Local search algorithms for memetic algorithms: understanding behaviors using biological intelligence

Beatrice Luca
Department of Computer Science
Faculty of Automatic Control and Computer Engineering
Iasi, Romania
beatrice.002@gmail.com

Mitica Craus
Department of Computer Science
Faculty of Automatic Control and Computer Engineering
Iasi, Romania
mitica.craus@gmail.com

***Abstract*—Memetic Algorithms (MAs) are a class of stochastic global search heuristics in which Evolutionary Algorithms (EAs) - based approaches are combined usually with heuristic local searches. This hybridization is meant to reach solutions that would otherwise be unreachable by evolution or a local method alone. In this work, we propose three Local Search (LS) algorithms for hybridization with an existing Evolutionary Algorithm with Pareto ranking in order to define biological intelligence using the concepts of useful and utility and therefore to zoom on the basin of attraction of promising realistic solutions. Our experimental results with these memetic algorithms in the game of Checkers show how we can learn the organization of behaviors into paths of behaviors of different lengths and frequencies and then reveal the true nature of these behaviors.**

***Index Terms*—memetic algorithms, evolutionary algorithms, local search algorithms, mechanical and biological intelligence**

I. Introduction

Behaviors are exceptionally complex traits. Their sensitivity to environmental variation poses daunting challenges for unraveling their underlying nature. In this work, we propose a new approach for understanding complex behaviors: first, we consider information as the fabric or the creator of everything that exists in the world; then, using memetic algorithms [1], [12] we explain the formation of organized components (called paths of behaviors) as a result of useful work; finally, we show that utility can be used to discover the true nature of behaviors.

Our design of memetic algorithms combines traditional components initially proposed by `ICE` agent model [11] (i.e., motion and reflection operators, biological properties, Evolutionary Algorithm (EA)) with Local Search (LS) components that use information in relation with reality. All these components hold the concepts of “motivation”, “development”, “adaptation” and “experience”, emphasizing both mechanical and biological intelligence. Concretely, in `ICE` model, mechanical intelligence takes inspiration from classical physics [14] and is implemented using operators for agent motion ($>$) and reflection ($>>$) to best future opportunities and parameters to control or limit the agent motion. These operators dictate how information is accessed and processed, therefore explain how reality emerges from a multitude of opportunities. In `ICE` model, biological intelligence is implemented using the concepts of (1) embodiment - individuals have properties such as Universality and Embodiment properties, (2) evolution - evolution represents a high level manifestation of individuals derived from the following rule of life: when individuals reproduce, their children have the same Universality and Embodiment properties and (3) cognition - cognition represents the ability to have an embodiment as described by the Embodiment property, that is to have cognitive skills to acquire more information as the world becomes more complex.

In this work, we define biological intelligence as the ability of individuals to access information directly or indirectly using the concepts of useful and utility. Useful means anything that leads to a greater organization of complex behaviors and utility means anything that is profitable. Concretely, our solution to biological intelligence proposes three new approaches to optimization: (1) first approach converts row energy of individuals into useful work, (2) second approach improves upon the previous approach by increasing the efficiency of energy use (3) third approach uses the concept of utility to discover the true nature of behaviors.

`LocalSearch-ICE` (`LS-ICE`) algorithm implements our first approach to optimization. This algorithm explores the local space by creating so called useful correlations - connections created between consecutive behaviors that share information and are sufficiently powerful to drive useful work. We say that information is accessed indirectly because we don’t care about the exact distance (or similarity) between these behaviors. Although this algorithm is a good local explorer, it is not sufficiently powerful to organize behaviors because it only looks at the useful correlations produced by the most profitable individuals.

Therefore, we propose a second LS algorithm called `AlternateLocalSearch-ICE` (`ALS-ICE`) to implement the second approach to optimization. This algorithm improves `LS-ICE` by diversification of individuals selected for local search - now both profitable and non-profitable individuals are used to search for useful correlations. This algorithm leads to

978-1-5386-4444-7/18/$31.00 ©2018 IEEE

a better organization of behaviors into paths of behaviors of different lengths and frequencies. Once we learn this organization, we use `BestLocalSearch-ICE` (`BLS-ICE`) to implement the third approach to optimization. This algorithm shifts from the concept of useful to the concept of utility as follows: first it proposes that information is accessed in a controlled manner, directly through high similarity measures between two consecutive behaviors, then both profitable and non-profitable individuals are used to search for profitable correlations, gradually revealing the true nature of behaviors.

We combine the above LS algorithms with the EA algorithm proposed in [10] and obtain three memetic algorithms (`MA-LS`, `MA-ALS` and `MA-BLS`). Since designing MAs is usually improvised, we discuss several important design issues: where exactly should local search be applied, which individuals in the population should be improved and how should they be improved and how can global information can be integrated with local search?

In Section II, we discuss related works that look into the design of memetic algorithms. In Section III, we describe the modeling of intelligent agents using `ICE` agent model, introduce the notions of correlation, paths of behaviors and useful/profitable correlations and present in detail three LS algorithms: `LS-ICE`, `ALS-ICE` and `BLS-ICE`. In Section IV, we present our experimental results with MAs `MA-LS`, `MA-ALS` and `MA-BLS` in the game of Checkers.

II. Related work

The combination of evolutionary algorithms with local search was named "memetic algorithms" (MAs) [1]. These methods are inspired by models of adaptation in natural systems that combine the evolutionary adaptation of a population with individual learning within the lifetimes of its members. From an optimization point of view, MAs have been shown to be both more efficient and more effective because they require fewer evaluations to find optimal higher quality solutions [2], [3].

Despite the impressive results achieved by MAs, their design is still improvised. Furthermore, the design of MAs rises a number of important issues which must be addressed by the practitioner. [6] proposes a syntactic model and a taxonomy of MAs in order to clarify the main design issues of MAs. Namely, the local search stage can happen before or after an evolutionary operator such as mutation, reproduction or selection. Local search can also be coordinated with these operators - for example reproduction is applied to an individual and the result of this operation is given as an argument to a search function or individual is improved with a local search function and then reproduction is applied to this result [4], [5]. In a similar spirit, [17] proposes "crossover-aware" and "mutation-aware" local searches.

According to [6], [8] and [7], local search can also be coordinated with population management - for example organizing the application of a local searcher to a particular subset of the population. In this case we may have coarse-grain coordination (provide population statistics to its local searcher) or fine-grain coordination (just knows one individual at a time). In addition to coordination with population management, it is possible to incorporate historical information into the search mechanism - for example we can use historical information when we want to combine global and local information across time or update the probabilities of applying genetic operators [9].

From the above syntactic modelings, a taxonomy of architectural classes can be naturally derived based on index number (i.e., a 4 bit binary number) $D(A) = b_0b_1b_2b_4$ which can be assigned to any $MA(A)$ to describe whether a type of coordination is absent or present in A. In general, a lower D algorithm should be preferred to one with a larger D.

In the following, we summarize some of the important issues in the design of competent MAs [13]: (1) choice of local search operators, (2) integration into EA cycle and (3) managing the global-local search trade-off. Looking into MA literature, we found that MAs usually use multiple local search operators because they avoid spending time utilizing nonproductive operators. Most recent work tend to simply incorporate one or more local search algorithms into an EA which can lead to a rapid loss of diversity with clear implications on the quality of the EA.

Finally, some works look into the integration of the local search operators with the genetic operators (perform a partial local search - only those solutions that are found promising will be assigned for local search) and the issue of deep local optima (to avoid getting trapped in a local optimum multiple local searchers can be used simultaneously in the population). In [16] and [17], the issue of large neutral plateaus and deep local optima is addressed by providing modified local searchers that can change their behavior accordingly to the convergence state of the evolutionary search.

III. Methodology

A. ICE agent modeling

We describe an agent modeling W_{ICE} for a given world W using the following elements: (a) a map M of physical locations with dimensions and colors, (b) a set of entities E that generate all the information R_B that is ever used in W and which is distributed across a set of consecutive frames F, (c) a set of agents A with their own behavior R_C, local and global graphs D_S of interactions with the world and a set of evolutionary mechanisms h_{EA} associated to each population in the world (Eq. 1).

$$W_{ICE} = (M\{D, L, C\}, E\{R_B, F\}, A\{R_C, D_S, h_{EA}\}) \quad (1)$$

where,

- M represents the matrix of coordinates (x, y) of locations in the physical world. Specification $\{D, L, C\}$ describes the world as a two-dimensional world D, with colored locations L, according the a given color set C.
- E represents the set of all entities that move in the world (e.g., game pieces, cars, artificial robot legs, etc.). Specification $\{R_B, F\}$ describes each entity e_i in E as an object that uses a particular set of basic rules R_{B_i}

from a larger set R_B, rules which are executable in several computational contexts $(\tau_{i,1}, R_{B_{i,1}})$,...,$(\tau_{i,n}, R_{B_{i,n}})$ within current frame (or environment) $F_c \in F$. We call U_{c_i} an instance of frame F_c, that describes how rules in R_B have been applied in the physical world.

- A represents the set of all agents in world W. Specification $\{R_C, D_S, h_{EA}\}$ describes each agent a_i in A as the owner of a set of complex rules R_{Ci} in R_C that represent his behavior, an interaction topology D_{S_i} from D_S made of local and global computational contexts and heuristic h_{EA_i} from h_{EA} addressing search and multi-criteria decision making. In this work, we consider that each h_{EA_i} is a DICE algorithm, i.e., an evolutionary algorithm with Pareto-ranking first introduced in [10].

B. *Agent optimization using* LocalSearch-ICE *Algorithm*

The basic algorithmic structure we use for storing all rules and correlations is an undirected weighted graph $G = (\mathcal{V}, \mathcal{E})$ which stores in $\mathcal{V}$ all complex rules that are active in the current frame F_c and in $\mathcal{E}$ all relations between any two complex rules. A relation, therefore an edge $e(v_i, v_j)$ exists between two vertices v_i and v_j in $\mathcal{V}$, if their corresponding rules rc_i in v_i and rc_j in v_j share information, i.e., $I(rc_i) \cap I(rc_j) \neq \emptyset$. We introduce the concept of correlation which represents the amount of information shared by any two vertices. Eq. 2 describes the concept of correlation using an indicative conditional: for any two vertices v_i and v_j in graph G, if there is an edge $e(v_i, v_j)$ between these vertices, then a correlation $c(v_i, v_j)$ between v_i and v_j represents the amount of information $\mid I(rc_i) \cap I(rc_j) \mid$ shared by rules rc_i and rc_j in v_i and v_j.

(correlation)

$$\forall v_i, v_j \in \mathcal{V}, if\ e(v_i, v_j), then\ c(v_i, v_j) = \mid I(rc_i) \cap I(rc_j) \mid \quad (2)$$

A path of behaviors represents an ordered set of behaviors, therefore a path of correlations between consecutive vertices, that starts from a single source v_0, is finite and can contain cycles. Eq. 3 describes the concept of path using the following proposition: a path p is a sequence of vertices $v_0, v_1, \ldots, v_m$, such that, there is a correlation $c(v_i, v_{i+1})$ between any two consecutive vertices.

(path of behaviors)

$$p = v_0, v_1, \ldots, v_m, s.t. \forall\ v_i, v_{i+1}, i = \overline{0, m-1}, \exists\ c(v_i, v_{i+1}) \quad (3)$$

where, $c(v_i, v_{i+1})$ is well defined according to Eq. 2.

So far, we have not assumed any connection of agents with their reality. Next, we define useful correlations and propose calculation of point estimate of the profit accumulated by an agent (p_{ep}) using a random data sample taken from previous universe U'. Main advantage of using point estimates is to increase the level of 'realism' that we include in our evaluation of correlations. Eq. 4 formalizes the concept of useful correlation using the following bi-conditional: for a correlation created from vertex v_i to vertex v_j to be useful in current universe U it is necessary and sufficient that the profit $f_p(rc_j)$ of rc_j gets closer to the point estimate of the profit p_{ep} calculated from U' by a small error ϵ.

(useful correlation)

$$c(v_i, v_j)\ is\ useful \Leftrightarrow f_p(rc_j) \in [p_{ep} - \epsilon, p_{ep} + \epsilon] \quad (4)$$

where, p_{ep} represents the point estimate of the profit calculated from universe U' that precedes current universe U.

The algorithm LocalSearch-ICE (or LS-ICE) for local search is presented in Fig. 1. It implements a local search mechanisms which receives as input the correlation matrix M of all complex rules available in the current universe U and it returns the set of all paths of behaviors $Paths$. The algorithm starts with an initialization step to set the index of the current path and the set $Paths$ found so far to 0 and $\emptyset$ respectively, and $firstactive$ Boolean variable for the state of the first element in a path to true (*line* 1). It processes a list of paths that emerge in U until a transition to the next frame occurs, it doesn't exist any vertex v_i that is active or a counter on the maximum number of moves in a single universe is reached (*line* 2). The search for a new path p_l is initialized from a random vertex v_i in $\mathcal{V}$ containing a complex behavior rc_i that is active (i.e., can be instantiated in the world) (*lines* $4, 5, 6$). A useful correlation is created from v_i to v_j as follows:

- First, the set of strategies (individuals) S is processed and only 50% most profitable strategies that lead to useful correlations are returned in S_{Useful} (*line* 9);
- Second, a random vertex is chosen from a random strategy in S_{Useful} such that v_j is active, i.e., behavior rc_j can be instantiated in the world (*line* 10);
- Third, v_j is stored in the current p_l and behavior rc_j gets instantiated in the world (*lines* 11, 12). The index of the current vertex in M is changed from i to j (*line* 13) and the algorithm loops to search the next node in p_l until no correlations can be created in p_l (i.e. Boolean variable $active$ becomes $false$).

The algorithm returns the set of all paths $Paths$ that emerge from current universe.

C. *Extensions of* LS-ICE

LS-ICE algorithm limits access to information to only individuals which are profitable and produce useful correlations. In this section, we discuss two extensions of LS-ICE called AlternateLocalSearch-ICE (ALS-ICE) and BestLocalSearch-ICE (BLS-ICE) algorithms.

ALS-ICE algorithm is a modified LS-ICE which allows optimization for all individuals. Concretely, it searches for the most profitable strategies to create useful correlations, while introducing genetic variation by searching also for the least profitable strategies. Implementation wise, this algorithm only adds a $switch$ variable to alternate search among the 50% most profitable and 50% least profitable strategies, so it does not change the overall complexity of the algorithm.

BLS-ICE algorithm improves ALS-ICE by searching for the most profitable strategies to create this time - profitable

Algorithm LocalSearch-ICE(Matrix M) - local search with adaptation to useful correlations
INPUT: M - correlation matrix of all complex rules.
OUTPUT: $Paths$ - set of all paths $\{p_1, \ldots, p_{N_P}\}$ that describe agent behavior in the current universe U

1\. Init: $l = 0$ (index of current path), $Path = \emptyset$ (empty set of paths), $i = 0$ (index of the first element in a path) and $firstactive = true$ (Boolean variable for the state of the first element in a path)
2\. **While** $((transition(F_{curr}, F_{next})! = true)$ && $(firstactive == true)$ && $(counter \leq 10))$ {
// process all paths of correlations in the current universe U
3\. $v_i \leftarrow randomActive(S)$;
4\. **If** $(v_i! = null)$ {
5\. *// process a single path of correlations*
6\. $p_l \leftarrow addtoList(v_i, p_l)$;
7\. **While** $(active == true)$ {
8\. $S_{Useful} \leftarrow ProcessStrategies(S, M, v_i)$;
9\. $v_j \leftarrow randomActive(S_{Useful})$;
10\. **If** $(v_j! = null)$ {
11\. $p_l \leftarrow addtoList(v_j, p_l)$;
12\. $instantiate(rc_j)$;
13\. $i = j$;
}
14\. **Else** $active = false$;
}
15\. $Paths \leftarrow addtoList(Paths, p_l)$;
}
16\. **Else** $firstactive = false$;
}
17\. return $Paths$;

Fig. 1. LocalSearch-ICE algorithm with adaptation to useful correlations.

correlations - while introducing genetic variation by searching also for the least profitable strategies. For both profitable and non-profitable strategies, only neighboring vertices with high similarity are considered for execution. High similarity between rc_i and a neighboring vertex rc_j is measured in Eq. 5 using an Euclidean distances measure $d_{rc_i,rc_j} < c_{min}$.

(high similarity)

$$d_{rc_i,rc_j} = \mid I(rc_i) - I(rc_j) \mid \leq c_{min}, c_{min} \in \mathcal{N}^+ \quad (5)$$

where, $I(rc_i)$ and $I(rc_j)$ represent the sets of information that make up complex behaviors rc_i and rc_j.

The concept of profitable correlation is described in Eq. 6 as a correlation that produces a profit greater than the point estimate of the profit p_{ep} calculated from the previous universe U'.

(profitable correlation)

$$c(v_i, v_j) \ is \ profitable \Leftrightarrow f_p(rc_j) > p_{ep} \quad (6)$$

where, p_{ep} represents the point estimate of the profit calculated from previous universe U'.

Finally, we argue that `BLS-ICE` implements 'true optimizations' because it unfolds information in a direct, controlled manner such that only behaviors with high similarity are executed, it allows only profitable correlations to occur and it assumes diversity of created correlations. Therefore, this optimization results in smooth transitions from one behavior to another and the creation of more meaningful contexts which can reveal the true nature of behaviors.

D. Memetic algorithms

In this section we discuss the design of `MA-LS`, `MA-ALS` and `MA-BLS` algorithms:

1) Each local search algorithm - `LS-ICE`, `ALS-ICE` or `BLS-ICE` - is applied before any mutation or reproduction operations in order to optimize the individuals in a given population, that is to adapt them to the existing reality. We call this biological intelligence because reality takes central stage.
2) Each local search algorithm explores different mechanisms for accessing individuals, i.e., local search is applied selectively - only to the most profitable individuals, or to both profitable and non-profitable individuals. At dynamic time, in order to avoid complete local searches, only 50% of the population is searched. This reduces the performance time of the MAs.
3) Each local search algorithm controls the intensity of the search differently: `LS-ICE` and `ALS-ICE` use only useful correlations to search among the existent paths of behaviors, while `BLS-ICE` uses only correlations that are profitable. `LS-ICE` and `ALS-ICE` focus on producing more useful work and therefore explain the formation of organized paths of behaviors and `BLS-ICE` focuses on the utility of these paths and therefore explains the nature of behaviors.
4) Each local search algorithm combines local and global information by using information about the point estimates of the profit computed from previous population. This approach increases the level of realism that we include in our evaluation of correlations.

IV. Experimental evaluation

We start our experiments with the modeling of space in Checkers according to the `ICE` agent model. All complex rules in Checkers,

- Rule for moving forward left or right
$rc_1 = ((rb_1 >> rb_2)[k_{1,1}])(k_{1,2})$, $k_{1,2} = 1$
- Rule for moving forward/backward left or right
$rc_2 = ((rb_1 >> rb_2 >> rb_3 >> rb_4)[k_{2,1}])(k_{2,2})$, $k_{2,2} = 1$
- Rule for capturing opponent pieces forward left or right
$rc_3 = ((rb_5 >> rb_6)[k_{3,1}])(k_{3,2})$, $k_{3,2} \in [1, 2]$
- Rule for capturing opponent pieces forward/backward left or right
$rc_4 = ((rb_5 >> rb_6 >> rb_7 >> rb_8)[k_{4,1}])(k_{4,2})$, $k_{4,2} \in [1, 2]$

- Rule for blocking opponent pieces
  $rc_5 = ((rb_9 >> \cdots >> rb_{34})[k_{5,1}])(k_{5,2})$, $k_{5,2} = 1$

are grouped into two frames F_1 and F_2 (rc_1 and rc_3 are active rules in F_1, while rc_1, rc_2, rc_3, rc_4 and rc_5 are active rules in F_2) and for simplicity all depth parameters are zero (i.e., $k_{1,1} = k_{2,1} = k_{3,1} = k_{4,1} = k_{5,1} = 0$). These two frames have been set up with the following objectives: F_1 has one goal - g_1 to move pieces, capture opponent pieces and transform one piece into a king, F_2 has two goals - g_1 from F_1 and g_2 to block all opponent pieces.

We explain the concepts of correlation, path of behaviors, useful and profitable correlation in the context of Checkers game as follows: a) if an agent executes a rule rc_i followed by another rule rc_j in two consecutive moves and their instances share some information, then we obtain a correlation between rc_i and rc_j; b) if an agent executes a set of rules $rc_0, rc_1, \ldots, rc_m$ in $m+1$ consecutive moves and there is a correlation between any two consecutive instances, then we obtain a path of behaviors; and c) useful and profitable correlations add some conditions on the correlations related to their usefulness or profitability.

The implementation is done in Java, using an 8×8 checkerboard with 12 pieces per side. Experimentally, we will test the memetic algorithms on two agents that play Checkers against each other ((a) Game 1: agent A plays Checkers against agent B, both using `MA-LS`, (b) Game 2: agent A plays Checkers against agent B, both using `MA-ALS` and (c) Game 3: agent A plays Checkers against agent B, both using `MA-BLS`) and analyze two of their most frequent behaviors, rc_1 and rc_3 respectively. Namely, we split our experiments in two parts - the first part is concerned with learned paths of behaviors in F_1 and the second part is concerned with learned paths of behaviors in F_2 and how they compare with those from F_1.

A. *Evaluation of frame F_1*

In this section we discuss the experimental results obtained from Games 1,2 and 3 in frame F_1. Figure 2 depicts the results of experiments for behaviors rc_1 and rc_3, however similar discussion holds for the other behaviors as well.

First, we run Game 1 and learn that there is no clear organization of paths of behaviors by their frequency although `LS-ICE` does some useful work using only profitable individuals. For instance, for behavior rc_1 (blue line in Fig. 2(a)), small and big paths of correlations have mixed frequencies (between 0 and 8) with one exception - paths of length 10 which have a high frequency (14). Behavior rc_3 has only two lengths for paths of correlations (1 and 2 since 3 produced no instances) which are clearly very different in their frequencies (blue line in Fig. 2(b)), therefore we can conclude that Game 1 produced a good organization of paths of behaviors for rc_3 based on their frequencies.

Next, we want to increase the amount of useful work by running Game 2 in order to produce a better organization of paths of behaviors for rc_1. From experimental results (red line in Fig. 2(a)) we clearly see that Game 2 successfully organizes paths of correlations in different classes of frequencies based on path lengths. We distinguish between several sub-systems:

- sub-system 1 - paths with lengths 1 and 2 have frequencies in the interval $[2, 4]$;
- sub-system 2 - paths with lengths between 3 and 5 have frequencies in the interval $[6, 7]$;
- sub-system 3 - paths with lengths between 6 and 8 have frequencies in the interval $[2, 3]$;
- sub-system 4 - paths with lengths 10 have frequencies in the interval [13,13].

Since sub-systems 2 and 4 are the most active, they dictate the flow of energy through the complex system and drive the processes of self-organization and the emergence of other sub-systems (like sub-systems 1 and 3).

The final step in our experiment is to run Game 3 and learn what is the exact nature of rc_1 and rc_3 in the above sub-systems, i.e., learn what are the frequencies of paths of behaviors in different sub-systems when more realistic conditions are set in (i.e., using profitable correlations only). For a given sub-system, the nature of behavior is described by its most frequent paths of behaviors where the agent spends most of its time.

From Game 3 (green line in Fig. 2(a)) we learn that:

- in sub-system 1 the true nature of rc_1 is described by the most frequent paths - with lengths 1 and 2;
- in sub-system 2 the true nature of rc_1 is described by the most frequent paths - with lengths 3 and 5;
- in sub-system 3 the true nature of rc_1 is described by the most frequent paths - with length 6;
- in sub-system 4 the true nature of rc_1 is described by the most frequent paths - with length 10.

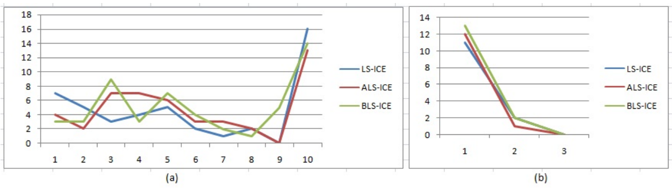

Fig. 2. Number of paths of correlations in frame F_1 for Games 1,2 and 3 using `LS-ICE`, `ALS-ICE` and `BLS-ICE` local search algorithms: Fig 2(a) - frequency of useful/profitable paths of correlations of different lengths for behavior rc_1, and Fig 2(b) - frequency of useful/profitable paths of correlations of different lengths for behavior rc_3.

B. *Evaluation of frame F_2*

In the previous section we have seen how the true nature of agents behavior can be incrementally revealed by producing more useful work and shifting from the concept of useful to that of utility. In this section, we would like to compare the results obtained from both F_1 and F_2.

First, we notice a change in dynamics. In Game 1 (blue line in Fig 3.(a)), for instance, F_2 produces an overall growth in the number of paths of correlations for almost all path lengths. When more useful work is introduced in the system through

Game 2, the system is again organized in sub-systems (red line in Fig 3.(a)):

- sub-system 1 - paths with lengths 1,2 and 3 have frequencies in the interval $[9, 10]$;
- sub-system 2 - paths with lengths 4,5,6,7 and 8 have frequencies in the interval $[1, 5]$;
- sub-system 3 - paths with lengths 9 and 10 have frequencies in the interval $[9, 10]$.

Since now sub-systems 1 and 3 are the most active, they dictate the process of self-organization and the emergence of other sub-systems (like sub-system 2).

When comparing F_1 with F_2 in Game 2, we notice that sub-system 1 (now incorporating parts of previous sub-systems 1 and 2) becomes very active surpassing sub-system 2.

The final step is to run Game 3 (green line in Fig. 3(a)) and observe the true nature of behaviors:

- in sub-system 1 the true nature of rc_1 is described by the most frequent paths - with length 3;
- in sub-system 2 the true nature of rc_1 is described by the most frequent paths - with lengths 4, 6;
- in sub-system 3 the true nature of rc_1 is described by the most frequent paths - with length 10.

In conclusion, this result shows that rc_1 has changed its nature from frame F_1, now becoming more complex - allowing only for larger paths lengths to occur, while rc_3 has become simpler in F_2.

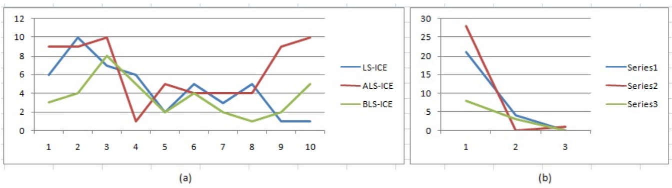

Fig. 3. Number of paths of correlations in frame F_2 for Games 1, 2 and 3 using `LS-ICE`, `ALS-ICE` and `BLS-ICE` local search algorithms: Fig 3(a) - frequency of useful/profitable paths of correlations of different lengths for behavior rc_1, and Fig 3(b) - frequency of useful/profitable paths of correlations of different lengths for behavior rc_3

V. Conclusion

In this work we proposed three memetic algorithms that use novel LS algorithms to improve the solution of multi-objective optimization problems and define biological intelligence. First, we introduce the concepts of useful and utility and show how to implement them in LS algorithms. Then, we discuss several important design issues of memetic algorithms. Our experiments explain the complexity of behaviors in Checkers and how to reveal their true nature.

We conclude that `MA-ALS` algorithm is a good method to learn the organization of behaviors into paths of behaviors and `MA-BLS` algorithm is a good method to understand the nature of these behaviors. In the future, we plan to use these memetic algorithms in more realistic environments in order to learn about the nature of more complex behaviors (i.e., like human behaviors).

References

[1] P. Moscato, “On evolution, search, optimization, GAs and martial arts: toward memetic algorithms,” California Inst. Technol., Pasadena, CA, Tech. Rep. Caltech Concurrent Comput. Prog. Rep. 826, 1989.

[2] L. Davis, Handbook of Genetic Algorithms. New York: Von Nostrand, 1991.

[3] D. Goldberg and S. Voessner, “Optimizing global-search hybrids,” in Proc. Genetic Evol. Comput. Conf., W. Banzhaf, J. Daida, A. Eiben, M. Garzon, V. Honavar, M. Jakaiela, and R. Smith, Eds., 1999, pp. 220-228.

[4] Y. Nagata and S. Kobayashi, “Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem.” in Proc. 7th Int. Conf. Genetic Algorithms, T.Back, Ed., 1997, pp. 450-457.

[5] J. Sakuma and S. Kobayashi, “Extrapolation-directed crossover for job-shop scheduling problems: Complementary combination with JOX,” in GECCO-2000:Proc. Genetic Evol. Comp. Conf., D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, and H.-G. Bayer, Eds., 2000, pp. 973-980.

[6] N. Krasnogor and J. Smith, “A tutorial for competent memetic algorithms: model, taxonomy, and design issue,” in IEEE Transactions on Evolutionary Computation, vol. 9, no. 5, October 2005.

[7] W. E. Hart, “Adaptive Global Optimization With Local Search,” Ph.D. dissertation, Univ. California, San Diego, Ca, 1994.

[8] C. Houck, J. Joines, M. Kay, and J. Wilson, “Empirical investigation of the benefits of partial Lamarckhianism,” Evol. Comput., vol.5, no. 1, pp. 31-60, 1997.

[9] L. Davis, “Adapting operator probabilities in genetic algorithms,” in Proc. 3rd Int. Conf. Genetic Algorithms, Ed.,1989, pp. 61-69.

[10] B. Luca and I. Petrila, “Construction and Deconstruction of Checkers Game using ICE Agent Model,” in Proceedings of the 20th International Conference on System Theory, Control and Computing (ICSTCC), October 13-15, Sinaia, Romania, 2016.

[11] B. Luca, “Information-Computation-Evolution Agent Model for Natural Computation”, in Proceedings of the 18th International Conference on System Theory, Control and Computing (ICSTCC), October 17-19, Sinaia, Romania, 2014.

[12] P. Merz, “Memetic algorithms for combinatorial optimization problems: Fitness landscapes and effective search strategies,” Ph.D. dissertation, Parallel Syst. Res. Group, Dept. Elec. Eng. Comput. Sci., Univ. Siegen, Siegen, Germany, 2000.

[13] D. Goldberg, The Design of Innovation: Lessons From and for Competent Genetic Algorithms. Norwell, MA: Kluwer, 2002.

[14] D. Fred, “Laws of Nature,” in Philosophy of Science, 1977, vol. 44, no.2, pp. 248-268.

[15] N. Krasnogor, “Studies on the theory and design space of memetic algorithms,” Ph.D. dissertation, Univ. West of England, Bristol, U.K., 2002. [Online]. Available: http://www.cs.nott.ac.uk/ nxk/papers.html.

[16] N. Krasnogor and J. Smith, “A memetic algorithm with self-adaptive local search: TSP as a case study,” in Proc. Genetic Evol. Comput. Conf., D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, and H.-G. Beyer, Eds., 2000, pp. 987994.

[17] N. Krasnogor and D. Pelta, “Fuzzy memes in multimeme algorithms: A fuzzy-evolutionary hybrid,” in Fuzzy Sets Based Heuristics for Optimization, J. Verdegay, Ed. Berlin, Germany: Springer-Verlag, 2002, pp. 4966.