1\}$.

The following lemma states that for each Hamiltonian cycle in a digraph G there are exactly l distinct permutations transforming it into the Hamiltonian cycle $\mathbf{1}$; moreover, if $\underline{\omega}'', \underline{\omega}'''$ are two permutations transforming the cycles $\underline{\omega}, \underline{\omega}'$ into the cycle $\underline{\omega} \circ \underline{\omega}'' = \underline{\omega}' \circ \underline{\omega}''' = \mathbf{1}$ respectively, where $\underline{\omega}$ and $\underline{\omega}'$ are distinct, then $\underline{\omega}''$ and $\underline{\omega}'''$ are also distinct.

Lemma 1: For every cycle $\underline{\omega}$ there are exactly l distinct permutations $\underline{\omega}_{i_r}$ $(1 \leq r \leq l)$ such that $\underline{\omega} \circ \underline{\omega}_{i_1}, \ldots, \underline{\omega} \circ \underline{\omega}_{i_l}$ represent the Hamiltonian cycle $\mathbf{1}$ in the corresponding isomorphic graphs $\underline{\omega}_{i_1}(G), \ldots, \underline{\omega}_{i_l}(G)$; moreover, if $\underline{\omega}$ and $\underline{\omega}'$ are distinct cycles, then there not exists a permutation $\underline{\omega}''$ such that $\underline{\omega} \circ \underline{\omega}''$ and $\underline{\omega}' \circ \underline{\omega}''$ represent the same cycle $\mathbf{1}$.

By the previous lemma we get that, for every Hamiltonian cycle $\underline{\omega}$ in G, there exist exactly l distinct solutions (permutations) of System (1) that map $\underline{\omega}$ in the unit cycle $\mathbf{1}$. In this regard note that there exist exactly $l!$ permutations X and it is well known that the number of possible Hamiltonian cycles in graphs with l vertices is $(l-1)!$.

V. Procedure SSC

Reformulating *DHCP* as an algebraic problem provides the basic insights to design and interpret a suitable instance of the genetic model described in Section III. In particular, the polynomial time reduction of *DHCP* to the problem of deciding the compatibility of some system (1) suggests to solve it by searching for isomorphisms mapping the input digraph G into another graph containing the unit cycle (or however some other fixed Hamiltonian cycle). Since we are interested in the performance of the genetic algorithm (simulating the model) for several $k-$values and the simulation time increases at least as the number of $k-$length paths in G, we require as quick as possible convergence to the fittest individual. The basic procedure implements a specific (simplified, slightly adapted) instance of the genetic model with stochastic crossover. In such instance, a maximum fitness chromosome $\underline{\omega}_{i_r}$ is initially selected and the crossover of its copy $\underline{\omega}$ is performed with the copy of each other chromosome $\underline{\omega}_{i_s}$ $(1 \leq s \leq n)$ until a better child $\underline{\omega}'_{i_s}$ such that $f(\underline{\omega}'_{i_s}) > f(\underline{\omega}_{i_r})$ is found. In such a case, $\underline{\omega}_{i_s}$ is replaced with $\underline{\omega}'_{i_s}$ and the procedure is (recursively) applied to the maximum fitness chromosome (now $\underline{\omega}_{i_s}$) until either a Hamiltonian cycle is found or the fitness of the fittest chromosome cannot further on be improved. The stochastic crossover is implemented as follows. The child $\underline{\omega}'_{i_s}$ is obtained from the parents $\underline{\omega}_{i_r}$ and $\underline{\omega}_{i_s}$ first copying the first k alleles of $\underline{\omega}_{i_s}$ into (the same loci of) $\underline{\omega}'_{i_s}$. After that, if, in forming the rest of the $\underline{\omega}'_{i_s}$ chromosomic complement, we set an allele $\omega'_{i_s,j} = \omega_u$ $(j \geq k)$ that is connected to its successor $\omega_{u \bmod l+1}$ in the sequence specified by $\underline{\omega}_{i_r}$ (and it holds that $\omega_{u \bmod l+1} \notin \{\omega'_{i_s,1}, \ldots, \omega'_{i_s,j}\}$), the next allele $\omega'_{i_s,j+1}$ is set to $\omega_{u \bmod l+1}$; if, instead, the allele $\omega'_{i_s,j}$ is not connected to its successor in the sequence specified by $\underline{\omega}_{i_r}$ (or the successor is in $\{\omega'_{i_s,1}, \ldots, \omega'_{i_s,j}\}$), the next allele $\omega'_{i_s,j+1}$ is selected according to a (stochastic) greedy choice stating that $\omega'_{i_s,j+1}$ is a random vertex ω adjacent to $\omega'_{i_s,j}$ with minimum nonzero out-degree in the restriction $G_{V-\{\omega'_{i_s,1}, \ldots, \omega'_{i_s,j}\}}$ of the graph G to the vertex set $V - \{\omega'_{i_s,1}, \ldots, \omega'_{i_s,j}\}$; if such a vertex does not exist, $\omega'_{i_s,j+1}$ is set to a random vertex in $V - \{\omega'_{i_s,1}, \ldots, \omega'_{i_s,j}\}$. Note that such a kind of choice is based on the idea of searching for information about a new vertex to select by performing a two-level deterministic simulation of a nondeterministic procedure designed to track all simple paths that can be formed starting from $\omega'_{i_s,j}$ visiting only vertices in $V - \{\omega'_{i_s,1}, \ldots, \omega'_{i_s,j}\}$. Such kind of criterion is similar to that used in [9] but that is dynamically predetermined by the already assigned vertices $\omega'_{i_s,1}, \ldots, \omega'_{i_s,j}$. Moreover, the reader could recognize in the completion process adopted in implementing the stochastic crossover a variant of the Edge Recombination Operator introduced by Whitley et al. in [55], [56] (see also [16], [35], [48], [50]). The genetic procedure is named *SSC* and is reported in Figure 2.

Note that the procedure *SSC*, starting from a permutation $\underline{\omega}_{i_r}$, searches for a new offspring $\underline{\omega}'_{i_s}$ such that the graph ${\underline{\omega}'}^{-1}_{i_s}(G)$ has more edges in the set $\{(l, 1), (\omega, \omega + 1) \text{ for } \omega = 1, \ldots, l-1\}$ than in $\underline{\omega}^{-1}_{i_r}(G)$; the procedure tries to compute (or better complete) $\underline{\omega}'_{i_s}$ in such a way that ${\underline{\omega}'}^{-1}_{i_s}$ shifts simple subpaths along the sequence of vertices in the unit cycle $\mathbf{1}$ from $\underline{\omega}^{-1}_{i_r}(G)$ to ${\underline{\omega}'}^{-1}_{i_s}(G)$ (the shifts are interleaved adding new edges determined by means of an operator based on a small depth simulation of a nondeterministic visit of suitable restrictions of the input digraph). In other words, *SSC* searches for a permutation transforming G

```
Procedure $SSC(G,l)$
Set $\underline{\omega}$ to the fittest chromosome in $P$;
1. For each $k-$length path $(\omega'_1,\dots,\omega'_k)$ in $G$
    For $z=k+1$ to $l-1$
        If $\omega'_{z-1}=\omega_u$ And $(\omega_u,\omega_{u\mathrm{mod}l+1})\in E$
                And $\omega_{u\mathrm{mod}l+1}\neq\omega'_j$ for $j<z$ then
            $\omega'_z=\omega_{u\mathrm{mod}l+1}$;
        Else
            If there isn't a vertex in $G_{V-\{\omega'_1,\dots,\omega'_{z-1}\}}$
                    adjacent to $\omega'_{z-1}$ with minimum non-
                    zero out-degree then
                $\omega'_z=\omega$ where $\omega$ is random in $V-$
                    $\{\omega'_1,\dots,\omega'_{z-1}\}$;
            Else
                $\omega'_z=\omega$ where $\omega$ is a random minimum
                    nonzero out-degree vertex in $G_{V-}$
                    $\{\omega'_1,\dots,\omega'_{z-1}\}$ adjacent to $\omega'_{z-1}$;
    If $f(\underline{\omega}')>f(\underline{\omega})$ then
        $\underline{\omega}=\underline{\omega}'$; Go to 1.;
```

Fig. 2. Procedure SSC.

into another isomorphic graph containing the unit cycle $\mathbf{1}$ by subsequent approximations.

In Figure 4 an example of the way in which the stochastic crossover operates is displayed starting from the graph (in Figure) 3 and two initial chromosomes $\underline{\omega}_i=(1,2,4,3)$ and $\underline{\omega}_j=(1,4,3,2)$.

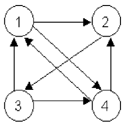

Fig. 3. A Hamiltonian Graph.

VI. Simulation

The heuristic in Section II-C can be quite straightforwardly incorporated in the genetic model and consequently a procedure, we shall name SAV, similar to SSC, can be designed by using the double rotation method suggested by Angulin and Valiant to complete the chromosomic structure of the offspring after the exchange of the alleles in the first k loci has been performed. However, in the case of the genetic model based on Angulin and Valiant method, the crossover completion does not depend on the parents chromosomic complement. This is due to the fact that the replacement operator substitutes parents having the same initial $k-$alleles of the offspring and since the Angulin and Valiant technique is basically a unary operator. As a consequence, the procedure SAV simply produces a random process in which all initial $k-$length paths of the input graph G are completed as explained in Section II-C (in other words the initial vertex ω_s is the endpoint of a short simple path). In the tables 1.1 − 2.4 an experimental comparison between the genetic algorithms $SAVGA$ based on SAV and $SSCGA$ (on SSC) is performed for undirected graphs (with $k=1$), where the (randomized) algorithms simply consist of independent repetitions of a (balanced) number (not less than l) of the corresponding procedure. Note that, taking into account of the fact that the population sizes depend also on the number of $k-$length paths in the input digraphs, the first success expected times, reported in bold in the tables, indicate how much large a population has to be, in average, in order to solve $DHCP$ (mostly with high confidence) for all $p-$values greater than some threshold p_{min} ($p-$values in the heading). Notation −− in the tables means that, for the considered p and l values, we are not able to provide suitable estimates with high confidence since the population sizes, and consequently the simulation time, become too large. Similar results can be observed in the more general case of directed graphs, but that the performances are slightly worse. Moreover, in the experiments with directed graphs we have noticed that, conversely to what happens in the case of undirected graphs, the simpler random selection of a next (adjacent) vertex can be used instead of the (stochastic) greedy choice in completing the offspring chromosomes without meaningful changes in the performances. A more general comparison between the two algorithms is displayed in Figure 5 in which the minimum $p-$values are reported (interpolated by least squares method against the input size l) such that the two algorithms (dotted plots for $SAVGA$) are estimated to solve $DHCP$ with high confidence (for $k=1,2,3$ and

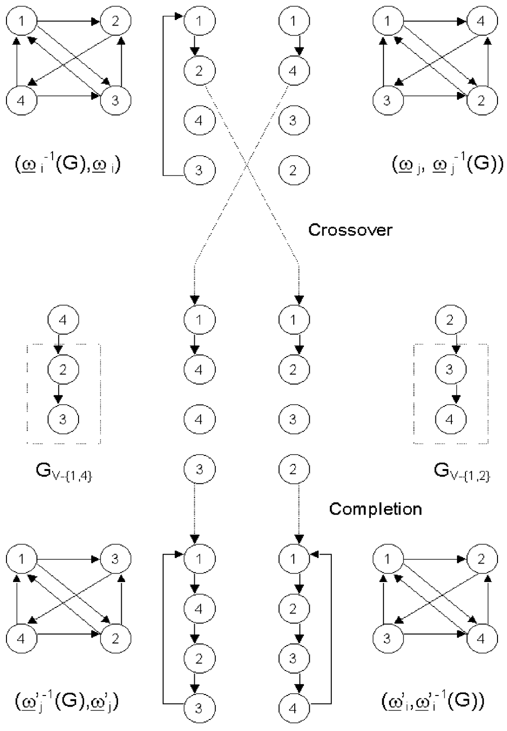
$(\underline{\omega}_i^{-1}(G),\underline{\omega}_i)$
$(\underline{\omega}_j,\underline{\omega}_j^{-1}(G))$
Crossover
$G_{V-\{1,4\}}$
$G_{V-\{1,2\}}$
Completion
$(\underline{\omega}'^{-1}_j(G),\underline{\omega}'_j)$
$(\underline{\omega}'_i,\underline{\omega}'^{-1}_i(G))$

Fig. 4. Stochastic Crossover: Example ($k=2$).

in case of populations of sizes at most cl, where $c \geq 1$ is a suitable constant). We observe that the stochastic crossover consents to obtain much better performances. Note that preliminary (similar) experiments state that the performances obtained by the genetic algorithm based on the SSC procedure cannot further on be improved by using other techniques such those described (for digraphs) in [20], [52] (note a comparison with more sophisticated techniques, requiring computational complexity greater than Angulin and Valiant or Frieze procedures, tends to be prohibitive as the input digraphs size l increases).

Table 1.1. - First Success Expected Time of $SAVGA$ ($k = 1$).

$_l/^p$	1/6	1/8	1/10	1/12	1/14
60	**1.60**	**3.00**	−−	−−	−−
σ_{60}	1.20	2.15	−−	−−	−−
120	**1.00**	**1.00**	**1.25**	**8.80**	**15.60**
σ_{120}	0.00	0.00	0.36	5.24	6.75
180	**1.00**	**1.00**	**1.00**	**1.00**	**1.20**
σ_{180}	0.00	0.00	0.00	0.00	0.40
240	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
σ_{240}	0.00	0.00	0.00	0.00	0.00

Table 1.2. - First Success Expected Time of $SAVGA$ ($k = 1$).

$_l/^p$	1/16	1/18	1/20	1/22	1/24
60	−−	−−	−−	−−	−−
σ_{60}	−−	−−	−−	−−	−−
120	−−	−−	−−	−−	−−
σ_{120}	−−	−−	−−	−−	−−
180	**10.00**	−−	−−	−−	−−
σ_{180}	8.18	−−	−−	−−	−−
240	**1.00**	**2.00**	−−	−−	−−
σ_{240}	0.00	1.55	−−	−−	−−

Table 2.1. - First Success Expected Time of $SSCGA$ ($k = 1$).

$_l/^p$	1/6	1/8	1/10	1/12	1/14
60	**1.00**	**1.40**	**1.40**	**4.60**	**9.40**
σ_{60}	00	0.49	0.80	4.45	8.59
120	**1.00**	**1.00**	**1.20**	**2.00**	**2.80**
σ_{120}	0.00	0.00	0.40	0.89	1.26
180	**1.00**	**1.00**	**1.00**	**1.00**	**1.20**
σ_{180}	0.00	0.00	0.00	0.00	0.40
240	**1.00**	**1.00**	**1.00**	**1.00**	**1.40**
σ_{240}	0.00	0.00	0.00	0.00	0.49

Table 2.2. - First Success Expected Time of $SSCGA$ ($k = 1$).

$_l/^p$	1/16	1/18	1/20	1/22	1/24
60	−−	−−	−−	−−	−−
σ_{60}	−−	−−	−−	−−	−−
120	**3.00**	**5.60**	**6.20**	**8.20**	**14.25**
σ_{120}	1.47	1.94	5.28	7.61	11.25
180	**1.40**	**2.00**	**2.40**	**2.60**	**5.20**
σ_{180}	0.49	1.02	1.50	2.40	4.45
240	**1.60**	**1.80**	**2.00**	**3.00**	**3.40**
σ_{240}	0.50	0.80	1.55	1.62	1.72

Table 2.3. - First Success Expected Time of $SSCGA$ ($k = 1$).

$_l/^p$	1/26	1/28	1/30	1/32	1/34
60	−−	−−	−−	−−	−−
σ_{60}	−−	−−	−−	−−	−−
120	**44.60**	−−	−−	−−	−−
σ_{120}	29.37	−−	−−	−−	−−
180	**8.40**	**10.60**	**10.80**	**19.00**	**35.00**
σ_{180}	6.70	6.83	8.47	16.76	21.67
240	**3.40**	**4.80**	**6.00**	**7.20**	**9.83**
σ_{240}	1.85	1.90	3.49	4.15	7.07

VII. Conclusion

We have introduced a genetic model based on stochastic crossover to solve the Hamiltonian cycle problem ($DHCP$) for random digraphs to which a random Hamiltonian cycle is superposed. The computation technique is inspired and interpreted by reformulating $DHCP$ as an algebraic problem. To make a comparison, we have chosen Angulin and Valiant heuristic, that is, a very efficient method, designed for (a single-processor) Random Access Computer (RAC) [2], to solve $DHCP$ in time $O(l \log^2 l)$ for random digraphs with sufficiently large edge densities ($p \geq \frac{c \log l}{l-1}$, where $c > 1$). Frieze's technique [20] improves Angulin and Valiant result reducing the edge density ($p \geq \frac{\log l}{l-1}$) against a worst case time bound $O(l^{1.5})$. In our simulations, Angulin-Valiant operator has not exhibited meaningful improvements in the performance, with respect to what indicated by the bound $p \geq \frac{\log l}{l-1}$, in case of random digraphs with superposed random Hamiltonian cycles (for $l \leq 2^{10}$). Conversely, the genetic algorithm based on stochastic crossover, in which a random choice of a next vertex is used in the completion process of the offspring (instead of the greedy rule depicted in Section V), improves the performances of $SAVGA$ for $k \geq 1$. More specifically, the genetic algorithm with stochastic crossover sensibly scales down Frieze's bound $p \geq \frac{\log l}{l-1}$ for small $k-$values ($k \geq 2$ and $l \leq 2^9$). In case of undirected graphs the greedy rule is useful to get (also significant) improvements of the performances. Our results are experimental, while it is due to remark that Angulin-Valiant and Frieze results are theoretical and hold for large size graphs (as $l \to \infty$). Starting from such remarks, considering that the stochastic crossover can incorporate several (also adapted) unary operators and since GAs represent heavily distributed computing systems, we conclude that a suitable implementation of the genetic model is useful, in practice, to improve the performance of several principal specialized techniques [2], [20], [8] designed to solve $DHCP$ for random graphs (with superposed random Hamiltonian cycles). This work aims at supporting the

Table 2.4. - First Success Expected Time of *SSCGA* ($k = 1$).

$_{l}/^{p}$	1/36	1/38	1/40	1/42	1/44
60	−−	−−	−−	−−	−−
σ_{60}	−−	−−	−−	−−	−−
120	−−	−−	−−	−−	−−
σ_{120}	−−	−−	−−	−−	−−
180	−−	−−	−−	−−	−−
σ_{180}	−−	−−	−−	−−	−−
240	**10.00**	**30.40**	**44.00**	−−	−−
σ_{240}	8.61	19.05	37.82	−−	−−

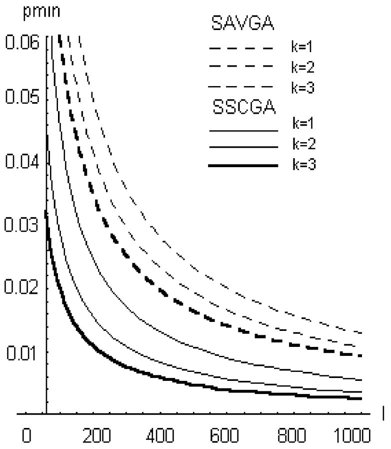

Fig. 5. Comparison between *SAVGA* and *SSCGA* ($1 \leq k \leq 3$).

belief that there is an evolutionary advantage in applying, suitably designed, (stochastic) binary operators, with respect to the simpler unary ones, to fundamental case study such as $NP-$complete problems. We leave theoretical analysis as an important open problem for further research; some fundamental guidelines can be found in [8] and in the Goemans work [22], [25].

Acknowledgment

This paper is in memory of my father Mario Carpentieri.

References

[1] A. Aho, J. Hopcroft, J. Hullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, 1974.

[2] D. Angulin, L. G. Valiant, Fast Probabilistic Algorithms for Hamiltonian Circuits and Matchings, J. Computer and System Sciences, Vol. 18, pp. 155 – 193, 1979.

[3] A. Bertoni, P. Campadelli, M. Carpentieri, G. Grossi, A Genetic Model and the Hopfield Networks, Icann96: International Conference on Artificial Neural Networks, Lectures Notes in Computer Science, pp. 463 – 468, Springer-Verlag, Berlin, Germany, 1996.

[4] A. Bertoni, P. Campadelli, M. Carpentieri, G. Grossi, Analysis of a Genetic Model, ICGA97: Proceedings of the Seventh International Conference on Genetic Algorithms, pp. 121 – 126, Morgan-Kaufmann, San Mateo, California, 1997.

[5] A. Bertoni, P. Campadelli, M. Carpentieri, G. Grossi, A Genetic Model: Analysis and Application to MAXSAT, Evolutionary Computation, pp. 291 – 309, Vol. 8, n. 3, Massachusetts Institute of Technology, Mit Press, Cambridge, MA, 2000.

[6] B. Bollobàs, T. I. Fenner, A. M. Frieze, An Algorithm for Finding Hamilton Paths and Cycles in Random Graphs, Combinatorica, Vol. 7 (4), pp. 327 – 341, 1987.

[7] R. M. Brady, Optimization Strategies Gleaned from Biological Evolution, Nature, Vol. 317, pp. 804 – 806, 1985.

[8] A. Z. Broder, A. M. Frieze, E. Shamir, Finding Hidden Hamiltonian Cycles, Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, pp. 182 – 189, 1991.

[9] F. A. Brunacci, DB2, DB2A: Two Useful Tools for Constructing Hamiltonian Circuits, European Journal of Operation Research, Vol. 34, pp. 231 – 236, 1988.

[10] T. G. Bui, B. R. Moon, A New Genetic Approach for the Traveling Salesman Problem, Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 7 – 12, 1994.

[11] M. Carpentieri, A Bivariate Marginal Distribution Genetic Model, CEC2006: 2006 IEEE Congress on Evolutionary Computation, Vancouver, Canada, pp. 1016 – 1022, 2006.

[12] M. Carpentieri, A Genetic System Based on Simulated Crossover of Sequences of Two-Bit Genes, Theoretical Computer Science, Vol. 367, n. 3, pp. 324–335, Elsevier Science, 2006.

[13] T. Cormen, R. Rivest, C. Leiserson, Introduction to Algorithms, MIT Press, 1990.

[14] Y. Davidor Epistasis Variance: a Viewpoint on GA-Hardness, Foundations of Genetic Algorithms, G. J. E. Rawlings Editor, pp. 23 – 35, San Mateo, CA: Morgan Kaufmann, 1991.

[15] L. Davis, Applying Adaptive Algorithms to Epistatic Domains, Proceedings of the International Joint Conference on Artificial Intelligence, pp. 162 – 164, 1985.

[16] J. Dzubera, D, Whitley, Advanced Correlation Analysis of Operators for the Traveling Salesman Problem, Parallel Problem Solving from Nature - Proceedings of the Third Workshop, PPSN III, Springer-Verlag, 1994.

[17] B. Freisleben, P. Merz, A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems, Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, Nagoya, Japan, pp. 616 – 621, 1996.

[18] B. Freisleben, P. Merz, New Genetic Local Search Operators for the Traveling Salesman Problem, Proceedings of the Fourth Conference on Parallel Problem Solving from Nature, Lecture Notes in Computer Science, H. M. Voight, W. Ebeling, I. Rechenberg and H. P. Schwefel Editors, Vol. 1141 pp. 890 – 899, Springer, 1996.

[19] B. Freisleben, P. Merz, Genetic Local Search for the TSP: New Results, Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, IEEE Press, pp. 159 – 164, Indianapolis, USA, 1997.

[20] A. M. Frieze, An Algorithm for Finding Hamilton Cycles in Random Directed Graphs, Journal of Algorithms, Vol. 9, pp. 181 – 204, 1988.

[21] M. Garey, D. Johnson, Computer and Intractability: A Guide of the Theory of the NP-Completeness, W. H. Freeman, San Francisco, 1979.

[22] M. X. Goemans, Approximation Algorithms, 18.415/6.854 Advanced Algorithms, 1994.

[23] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, *MA*, 1989.

[24] D. E. Goldberg, R. Lingle, Alleles, Loci and the TSP, Proceedings of the First International Conference on Genetic Algorithms ana their Applications, pp. 154 – 159, 1985.

[25] M. X. Goemans, D. P. Williamson, Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming, Journal of the Association for Computing Machinery, Vol. 42, n. 6, 1995.

[26] J. J. Grefenstette, Incorporating Problem Specific Knowledge into Genetic Algorithms, Genetic Algorithms and Simulated Annealing, Morgan-Kaufmann, pp. 42 – 60, 1987.

[27] J. J. Grefenstette, R. Gopal. B. Rosimaita, D. V. Gucht, Genetic Algorithms for the Traveling Salesman Problem, Proceedings of the First International Conference on Genetic Algorithms and Their Applications, pp. 160 – 168, 1985.

[28] J. H. Holland, Induction, Processes of Inference, Learning and Discovery, *MIT* Press, Cambridge, 1989.

[29] J. H. Holland, Adaptation in Natural and Artificial Systems, *MIT* Press, Cambridge, 1992.

[30] A. Homaifar, S. Guan, G. E. Liepins, A New Approach to Traveling Salesman Problem by Genetic Algorithms, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 460 – 466, Morgan-Kaufmann, 1993.

[31] J. E. Hopcroft and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 1979.

[32] J J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, Proceedings of the National Academy of Sciences, pp. 2554 – 2558, 1982.

[33] G. E. Liepins, M. R. Hilliard, M. Palmer, M. Morrow, Greedy Genetics, Genetic Algorithms and their Applications: Proceedings of the Second International Conference, pp. 90 – 99, 1987.

[34] J. R. Koza, Genetic Programming, MIT Press, Cambridge, *MA*, 1992.

[35] K. Mathias, D. Whitley, Genetic Operators, the Fitness Landscape and the Traveling Salesman Problem, Parallel Problem Solving from Nature, R. Männer and B. Manderick Editors, pp. 219 – 228, North Holland-Elsevier, 1992.

[36] P. Merz, B. Freisleben, Memetic Algorithms for the Traveling salesman Problem, Complex Systems, Vol. 11, 1997.

[37] C. J. H. McDiarmid, Clutter Percolation and Random Graphs, Math. Programming Stud., Vol. 13, pp. 17 – 25, 1980.

[38] A. Mendiburu, J. A. Lozano, J. Miguel-Alonso, Parallel Implementation of EDAs Based on Probabilistic Graphical Models, *IEEE* Transactions on Evolutionary Computation, Vol. 9, n. 4, 2005.

[39] H. Mühlenbein, Parallel Genetic Algorithms, Populations, Genetics and Combinatorial Optimization, Proceedings of the First International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, pp. 416 – 421, 1989.

[40] H. Mühlenbein, R. Höns, The Estimation of Distributions and the Minimum Relative Entropy Principle, Evolutionary Computation, Vol. 13, n. 1, pp. 1 – 27, 2005.

[41] H. Mühlenbein, M. George-Schleuter, O. Kramer, Evolution Algorithms in Combinatorial Optimization, Parallel Computing, Vol. 7, pp. 65 – 85, 1988.

[42] B. Naudts and L. Kallel A Comparison of Predictive Measures of Problem Difficulty in Evolutionary Algorithms, IEEE Transactions on Evolutionary Computation, Vol. 4 (1), pp. 1 – 15, 2000.

[43] I. M. Oliver, D. J. Smith, J. R. C. Holland, A Study of Permutation Crossover Operators on the TSP, Genetic Algorithms and their Applications, Proceedings of the Second International Conference on Genetic Algorithms, pp. 224 – 230, 1987.

[44] V. A. Perepelica, On Two Problems from the Theory of Graphs, Soviet. Math. Dokl., Vol. 11, pp. 1376 – 1379, 1970.

[45] M. Pelikan, H. Mühlenbein, The Bivariate Marginal Distribution Genetic Algorithm, Advances in Soft Computing - Engineering Design and manufacturing, Springer-Verlag, London, pp. 521 – 535, 1999.

[46] S. Rochet, Epistasis in Genetic Algorithms Revisited, Information Sciences, Vol. 102, pp. 133 – 155, 1997.

[47] E. Shamir, How Many Edges Make a Graph Hamiltonian ?, Combinatorica, Vol. 3 (1), pp. 123 – 131, 1983.

[48] T. Starkweather, S. McDaniel, K. Mathias, C. Whitley, D. Whitley, A Comparison of Genetic Sequencing Operators, Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, pp. 69 – 76, 1991.

[49] G. Syswerda, Scheduling Optimization Using Genetic Algorithms, Handbook of Genetic Algorithms, pp. 332 – 349, 1991.

[50] A. Y. C. Tang, K. S. Leung, A Modified Edge Recombination Operator for the Traveling Salesman problem, Parallel Problem Solving from Nature - Proceedings of the Third Workshop - PPSN III, pp. 180 – 188, Springer, 1994.

[51] N. L. J. Ulder, E. H. L. Aarts, H. J. Bandelt, P. J. M. van Laarhoven, E. Pesch, Genetic local Search Algorithms for the Traveling Salesman Problem, Parallel Problem Solving from Nature - Proceedings of the First Workshop - PPSN I, pp. 109 – 116, Springer, 1991.

[52] B. Vandegriend, Finding Hamiltonian Cycles: Algorithms, Graphs and Performance, Master of Science Thesis, Department of Computing Science, University of Alberta, Edmonton, Alberta, 1998.

[53] H. E. Voight, H. Mühlenbein, Gene Pool Recombination and the Utilization of Covariances for the Breeder Genetic Algorithm, Proceedings of the Second *IEEE* International Conference on Evolutionary Computation, pp. 172 – 177, Z. Michalewitch Editor, *IEEE* Press, New York, 1995.

[54] M. D. Vose, The Simple Genetic Algorithm: Theory and Foundations, *MIT* Press, Cambridge, *MA*, 1999.

[55] D. Whitley, T. Starkweather, D'Ann Fuquay, Scheduling Problems and Traveling Salesman: the Genetic Edge Recombination Operator, Proceedings of the Third International Conference on Genetic Algorithms, pp. 133 – 140, 1989.

[56] D. Whitley, T. Starkweather, D. Shaner, The Traveling Salesman and Sequence Scheduling: Using Genetic Edge Recombination, Handbook of Genetic Algorithms, pp. 350 – 372, 1991.

[57] E. M. Wright, For How Many Edges is a Digraphs Almost Certainly Hamiltonian ?, Proc. Amer. Math. Soc. , Vol 41 (2), pp. 384 – 388, 1973.

[58] Q. Zhang, On the Convergence of a Class of Estimation of Distribution Algorithms, *IEEE* Transactions on Evolutionary Computation, Vol. 8, n. 2, 2004.

[59] Q. Zhang, On Stability of Fixed Points of Limit Models of Univariate marginal Distribution Algorithm and Factorized Distribution Algorithm, *IEEE* Transactions on Evolutionary Computation, Vol. 8, n. 1, 2004.