An Immune Metaheuristics for Large Instances of the Weighted Feedback Vertex Set Problem

Vincenzo Cutello, Maria Oliva, Mario Pavone, Rocco A. Scollo

***Abstract*— An immune metaheuristic has been developed for solving the Weighted Feedback Vertex Set problem, known to be a $\mathcal{NP}$-complete problem, which finds applicability in many real-world problems. The algorithm takes inspiration by the immune system, and it is based on three main immune operators, such as cloning, hypermutation and aging. In addition to these operators a local search has been also designed with the goal to refine in deterministic way all solutions produced by the stochasticity of these operators. This local search has proved to be fruitful and effective, improving considerably both the performances of immune algorithm and its learning ability. For evaluate the robustness and efficiency of the proposed algorithm several experiments have been performed on a total of 60 graph instances of different large dimensions (from 100 to 529 vertices). Each of these instances shows different topologies; different problem dimensions; different graph density; and different weights on the vertices. The algorithm has been also compared with other three metaheuristics that represent the state-of-the-art on this problem: (1) memetic algorithm based on a genetic algorithm (MA); (2) tabu search metaheuristic (XTS); and (3) iterative tabu search (ITS). From this comparison emerges that the immune metaheuristic has been able to reach the global best solution on 46 instances, unlike of MA that instead found it in only 23 over 60. Furthermore, interestingly that the proposed immune algorithm, among the 46 instances, was able to improve the results with respect to the well known values on 25 instances of these.**

I. INTRODUCTION

Metaheuristics have proven to be successful established techniques able to solve complex and hard optimization problems, which arise in many daily real-life activities and in many application domains, thanks their ability in efficiently exploring large solution search spaces [20]. Today they constitute a highly diverse family of optimization algorithms, each of which shows individual properties, different strengths, and own historical background. Some are iterative extensions of sophisticated algorithms (e.g. greedy heuristics or local search) whilst most are inspired by natural processes; however, each of them includes a mechanism for jumping out from local optima.

Immune computation is one of the most active population-based metaheuristics, successful applied in searching and optimization tasks [5], [21], which take inspiration by the principles and dynamics of the immune system of the living things. It besides being the most reliable security system against potential diseases, they represent a great source of inspiration due to their ability in detect, recognise, distinguish, and remember foreign entities to the living organism, as well as efficient learning skill [12]. Among all the immune inspired algorithms, those that have proved better reliability and robustness performances are the ones that mimic the clonal selection principle, called *Clonal Selection Algorithms* ($CSAs$) [18], whose key feature is the proliferation and differentiation of all those cells that better detect and recognize foreign entities: the Antigen (Ag). The efficiency and effectiveness of this class of algorithms, primarily in optimization field, is experimentally supported by their application in several scientific areas [8], [7], [9], and by theoretical analyses [26], [14], [24] that prove how $CSAs$ are more reliability with respect many randomize search heuristics.

One of the most interesting combinatorial optimization problems is the *Feedback Vertex Set* (FVS) problem, whose resolution consists in removing from an undirected and cyclic graph the minimal number of vertices that makes the graph acyclic. The interest in this challenging problem has grown over the last years thanks to its applicability in many real-world problems, such as primarily, in preventing or removing deadlocks [23]; and information security [13]. However, own in combinatorial optimization field the research in meta-heuristics has lately shifted towards a their hybridization, combining them with components or concepts from other different optimization techniques (e.g. local search; mathematical programming; constraint programming, etc.) [1]. The aim behind such hybridization is improve some, or all generated solutions so far, because the evolutionary search operators are not able to fine-tuning the solutions due to their stochastic nature.

In light of the above, in this research paper we present a hybrid immune metaheuristics, hereafter simply called HYBRID-IA, which has been designed and developed for solving the weighted variant of the FVS problem, and specifically large-scale instances. HYBRID-IA is based on three immune operators - *cloning*, *hypermutation*, and *aging* - that allow the algorithm to carefully explore the search space, avoiding to get trapped into local optima, and perform a properly exploit the information learned, with the addition also of a local search technique that deterministically trying to refine the solutions removing those vertices in the graph that break off one or more cycles.

Several experiments have been performed on large instances in order to evaluate and prove the reliability and robustness of HYBRID-IA, as well as the goodness of the local search designed and included into the algorithm. From the analysis of the all outcomes, it is possible to assert

V. Cutello, M. Oliva, M. Pavone (*Member, IEEE*), R. A. Scollo are with the Department of Mathematics and Computer Science, University of Catania, viale A. Doria 6, I-95125 Catania, Italy (email: cutello@unict.it, mpavone@dmi.unict.it).

that the immune metaheuristic presented is competitive with the state-of-the-art on this problem, outperforming often the compared algorithms, being able to reach better solution values, and, consequently, improving and updating the best-known solutions found so far. All instances tested, the experimental protocol used and the three metaheuristics used for the comparisons (currently the state-of-the-art) have been taken from [3].

Finally, it is worth emphasizing the positive affect that the designed local search has on the performances and, then, on the convergence of HYBRID-IA, driving it towards optimal solutions, or, in general, towards best solutions than the well-known ones so far. This fruitful impact of the local search emerges also in the learning process of HYBRID-IA, helping it to learn as much information as possible.

II. THE WEIGHTED FEEDBACK VERTEX SET PROBLEM

Feedback Vertex Set problem is one of the most interesting combinatorial optimization tasks that plays a crucial role in many real-world applications. It simply consists in removing a minimal number of vertices from a cyclic graph in order to make it acyclic. In light of this, for instance, solving the FVS plays a crucial role in the study of deadlock detection and recovery in operating systems, since resolving deadlocks means aborting, i.e. removing, the various blocked processes. Other real-life applications of FVS can be also considered in combinatorial circuit design [15]; in information security [13]; and in the study of monopolies in synchronous distributed systems [19].

Given a directed or undirected graph $G = (V, E)$, a feedback vertex set of G is a subset $S \subset V$ of vertices whose removal makes G acyclic. More formally, if $S \subset V$ we can define the subgraph $G[S] = (V \backslash S, E_{V \backslash S})$ where $E_{V \backslash S} = \{(u, v) \in E : u, v \in V \backslash S\}$. If $G[S]$ is acyclic, then S is a feedback set. The problem asks, therefore, to find a feedback vertex set of minimal cardinality. If S is a feedback set, we say that a vertex $v \in S$ is *redundant* if the induced subgraph $G[S \backslash \{v\}] = ((V \backslash S) \cup \{v\}, E_{(V \backslash S) \cup \{v\}}, w)$ is still an acyclic graph. It follows that S is a minimal FVS if it doesn't contain any redundant vertices. If we associate a positive weight $w(v)$ to each vertex $v \in V$, let S be any subset of V, then its weight is the sum of the weights of its vertices, i.e. $\sum_{v \in S} w(v)$. The *Weighted Feedback Vertex Set Problem* ($WFVS$) is then the problem of finding a feedback vertex set of minimal weight.

The problem of finding a feedback vertex set of minimal cardinality, or minimal weight, is known to be a $\mathcal{NP}$-complete problem [16], [25]. Many heuristics and metaheuristics have been developed for the simple FVS problem, whilst very few for its weighted variant.

III. THE IMMUNE METAHEURISTICS

The immune-inspired metaheuristics represents nowadays an established family of algorithms that take inspiration by the dynamics and processes of the immune system. There exist several immune theories with which the immune system defends the living organism against potentially dangerous entities, such as negative selection, immune networks, danger theory, but the best known, likely, is the one based on the clonal selection principle: all those cells able to detect and recognise the Ags (i.e. foreign entities) will proliferate proportionally to their recognition capacity, and undergone to a hypermutation but this time in an inversely proportional way to the same capacity. This principle is at the basis of the proposed algorithm: HYBRID-IA. For this class of algorithms, and hence for HYBRID-IA, the key operators are, respectively, the cloning, hypermutation and aging: the first mimics the proliferation of the cells, with the goal to generate new populations centered on higher affinity values; the second one instead has the purpose to carefully explore the neighborhood of each solution into the search space; and, the last plays the central role to help the algorithm in jumping out from local optimal, by remove old and less promising cells from the population, intensifying thus the variability between the solutions. In addition to these immune operators, a local search mechanism has been also included, whose main aim is to refine the stochastic choices done by the evolutionary process trying thus to fine-tuning of solutions.

HYBRID-IA is based on a population of B cells, where each cell represents a candidate solution to the problem tackled. Specifically for this problem, the B cell is a permutation of vertices that determines the visiting order of the nodes are to be removed. The vertices removal process starts from the first node of the permutation and it is iteratively repeated, following the order in the permutation, until an acyclic graph is obtained. However, an exception is done for those vertices in the permutation with degree 1, since they cannot be involved in any cycle, and, consequently, they appear to be irrelevant. In this case, the vertex in the permutation is simply skipped. At the end of removal process, that is when an acyclic graph is obtained, can be happen that one or more removed vertices are redundant. If so, these vertices are restored in V. Now, the set S of all removed vertices represents a solution to the problem, and the sum of the weights of its vertices is the fitness value associated to the permutation.

The scheme of HYBRID-IA is shown in the pseudocode in Algorithm 1. The algorithm is based on 4 main user-defined parameters, and respectively: population size (d); duplication factor (dup); mutation rate (ρ); and maximum age allowed for remaining into the population (τ_B). At the initial time-step ($t = 0$), the population of the solutions is created and initialized generating each permutation in random way ($3rd$ line of the pseudocode). After that, the fitness function of each generated solution is computed via the function Compute_Fitness($P^{(t)}$) ($4th$ line in Algorithm 1). Once generated and evaluated the initial population, begins the work by the main immune operators, in iterative way until a termination criterion is satisfied. In this research work a maximum number of generations ($maxgen$) has been considered for all experiments performed and included in this paper.

Algorithm 1 Pseudo-code of HYBRID-IA

1: **procedure** HYBRID-IA(d, dup, ρ, τ_B)
2: $t \leftarrow 0$;
3: $P^{(t)} \leftarrow$ Initialize_Population(d);
4: Compute_Fitness($P^{(t)}$);
5: **while** $\neg$ StopCriterion **do**
6: $P^{(clo)} \leftarrow$ Cloning ($P^{(t)}, dup$);
7: $P^{(hyp)} \leftarrow$ Hypermutation($P^{(clo)}, \rho$);
8: Compute_Fitness($P^{(hyp)}$);
9: $(P_a^{(t)}, P_a^{(hyp)}) \leftarrow$ Aging($P^{(t)}, P^{(hyp)}, \tau_B$);
10: $P^{(select)} \leftarrow (\mu + \lambda)$-Selection($P_a^{(t)}, P_a^{(hyp)}$);
11: $P^{(t+1)} \leftarrow$ Local_Search($P^{(select)}$);
12: Compute_Fitness($P^{(t+1)}$);
13: $t \leftarrow t + 1$;
14: **end while**
15: **end procedure**

The first immune operator to take place is the Cloning operator ($6th$ line in Algorithm 1) that simply copies dup times each B cell producing an intermediate population $P^{(clo)}$ of size $d \times dup$. Unlike to the natural immune system a proportional cloning is not considered but rather an its static version. From an algorithmic perspective, this is due because, producing a greater number of the best solutions, in a short time, we will have a population of B cells very similar to each other, and therefore not able to perform a good exploration of the search space, consequently leading the algorithm to a premature convergence towards local optimal. Once created the copies of any B cell, to each of those is assigned an age that determines how long they can live into the population during which they can mature, learn and improve. It follows that each cloned B cell will evolve for producing better and more robust offspring until reaching the maximum age allowed (τ_B), starting just from its assigned age. The age assignment together aging operator (described below) play a crucial role on the performances of HYBRID-IA, and any evolutionary algorithm in general, because they are able to keep high and proper diversity among the solutions, avoiding thus premature convergences [11], [22]. In this research work, to each clone is assigned a random age chosen in the range $[0 : \frac{2}{3}\tau_B]$. With this age assignment, each clone is guaranteed to live for at least a fixed number of generations for evolving and learning ($\frac{1}{3}\tau_B$ in the worst case).

The Hypermutation operator ($7th$ line in pseudocode) acts on each clone in $P^{(clo)}$ with the main goal to explore its neighborhood taking into account the goodness, or less, of the objective function of the clone it is working on. Unlike of any evolutionary algorithm, this operator works with no mutation probability and it performs M mutations on each B cell. Such M number is determined in *inversely proportional* way to the fitness value of the considered B cell: the better is the fitness value the less are the mutations performed; the worse the fitness the more mutations are. The combination of the Cloning and Hypermutation operators allow to perform a careful diversify local exploration on different points of the search space in order to produce offspring with better fitness values, to then be selected to form ever better progenies.

Let $\vec{x}$ a cloned B cell, the number of mutations is determined by $M = \lfloor(\alpha \times \ell) + 1\rfloor$, with ℓ the length of the B cell (i.e. $\ell =| V |$), and α representing the *mutation rate* obtained as $\alpha = e^{-\rho \hat{f}(\vec{x})}$, where ρ determines the shape of the mutation rate (user-defined parameter), and $\hat{f}(\vec{x})$ is the fitness function normalized in the range $[0, 1]$. The optimization idea behind HYBRID-IA is fundamentally based on finding the best permutation such that the removal of its first k vertices produce an acyclic graph without redundant vertices. Becomes, in this way, crucial which position each node occupies in the permutation. In light of this, the most suitable hypermutation operator to perform is the classical *Swap Mutations*: for any $\vec{x}$ B cell, chosen two positions i and j, the vertices x_i and x_j are exchanged of position, becoming $x'_i = x_j$ and $x'_j = x_i$, respectively.

Once performed the hypermutation and evaluated the quality of the produced solutions ($8th$ line), the aging operator ($9th$ line in pseudocode) acts on each hypermutated clones removing all old B cells from the two populations $P^{(t)}$ and $P^{(hyp)}$. Basically, once the age of a B cell exceeds the maximum age allowed (τ_B), it will be removed from the population of belonging independently from its fitness value. The parameter τ_B, then, represents the maximum number of generations allowed to every B cell for remaining and maturing into the population. The main goal of this operator is to allow the algorithm in escaping, and jumping out from local optima. Indeed this approach guarantees a proper turnover between the B cells in the population, producing high diversity among them, and consequently helps the algorithm to avoid premature convergences and get trapped into local optima. However, an exception about the removal may be done for the best current solution, which is kept into the population even if its age is older than τ_B. This variant of the aging operator is called *elitist aging operator*.

After the Aging operator, the best d survivors from both populations $P_a^{(t)}$ and $P_a^{(hyp)}$ are selected for generating the temporary population $P^{(select}$ that, afterwards, will be undergo to the local search. At this end, the $(\mu + \lambda)$-*Selection operator* has been developed, where $\mu = d$ and $\lambda = (d \times dup)$. Such operator guarantees monotony in the evolution dynamics, because it identifies the d best elements between the offspring set and the old parent B cells. However may happens that the survivors are less than the required d population size, that is $d_a < d$; in this case $d - d_a$ new B cells are randomly generated. Of course, this scenario depends, and is strictly related to the clone's age assignment, and maximum age allowed for living into the population (τ_B).

The last operator to be run is the local search that has the main purpose to refine and improve the best B cells selected. The main idea behind this operator is to repair in deterministic way the solutions produced by the stochastic mutation operator restoring one or more cycles and check if there exist deterministically an better alternative solution for breaking off the same cycle or the several cycles restored before. In fact, because each solution obtained by the hypermutation operator is produced by a randomly generated permutation that doesn't take into account any relation to the weights, the local search deterministically tries to replace one node in the solution with one or more nodes of lesser weight, belonging to the same cycle. Given a solution $\vec{x}$, all vertices in $\vec{x}$ are sorted in decreasing order with respect their weights. Then, starting from vertex u with largest weight, the local search procedure iteratively works as follows: the vertex u is inserted again in $V \setminus S$, generating then one or more cycles; via the classical DFS (Depth First Search) procedure [6], are computed the number of the cycles produced, and, thus one of these (if there are more) is taken into account together with all vertices involved in it. These last vertices are now sorted in increasing way with respect their weight-degree ratio; thus, the vertex v with smaller ratio is selected to break off the cycle. Of course, may also happens that v break off also more cycles. Anyway, if from the removal of v the subgraph becomes acyclic, and this removal improves the fitness function then v is considered for the solution and it replaces u; otherwise the process is repeated again taking into account the remaining vertices in the cycle. At the end of the iterations, if the sum of the new removed vertices (i.e. the fitness) is less to the previous one, then such vertices are inserted in the new solution in place of vertex u.

IV. RESULTS AND COMPARISONS

In this section all outcomes obtained and all comparisons done are presented, in order to evaluate the efficiency, robustness and reliability of HYBRID-IA, as well as the efficacy of the local search developed. For this kind of evaluation a set of benchmarks has been considered, proposed in [4], which includes instances with different topology of graph; different graph dimension ($|V|$); different graph density, and different range values for the vertex weights. In particular, the benchmark includes *toroidal*, *hypercube*, and *random* topology graphs. Note that in this research work, only the larger instances of them have been taken into account ($100 \leq |V| \leq 529$). These differences help us in evaluate the performances of HYBRID-IA in different scenarios, and different optimization conditions (density of the graph and weight ranges). Of course, this benchmark helps us also in evaluate the competitiveness of the proposed algorithm with respect the state-of-the-art [3]. Three different algorithms have been considered for the comparisons, which represent nowadays the state-of-the-art on this kind of problem: *Iterated Tabu Search* (ITS) [4]; *eXploring Tabu Search* (XTS) [10], [2]; and *Memetic Algorithm* (MA) [3]. The outcomes for these three algorithms, reported in the tables below, have been taken by [3]. For all experiments presented in this section, the parameters setting of HYBRID-IA were, respectively: population size $d = 100$; duplication parameter $dup = 2$; maximum age reachable $\tau_B = 20$; mutation rate parameter $\rho = 1.3$ if $|V| = 100$, and $\rho = 3.0$ otherwise. The values of these parameters have been determined through experimental tests and historical knowledge learned on the algorithm. Further, following the experimental protocol proposed in [3], each value reported in tables I and II is the average value obtained on five different scenarios of the same instance, each based with different assignment of weights on the vertices. In turn, each of the five scenarios was performed on 10 independent runs, and for $\{1000, 1250, 1500, 1750, 2000\}$ generations, respectively for $\{100, 200, 300, 400, 500\}$ nodes. It is important to highlight that in [3] any fixed number of generations is not given, but rather the authors consider as stop criterion $MaxIt$ consecutive generations without improvement, where the $MaxIt$ value is determined via a formula related to the density, and size of the graph ([3], subsection §4.1). Note that this threshold is reset every time an improvement occurs. From a simple calculus, the number of generations used in our experiments are almost always lowest or equal to the minimum ones performed by the algorithms presented in [3], considering specially the simplicity in the improvements of the fitness in the first steps of the generations.

Figure 1, left plot, shows the convergence behaviour of HYBRID-IA, in which are displayed the curves, respectively, of the best fitness, average fitness of the population, and average fitness of the cloned hypermutated population. The reported curves have been computed on the random instance L_R15, whose features are shown in table I. From this plot is possible to see how HYBRID-IA shows a very good convergence towards the (likely) optimal solution. Note that HYBRID-IA on this instance reaches a best solution value than the one found by the other three algorithms (see table I), and, consequently, than the currently best-known. This plot shows also how all three curves keep a right distance from each other, and this confirm us the existence of a good diversity degree among the solutions, useful for avoiding and/or escaping from local optima. However, after around 500 (or a little less) generations, the curve of the best fitness and the average fitness one meet in a point, carrying on in a common line: this is the time when HYBRID-IA has already reached (some generations before) the global best solution. Moreover, the achievement of the global best solution helps also the creation of better offspring: the curve decreases significantly in the same point as the others. The right plot in figure 1 shows the impact and effective of the Local Search on the performance of HYBRID-IA. In particular, the curves of the average fitness produced by HYBRID-IA as described above, and HYBRID-IA with turn off the Local Search are displayed. Analysing the plot, it is clear how the use of the Local Search helps the algorithm in improving considerably the performances, continuing to help the algorithm in improving even after 400 generations, albeit slightly. From the analysis of this plot, it is then possible to assert the goodness and usefulness produced by the local search, which is able to improve the solutions generated by the stochastic immune operators. The large distance between the two curves further proves this.

Once analysed the convergence behaviour, also a study on the learning ability of HYBRID-IA has been performed: how many information the algorithm is able to discover during the evolutionary process? At this end, the classical entropy

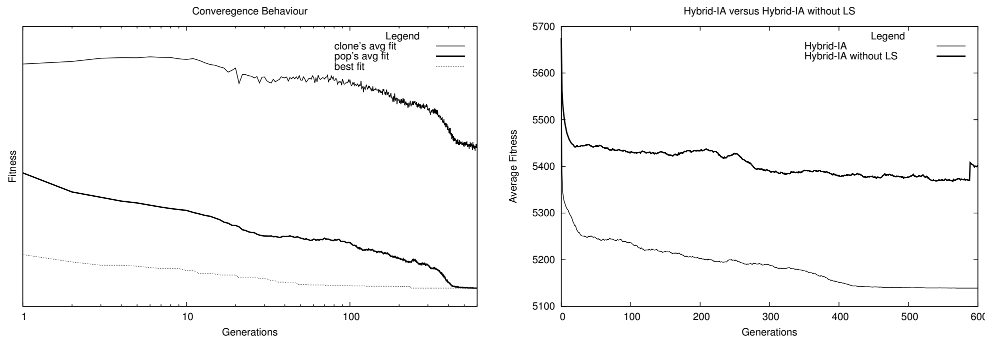

Fig. 1. Convergence behavior of the average fitness function values of $P^{(t)}$, $P^{(hyp)}$, and the best B cell versus generations on the random L_R15 instance (left plot). Average fitness function of $P^{(select)}$ vs. average fitness function $P^{(t)}$ on the random L_R15 instance (right plot).

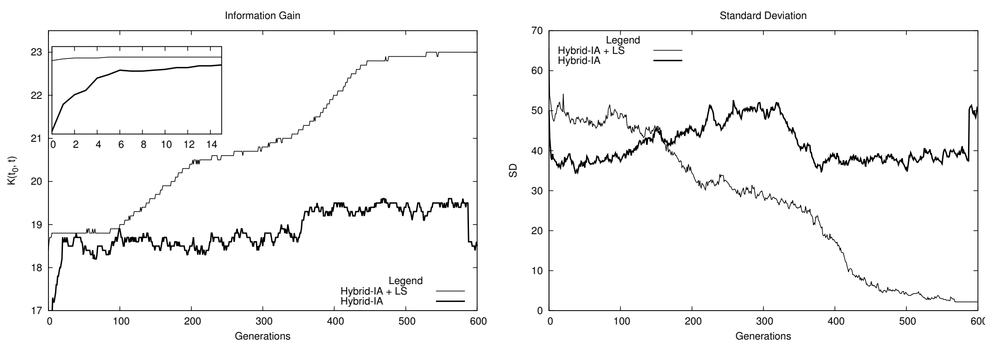

Fig. 2. Learning during evolutionary process: information gain (left plot) and standard deviation (right plot) curves produced by Hybrid-IA with, and without local search strategy.

function, *Information Gain*, has been used for measuring the quantity of information the algorithm gains during the learning phase [17], [7]. Let B^t_m be the number of the B cells that at the timestep t have the fitness function value m; we define the candidate solutions distribution function $f^{(t)}_m$ as the ratio between the number B^t_m and the total number of candidate solutions:

$$f^{(t)}_m = \frac{B^t_m}{\sum_{m=0}^{h} B^t_m} = \frac{B^t_m}{d}. \tag{1}$$

It follows that the information gain $K(t, t_0)$ and entropy $E(t)$ can be defined as:

$$K(t, t_0) = \sum_m f^{(t)}_m \log(f^{(t)}_m / f^{(t_0)}_m), \tag{2}$$

$$E(t) = \sum_m f^{(t)}_m \log f^{(t)}_m. \tag{3}$$

The gain is the amount of information the system learned compared to the randomly generated initial population $P^{(t=0)}$. Once the learning process begins, the information gain increases until to reach a peak point. The two plots in figure 2 show part of the analysis conducted on the study of the learning process quality of HYBRID-IA. This analysis helps us also in understand the goodness of the local search from a learning point of view. The left plot shows the information gain produced by the algorithm with and without the using of the local search. It appears quite clear how the local search helps HYBRID-IA learn more information, showing an increasing curve until to reach a steady-state (*maximum information gain principle*), which corresponds to the time when the global best solution is reached, and when all solutions begins to become similar (i.e. when the curve of the best fitness and the one of the average fitness meet each other; see figure 1 left plot). Disabling the local search emerges instead that the algorithm is not able to gain any information. In fact its curve doesn't show an increasing behavior (typical for a good learning), but rather an oscillatory one: it learns a bit of information but immediately after loses it. The inset plot shows a zoom on the first 15 generations. In the right plot is shown the comparison between the standard deviation obtained with and without the use of local search. Also for this plot, it is enough clear as the use of the local search helps the algorithm to have less uncertainty (less dispersion), showing a decreasing behavior until to reach the minimal value, which correspond at the same time the algorithm reaches its maximum information gain value. Thus, HYBRID-IA achieves its maximum learning at the same time as it shows less uncertainty, and at the same time it achieves the global best solution. This proves therefore not only the efficiency of the local search developed, but, in the overall, the goodness and reliability of the HYBRID-IA's performances. Reflecting the behavior of information gain, turning off the local search produce a high uncertainty, since the standard deviation is kept on high values.

In the overall, inspecting all dynamic behavior figures it is possible to assert how the designed local search affects considerably on the performances of the *Hybrid-IA* and on the quality of the information discovered and gained, which support the algorithm in the searching of the global best solution. Thanks to a careful exploration of the search space performed by the hypermutation operator, and an useful and successful refinement of the solutions carrying out by the local search, HYBRID-IA has proved to be a robust and reliable combinatorial optimization algorithm.

To further confirm this last statement, the experimental outcomes and the comparisons performed with respect the state-of-the-art are reported in the tables I and II. For this kind of experiments, only the large instances of the benchmark proposed in [3] have been considered. Moreover, HYBRID-IA has been compared with other three different metaheuristics: ITS [4], XTS [10], [2], and MA [3]. Of course, as written above, for having a correct comparison the same experimental protocol used in [3] has been considered. Three different set of instances have been considered for our experiments: *random* (table I), *hypercube* and *toroidal* (table II) graphs. Both tables report in each column, respectively, the name of the instance; number of vertices; number of edges; lower and upper bounds of the vertex weights; and best solution value known so far (K^*). In the other columns the results of HYBRID-IA and of the other three compared metaheuristics are reported, respectively. For any instance the best result obtained among all 4 compared algorithms is highlighted in boldface.

The comparisons on the random graphs are reported in table I. For this comparison instances with different problem dimensions have been taken into account (from 100 to 400 nodes); with different density of edges; and different range of weights. This helps us in testing the efficiency of HYBRID-IA on different degrees of problem complexity, and different optimization scenarios. Inspecting the table I, it is possible to assert that the proposed immune metaheuristic outperforms all compared algorithms, reaching best solution in 27 instances over 36, and improving the best-known value so far (i.e. K^*) in 11 of these instances. The MA algorithm, which is based on a genetic algorithm, is instead able to reach the best solution in 17 instances over 36, and only in 5 of them it is able to find better solutions than HYBRID-IA, and the other two algorithms. About the other two algorithms, XTS is able to reach the best solutions in 16 instances, and in 4 of these it reaches better solutions than the other compared algorithms; whilst ITS finds the best solution only on 2 instances. It is important to emphasize that: (1) in all those instances where HYBRID-IA reaches better values than K^*, the improvement is almost always quite considerable; (2) in those instances where HYBRID-IA is not able to find the best solutions, it is, anyway, always the second best; (3) if HYBRID-IA is compared to MA in all those instances where both are not able to find the best solution, i.e. where XTS is the winner, anyway the proposed immune metaheuristic find always better solutions than MA.

In table II are reported the outcomes on the hypercube and toroidal graphs. These instances present graphs dimension from 100 to 512 nodes, and from 100 to 529 vertices, respectively. For the first set of instances in this table, it is quite clear how HYBRID-IA reaches the optimal solution in all instances (9 instances in the overall), improving the best-known value in 6 of these instances. Also on the toroidal graphs, is possible to see how HYBRID-IA outperforms the compared algorithms finding the best solution in 10 instances over 15, unlike of MA that is able to reach the best solution only in 7 instances. XTS and ITS, instead, never have been able to reach the best solutions in any instance, except in L_T3 instance for XTS. Moreover, whilst MA finds better solutions than HYBRID-IA in 5 instances, our immune metaheuristic not only outperforms MA in 8 instances, but it improves significantly the best-known values on them. It is worth to emphasize that, due to the particular topology of the toroidal graphs, the local search finds some difficulties primarily when acting on the selection of a new vertex based on its degree, or weight-degree ratio, since the vertices tend to have all the same degree. A possible heuristic approach is to increase the mutation rate (ρ). In this way, we tend to obtain worse solutions, so local search has a better chance to improve its performance. Some preliminary results, which we have obtained and which are not included in this paper, justify this approach. Indeed, for the toroidal instances L_T7, and L_T8 ($\mid V \mid = 289$, and $\mid E \mid = 578$) setting $\rho = 1.9$, i.e. high mutation rate for each B cell, we have considerably improved the relative results with respect to what reported in table II: L_T7$= 1102.0$; and L_T8$= 1303.4$. In particular, such new results are also better than the best known values on these instances.

Thus, analysing all results of both tables, it is possible to assert that HYBRID-IA is competitive, and very often better than the other compared algorithms, especially with respect to MA algorithm. The comparisons have been done in the overall on a total of 60 different graphs, and at different problem dimensions: HYBRID-IA reached the global best solution in 46 instances, and on 25 of these it found solutions better than the best known values (K^*), whilst MA in only 23. If we focus on the comparison between HYBRID-IA and MA, we have that MA finds better solutions than our algorithm in only 10 instances, unlike of HYBRID-IA that outperforms MA in 35 instances over 60. In the remaining 15 instances both algorithms reach the same solutions that correspond also to the best known. In conclusion, from an overall analysis (convergence behavior, learning ability, and experimental results) it is possible to assert HYBRID-IA as: (i) a robust and reliable optimization algorithm, (ii) competitive with respect other optimization metaheuristics, and, finally, (iii) part of the new state-of-the-art for the Weighted Feedback Vertex Set problem.

V. CONCLUSION

In this research work an immune metaheuristics is presented and designed to tackle and solve one of the most

TABLE I

HYBRID-IA VERSUS MA, ITS AND XTS ON THE SET OF THE *random* INSTANCE GRAPHS.

	INSTANCE n	m	Low	Up	K^*	HYBRID-IA	ITS	XTS	MA
RANDOM GRAPHS									
L_R1	100	247	10	25	498.4	**498.4**	501.4	500.8	**498.4**
L_R2	100	247	10	50	836.8	**833.8**	845.8	840.0	836.8
L_R3	100	247	10	75	1207.6	**1207.2**	1223.8	1208.0	1207.6
L_R4	100	841	10	25	826.8	**826.8**	828.2	**826.8**	**826.8**
L_R5	100	841	10	50	1724.4	**1724.4**	1729.6	1724.6	**1724.4**
L_R6	100	841	10	75	2420.6	**2420.6**	2425.6	**2420.6**	**2420.6**
L_R7	100	3069	10	25	1134.0	**1134.0**	**1134.0**	**1134.0**	**1134.0**
L_R8	100	3069	10	50	2179.0	**2179.0**	**2179.0**	**2179.0**	**2179.0**
L_R9	100	3069	10	75	3228.6	**3228.6**	3228.8	3228.8	**3228.6**
L_R10	200	796	10	25	1468.2	**1466.6**	1488.4	1468.8	1468.2
L_R11	200	796	10	50	2399.0	2400.0	2442.6	2414.4	**2399.0**
L_R12	200	796	10	75	3089.6	**3087.0**	3157.0	3099.6	3089.6
L_R13	200	3184	10	25	1986.2	1987.2	2003.6	1986.8	**1986.2**
L_R14	200	3184	10	50	3650.6	**3649.2**	3683.6	3650.6	3651.8
L_R15	200	3184	10	75	5135.8	**5133.8**	5158.6	5137.2	5135.8
L_R16	200	12139	10	25	2447.8	**2447.8**	2450.0	2448.4	**2447.8**
L_R17	200	12139	10	50	4148.6	**4148.6**	4149.4	**4148.6**	4149.0
L_R18	200	12139	10	75	5528.4	**5528.4**	5531.4	**5528.4**	**5528.4**
L_R19	300	1644	10	25	2045.4	**2044.8**	2072.6	2045.4	2048.0
L_R20	300	1644	10	50	4175.4	4177.4	4239.4	4195.2	**4175.4**
L_R21	300	1644	10	75	6065.2	6072.4	6154.4	6102.8	**6065.2**
L_R22	300	7026	10	25	3203.0	3203.4	3231.0	**3203.0**	3207.6
L_R23	300	7026	10	50	6211.0	6214.2	6261.4	**6211.0**	6217.2
L_R24	300	7026	10	75	8585.4	**8573.2**	8660.6	8585.4	8613.2
L_R25	300	27209	10	25	3726.6	**3726.6**	3729.2	**3726.6**	**3726.6**
L_R26	300	27209	10	50	5734.8	**5734.8**	5738.0	**5734.8**	**5734.8**
L_R27	300	27209	10	75	10467.0	**10467.0**	10469.6	**10467.0**	**10467.0**
L_R28	400	2793	10	25	2989.6	**2987.2**	3015.2	2991.0	2989.6
L_R29	400	2793	10	50	6410.0	6421.6	6528.0	6435.8	**6410.0**
L_R30	400	2793	10	75	8597.2	**8581.2**	8730.0	8637.0	8597.2
L_R31	400	12369	10	25	4428.8	4434.0	4451.8	**4428.8**	4437.4
L_R32	400	12369	10	50	6785.8	6792.4	6837.4	**6785.8**	6800.6
L_R33	400	12369	10	75	10599.4	**10599.2**	10661.8	10599.4	10601.0
L_R34	400	48279	10	25	5060.4	**5060.4**	5060.8	**5060.4**	5060.6
L_R35	400	48279	10	50	7106.8	**7106.8**	7109.2	**7106.8**	7108.0
L_R36	400	48279	10	75	15103.2	**15103.2**	15114.6	**15103.2**	15117.8

challenging combinatorial optimization problems, such as the weighted variant of the feedback vertex set that finds applicability in many real-world tasks. The proposed algorithm, simply called HYBRID-IA, is based on the clonal selection metaphor (proliferation and differentiation of the cells), and it takes advantage of three main immune operators that allow to the algorithm to make a faithful exploration of the search space, avoid to get trapped into local optima, and exploit all information learned as better as possible. Also a Local Search has been developed in HYBRID-IA whose goal is to deterministically refine all solutions produced instead by the stochasticity. Several experiments have been performed for evaluating both efficacy of the developed local search, and the robustness and reliability of HYBRID-IA. A comparison with three other different metaheuristics has been performed, which represent nowadays the state-of-the-art on this problem: *Iterated Tabu Search* (ITS); *eXploring Tabu Search* (XTS); and *Memetic Algorithm* (MA).

In the overall, HYBRID-IA has been run and tested on a total of 60 different graph instances (in topology, dimension, and vertex weights), on which it was able in finding the best solution in 46 of them. Moreover, among these 46 instances, HYBRID-IA has improved the best-known values in 25 of them. An analysis on the convergence and learning process has been also performed in order to understand the dynamics features of HYBRID-IA, as well as how well the local search affects the performances of the presented immune metaheuristic. From these analyses emerges quite clear the goodness of the designed local search, which is able to considerably improve the performances of the algorithm, and allows it to learn much more information. Finally, inspecting all outcomes, from the dynamic behaviors to experimental results, it is possible to assert that HYBRID-IA is a robust, efficient and reliability optimization algorithm.

REFERENCES

[1] C. Blum, G. R. Raidl: "*Hybrid Metaheuristics: Powerful Tools for Optimization*", Artificial Intelligence: Foundations, Theory, and Algorithms, Springer, 2016.

[2] L. Brunetta, F. Maffioli, M. Trubian: "*Solving the Feedback Vertex Set Problem on Undirected Graphs*", Discrete Applied Mathematics, vol. 101, pp. 37-51, 2000.

[3] F. Carrabs, C. Cerrone, R. Cerulli: "*A Memetic Algorithm for the Weighted Feedback Vertex Set Problem*", Networks, vol. 64, no. 4, pp. 339–356, 2014.

TABLE II

HYBRID-IA VERSUS MA, ITS AND XTS ON THE SET OF THE INSTANCES: *hypercube* AND *toroidal* GRAPHS.

INSTANCE	n	m	Low	Up	K^*	HYBRID-IA	ITS	XTS	MA
HYPERCUBE GRAPHS									
L_H1	128	448	10	25	731.8	**731.4**	740.0	742.0	731.8
L_H2	128	448	10	50	1066.8	**1066.8**	1071.0	**1066.8**	1067.2
L_H3	128	448	10	75	1161.6	**1161.6**	1163.6	**1161.6**	1162.4
L_H4	256	1024	10	25	1487.4	**1486.6**	1542.6	1534.2	1487.4
L_H5	256	1024	10	50	2279.6	**2279.4**	2311.4	2282.0	2279.6
L_H6	256	1024	10	75	2572.4	**2572.4**	2590.8	2576.4	**2572.4**
L_H7	512	2304	10	25	3119.0	**3118.8**	3240.8	3146.0	3119.0
L_H8	512	2304	10	50	4852.2	**4843.6**	4921.8	4872.4	4852.2
L_H9	512	2304	10	75	5553.4	**5552.4**	5588.6	5563.8	5553.4
TOROIDAL GRAPHS									
L_T1	10	10	10	25	388.0	**387.8**	388.8	389.0	388.0
L_T2	10	10	10	50	457.6	**457.4**	458.6	457.6	457.6
L_T3	10	10	10	75	504.6	**504.6**	504.8	**504.6**	**504.6**
L_T4	14	14	10	25	748.8	**747.6**	750.8	748.8	749.6
L_T5	14	14	10	50	874.4	**874.2**	875.6	875.4	874.4
L_T6	14	14	10	75	1016.2	**1016.2**	1017.2	1016.4	**1016.2**
L_T7	17	17	10	25	1102.8	1105.2	1110.2	1107.4	**1102.8**
L_T8	17	17	10	50	1304.4	1304.6	1307.6	1306.0	**1304.4**
L_T9	17	17	10	75	1498.6	**1497.6**	1502.4	1499.6	1498.6
L_T10	20	20	10	25	1539.6	1540.2	1548.6	1540.0	**1539.6**
L_T11	20	20	10	50	1795.4	1795.6	1803.4	1797.6	**1795.4**
L_T12	20	20	10	75	2033.0	**2031.8**	2042.6	2033.6	2033.0
L_T13	23	23	10	25	2034.8	2039.0	2043.4	2043.8	**2034.8**
L_T14	23	23	10	50	2406.4	**2406.2**	2412.2	2410.8	2406.4
L_T15	23	23	10	75	2697.2	**2695.4**	2705.4	2704.2	2697.2

[4] F. Carrabs, R. Cerulli, M. Gentili,G. Parlato: "*A Tabu Search Heuristic Based on k-Diamonds for the Weighted Feedback Vertex Set Problem*", Proc. International Conference on Network Optimization (INOC), LNCS 6701, pp. 589–602, 2011.

[5] P. Conca, G. Stracquadanio, O. Greco, V. Cutello, M. Pavone, G. Nicosia: "*Packing Equal Disks in a Unit Square: an Immunological Optimization Approach*", Proc. International Workshop on Artificial Immune Systems (AIS), IEEE Press, pp. 1–5, 2015.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein: "*Introduction to Algorithms, Third Edition*", MIT Press, July 2009.

[7] V. Cutello, G. Nicosia, M. Pavone: "*An Immune Algorithm with Stochastic Aging and Kullback Entropy for the Chromatic Number Problem*", Journal of Combinatorial Optimization, vol. 14, no. 1, pp. 9–33, 2007.

[8] V. Cutello, G. Nicosia, M. Pavone, J. Timmis: "*An Immune Algorithm for Protein Structure Prediction on Lattice Models*", IEEE Transaction on Evolutionary Computation, vol. 11, no. 1, pp. 101–117, 2007.

[9] V. Cutello, G. Nicosia, M. Pavone, I. Prizzi: "*Protein Multiple Sequence Alignment by Hybrid Bio-Inspired Algorithms*", Nucleic Acids Research, Oxford Journals, vol. 39, no. 6, pp. 1980–1992, 2011.

[10] M. DellAmico, A. Lodi, and F. Maffioli: "*Solution of the Cumulative Assignment Problem with a Well-Structured Tabu Search Method*", Journal of Heuristics, vol. 5, no. 2, pp. 123-143, 1999.

[11] A. Di Stefano, A. Vitale, V. Cutello, M. Pavone: "*How Long Should Offspring Lifespan Be in Order to Obtain a Proper Exploration?*", Proc. IEEE Symposium Series on Computational Intelligence (IEEE SSCI), IEEE Press, pp. 1–8, 2016.

[12] S. Fouladvand, A. Osareh, B. Shadgar, M. Pavone, S. Sharafia: "*DENSA: An Effective Negative Selection Algorithm with Flexible Boundaries for SelfSpace and Dynamic Number of Detectors*", Engineering Applications of Artificial Intelligence, vol. 62, pp. 359–372, 2016.

[13] D. Gusfield: "*A Graph Theoretic Approach to Statistical Data Security*", SIAM Journal on Computing, vol. 17, no. 3, pp. 552–571, 1988.

[14] T. Jansen, C. Zarges: "*Computing Longest Common Subsequences with the B-cell Algorithm*", Proc. of $11th$ International Conference on Artificial Immune Systems (ICARIS), LNCS 7597, pp. 111-124, 2012.

[15] D. B. Johnson: "*Finding All the Elementary Circuits of a Directed Graph*", SIAM Journal on Computing, vol. 4, pp. 77–84, 1975.

[16] R. M. Karp: "*Reducibility among Combinatorial Problems*", Complexity of Computer Computations, The IBM Research Symposia Series, Springer, pp. 85–103, 1972

[17] S. Kullback: "*Statistics and Information Theory*", John Wiley and Sons, New York, 1959.

[18] M. Pavone, G. Narzisi, G. Nicosia: "*Clonal Selection - An Immunological Algorithm for Global Optimization over Continuous Spaces*", Journal of Global Optimization, vol. 53, no. 4, pp. 769–808, 2012.

[19] D. Peleg: "*Size Bounds for Dynamic Monopolies*", Discrete Applied Mathematics, vol. 86, pp. 263–273, 1998

[20] E.G. Talbi: " Metaheuristics: from Design to Implementation", John Wiley & Sons, 2008.

[21] Y. Tian, H. Zhang: " *Research on B Cell Algorithm for Learning to Rank Method Based on Parallel Strategy*", PLoS ONE, vol. 11, no. 8, e0157994, 2016.

[22] A. Vitale, A. Di Stefano, V. Cutello, M. Pavone: "*The Influence of Age Assignments on the Performance of Immune Algorithms*", Proc. $18th$ Annual UK Workshop on Computational Intelligence (UKCI), Advances in Computational Intelligence Systems, Advances in Intelligent Systems and Computing series, vol. 840, pp. 16–28, 2018.

[23] C. C. Wang, E. L. Lloyd, M. L. Soffa: "*Feedback Vertex Set and Cyclically Reducible Graphs*", Journal of Association of Computing Machinery (ACM), vol. 32, no. 2, pp. 296–313, 1985.

[24] X. Xia, Z. Yuren: "*On the Effectiveness of Immune Inspired Mutation Operators in Some Discrete Optimization Problems*", Information Sciences, vol. 426, pp. . 87–100.

[25] M. Yannakakis: "*Node-Deletion Problem on Bipartite Graphs*", SIAM Journal on Computing, vol. 10, no. 2, pp. 310–327, 1981.

[26] C. Zarges: "*On the Utility of the Population Size for Inversely Fitness Proportional Mutation Rates*", Proc. of $10th$ ACM SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA), pp. 39-46, 2009.