A Hybrid Immunological Search for the Weighted Feedback Vertex Set Problem

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

Department of Mathematics and Computer Science
University of Catania
V.le A. Doria 6, I-95125 Catania, Italy
cutello@unict.it, mpavone@dmi.unict.it

Abstract. In this paper we present a hybrid immunological inspired algorithm (HYBRID-IA) for solving the Minimum Weighted Feedback Vertex Set ($M\ W\ F\ V\ S$) problem. $MWFVS$ is one of the most interesting and challenging combinatorial optimization problem, which finds application in many fields and in many real life tasks. The proposed algorithm is inspired by the clonal selection principle, and therefore it takes advantage of the main strength characteristics of the operators of (i) cloning; (ii) hypermutation; and (iii) aging. Along with these operators, the algorithm uses a local search procedure, based on a deterministic approach, whose purpose is to refine the solutions found so far. In order to evaluate the efficiency and robustness of HYBRID-IA several experiments were performed on different instances, and for each instance it was compared to three different algorithms: (1) a memetic algorithm based on a genetic algorithm (MA); (2) a tabu search metaheuristic (XTS); and (3) an iterative tabu search (ITS). The obtained results prove the efficiency and reliability of HYBRID-IA on all instances in term of the best solutions found and also similar performances with all compared algorithms, which represent nowadays the state-of-the-art on for $MWFVS$ problem.

Keywords: Immunological algorithms, immune-inspired computation, metaheuristics, combinatorial optimization, feedback vertex set, weighted feedback vertex set.

1 Introduction

Immunological inspired computation represents an established and rich family of algorithms inspired by the dynamics and the information processing mechanisms that the Immune System uses to detect, recognise, learn and remember foreign entities to the living organism [11]. Thanks to these interesting features, immunological inspired algorithms represent successful computational methodologies in search and optimization tasks [4, 20]. Although there exist several immune theories at the basis of immunological inspired computation that could characterize their natural application to anomaly detection and classification tasks, one that has been proven to be quite effective and robust is based on the clonal selection principle. Algorithms based on the clonal selection principle work on a population of immunological cells, better known as *antibodies*, that proliferate, i.e. clone themselves – the number of copies depends on the quality of their

foreign entities detection – and undergo a mutation process usually at a high rate. This process, biologically called *Affinity Maturation*, makes these algorithms very suitable in functions and combinatorial optimization problems. This statement is supported by several experimental applications [7, 6, 8, 18], as well as by theoretical analyses that prove their efficiency with respect to several Randomized Search Heuristics [25, 15, 14, 23].

In light of the above, we have designed and developed an immune inspired hypermutation algorithm - based precisely on the clonal selection principle - in order to solve a classical combinatorial optimization problem, namely the *Weighted Feedback Vertex Set* ($WFVS$).

In addition, we take into account what clearly emerges from the evolutionary computation literature: in order to obtain good performances and solve hard a combinatorial optimization problem, it is necessary to combine together metaheuristics and other classical optimization techniques, such as local search, dynamic programming, exact methods, etc. For this reason, we have designed a hybrid immunological algorithm, by including some deterministic criteria inside in order to refine the found solutions and to help the convergence of the algorithm towards the global optimal solution.

The hybrid immunological algorithm proposed in this paper, hereafter simply called HYBRID-IA, takes advantage of the immunological operators of cloning, hypermutation and aging, to carefully explore the search space and properly exploit the information learned, but it also makes use of local search for improving the quality of the solutions, and deterministically trying to remove that vertex that break off one or more cycles in the graph, and has the minor weight-degree ratio. The many experiments performed have proved the fruitful impact of such a greedy idea, since it was almost always able to improve the solutions, leading the HYBRID-IA towards the global optimal solution. For evaluating the reliability and efficiency of HYBRID-IA, many experiments on several different instances have been performed, taken by [2] and following the same experimental protocol proposed in it. HYBRID-IA was also compared with other three metaheuristics: *Iterated Tabu Search* (ITS) [3]; *eXploring Tabu Search* (XTS) [9, 1]; and *Memetic Algorithm* (MA) [2]. From all comparisons performed and analysed, HYBRID-IA proved to be always competitive and comparable with the best of the three metaheuristics. Furthermore, it is worth emphasizing that HYBRID-IA outperforms the three algorithms on some instances, since, unlike them, it is able to find the global best solution.

2 The Weighted Feedback Vertex Set Problem

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 *Minimum Feedback Vertex Set Problem* ($MFVS$) is the problem of finding 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. It is well known that the

decisional version of the $MFVS$ is a $\mathcal{NP}$-complete problem for general graphs [12, 17] and even for bipartite graphs [24].

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 *Minimum Weighted Feedback Vertex Set Problem* ($MWFVS$) is the problem of finding a feedback vertex set of minimal weight.

The $MWFVS$ problem is an interesting and challenging task as it finds application in many fields and in many real-life tasks, such as in the context of (i) operating systems for preventing or removing deadlocks [22]; (ii) combinatorial circuit design [16]; (iii) information security [13]; and (iv) the study of monopolies in synchronous distributed systems [19]. Many heuristics and metaheuristics have been developed for the simple $MFVS$ problem, whilst very few, instead, have been developed for the weighted variant of the problem.

3 HYBRID-IA: a Hybrid Immunological Algorithm

HYBRID-IA is an immunological algorithm inspired by the clonal selection principle, which represents an excellent example of bottom up intelligent strategy where adaptation operates at local level, whilst a complex and useful behaviour emerges at global level. The basic idea of this metaphor is how the cells adapt for binding and eliminate foreign entities, better known as Antigene (Ag). The key features of HYBRID-IA are the cloning, hypermutation and aging operators, which, respectively, generate new populations centered on higher affinity values; explore the neighborhood of each solution into the search space; and eliminate the old cells and the less promising ones, in order to keep the algorithm from getting trapped into a local optimal. To these immunological operators, a local search strategy is also added, which, by restoring one or more cycles with the addition of a previously removed vertex, it tries again to break off the cycle, or the cycles, by conveniently removing a new vertex that improves the fitness of the solution. Usually the best choice is to select the node with the minimal weight-degree ratio. The existence of a cycle, or more cycles, is computed via the well-know procedure *Depth First Search* (DFS) [5]. The aim of the local search is then to repair the random choices done by the stochastic operators via more locally appropriate choices.

HYBRID-IA is based on two main entities, such as the antigen that represents the problem to tackle, and the B cell receptor that instead represents a point (configuration) of the search space. Each B cell, in particular, represents a permutation of vertices that determines the order of the nodes to be removed: starting from the first position of the permutation, the relative node is selected and removed from the graph, and this is iteratively repeated, following the order in the permutation, until an acyclic graph is obtained, or, in general, there are no more vertices to be removed. Once an acyclic graph is obtained, all possible redundant vertices are restored in V. Now, all removed vertices represent the S set, i.e. a solution to the problem. Such process is outlined in simple way in Algorithm 1. Some clarification about Algorithm 1:

- Input to the Algorithm is an undirected graph $G = (V, E)$;
- given any $X \subseteq V$ by $d_X(v)$ we denote the degree of vertex v in the graph induced by X, i.e. the graph $G(X)$ obtained from G by removing all the vertices not in X.

- if, given any $X \subseteq V$ a vertex v has degree $d_X(v) \leq 1$, such a vertex cannot be involved in any cycle. So, when searching for a feedback vertex set, any such a vertex can simply be ignored, or removed from the graph.
- the algorithm associates deterministically a subset S of the graph vertices to any given permutation of the vertices in V. The sum of the weights of the vertices in S will be the fitness value associated to the permutation.

Algorithm 1 Create Solution

```
X ← V
P ← permutation(V)
S ← ∅
for u ∈ P do
  if u ∈ X and G(X) not acyclic then
    X ← X \ {u}
    S ← S ∪ {u}
    while ∃v ∈ X : d_X(v) < 2 do
      X ← X \ {u}
    end while
  end if
end for
Removing all redundant vertices from S
return S
```

At each time step t, we maintain a population of B cells of size d that we label $P^{(t)}$. Each permutation, i.e. B cell, is randomly generated during the initialization phase ($t = 0$) using a uniform distribution. A summary of the proposed algorithm is shown below. HYBRID-IA terminates its execution when the fixed termination criterion is satisfied. For our experiments and for all outcomes presented in this paper, a maximum number of generations has been considered.

Algorithm 2 HYBRID-IA (d, dup, ρ, τ_B)

```
t ← 0;
P^(t) ← Initialize_Population(d);
Compute_Fitness(P^(t));
repeat
  P^(clo) ← Cloning (P^(t), dup);
  P^(hyp) ← Hypermutation(P^(clo), ρ);
  Compute_Fitness(P^(hyp));
  (P_a^(t), P_a^(hyp)) ← Aging(P^(t), P^(hyp), τ_B);
  P^(select) ← (μ + λ)-Selection(P_a^(t), P_a^(hyp));
  P^(t+1) ← Local_Search(P^(select));
  Compute_Fitness(P^(t+1));
  t ← t + 1;
until (termination criterion is satisfied)
```

Cloning operator: it is the first immunological operator to be applied, and it has the aim to reproduce the proliferation mechanism of the immune system. Indeed it simply copies dup times each B cell producing an intermediate population $P^{(clo)}$ of size $d \times dup$. Once a B cell copy is created, i.e. the B cell is cloned, to this is assigned an age that determines its lifespan, during that it can mature, evolves and improves: from the assigned age it will evolve into the population for producing robust offspring until a maximum age reachable (a user-defined parameter), i.e. a prefixed maximum number of generations. It is important to highlight that the assignment of the age together to the aging operator play a key role on the performances of HYBRID-IA [10, 21], since their combination has the purpose to reduce premature convergences, and keep an appropriate diversity between the B cells.

The cloning operator, coupled with the hypermutation operator, performs a local search around the cloned solutions; indeed the introduction of a high number of blind mutations produces individuals with higher fitness function values, which will be after selected to form ever better progenies.

Inversely hypermutation operator: this operator has the aim to explore the neighborhood of each clone taking into account, however, the quality of the fitness function of the clone it is working on. It acts on each element of the population $P^{(clo)}$, performing M mutations on each clone, whose number M is determined by an *inversely proportional law* to the clone fitness value: the higher is the fitness function value, the lower is the number of mutations performed on the B cell. Unlike of any evolutionary algorithm, no mutation probability has been considered in HYBRID-IA.

Given a clone $\boldsymbol{x}$, the number of mutations M that it will be undergo is determined by the following *potential mutation*:

$$\alpha = e^{-\rho \hat{f}(\boldsymbol{x})}, \tag{1}$$

where α represents the *mutation rate*, and $\hat{f}(\boldsymbol{x})$ the fitness function value normalized in $[0, 1]$. The number of mutations M is then given by

$$M = \lfloor (\alpha \times \ell) + 1 \rfloor, \tag{2}$$

where ℓ is the length of the B cell. From equation 2, is possible to note that at least one mutation occurs on each cloned B cell, and this happens to the solutions very close to the optimal one. Since any B cell is represented as a vertices permutation that determines their removing order, the position occupied by each vertex in the permutation becomes crucial for the achievement of the global best solution. Consequently, the hypermutation operator adopted is the well-known *Swap Mutations*, through which the right permutation, i.e. the right removing order, is searched: for any $\boldsymbol{x}$ B cell, choices 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.

Aging operator: this operator has the main goal to help the algorithm for jumping out from local optimal. Simply it eliminates the old B cells from the populations $P^{(t)}$ and $P^{(hyp)}$: 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. As

written above, the parameter τ_B represents the maximum number of generations allowed so that every B cell can be considered to remain into the population. In this way, this operator is able to maintain a proper turnover between the B cells in the population, producing high diversity inside it, and this surely help the algorithm to avoid premature convergences and, consequently, to get trapped into local optima.

An exception about the removal may be done for the best current solution that 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*.

$(\mu + \lambda)$-Selection operator: once the aging operator ended its work, the best d survivors from both populations $P_a^{(t)}$ and $P_a^{(hyp)}$ are selected for generating a temporary population $P^{(select}$ that, afterwards, will be undergo to the local search. For this process the classical $(\mu + \lambda)$-*Selection operator* has been considered, where, in our case, $\mu = d$ and $\lambda = (d \times dup)$. In a nutshell, this operator reduces the offspring B cell population ($P_a^{(hyp)}$) of size $\lambda \geq \mu$ to a new population ($P^{(select}$) of size $\mu = d$. Since this selection mechanism identifies the d best elements between the offspring set and the old parent B cells, then it guarantees monotonicity in the evolution dynamics. Nevertheless, may happens that all survived B cells are less than the required population size (d), i.e. $d_a < d$. This can easily happens depending on the chosen age assignment, and fixed value of τ_B. In this case, the selection mechanism randomly generates $d - d_a$ new B cells.

Local search: the main idea at the base of this local search is to repair in a proper and deterministic way the solutions produced by the stochastic mutation operator. Whilst the hypermutation operator determines the vertices to be removed in a blind and random way, i.e. choosing them independently of its weight, then the local search try to change one, or more of them with another node of lesser weight, improving thus the fitness function of that B cell. Given a solution $\boldsymbol{x}$, all vertices in $\boldsymbol{x}$ 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 procedure, 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 a new 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.

4 Results

In this section all analysis, experiments, and comparisons of HYBRID-IA are presented in order to measure the goodness of the proposed approach. The main goal of these ex-

periments is obviously to prove the reliability and competitiveness of the HYBRID-IA with respect the state-of-the-art, but also to test the efficiency and the computational impact provided by the designed local search. Thus, for properly evaluating the performances of the proposed algorithm, a set of benchmark instances proposed in [3] have been considered, whose set includes grid, toroidal, hypercube, and random graphs. Each instance considered, besides to the topology, differs in the number of vertices; number of edges; and range of values for the vertices weights. In this way, as suggested in [2], is possible to inspect the computational performances of HYBRID-IA with respect to the density of the graph and weight ranges. Further, HYBRID-IA has been also compared with three different algorithms, which represent nowadays the state-of-the-art for the WFVS problem: *Iterated Tabu Search* (ITS) [3]; *eXploring Tabu Search* (XTS) [9, 1]; and *Memetic Algorithm* (MA) [2]. All three algorithms, and their results, have been taken in [2].

Each experiment was performed by HYBRID-IA with population size $d = 100$, duplication parameter $dup = 2$, maximum age reachable $\tau_B = 20$, mutation rate parameter $\rho = 0.5$, and maximum number of generations $maxgen = 300$. Further, each experiment reported in the tables below is the average over five instances with the same characteristics but different assignment of the vertices weights. The experimental protocol used, and not included above, has been taken by [2]. It is important to highlight that in [2] a fixed maximum number of generators is not given, but the authors use a stop criterion based on a formula ($MaxIt$) that depends on the density of the graph: the algorithm will end its evolution process when it will reach $MaxIt$ consecutive iterations without improvements. Note that this threshold is reset every time an improvement occurs. From a simple calculus, it is possible to check how 300 generations used in this work are almost always lowest or equal to the minimum ones performed by the algorithms presented in [2], considering primarily the simplicity of fitness improvements in the first steps of the generations.

4.1 Dynamic Behaviour

Before to presents the experiments, and comparisons performed, the dynamic behavior and the learning ability of HYBRID-IA are presented. An analysis on the computational impact, and convergence advantages provided by the use of the local search developed has been conducted as well.

In figure 1, left plot, is shown the dynamic behavior of HYBRID-IA, where are displayed the curves of the (i) best fitness, (ii) average fitness of the population, and (iii) average fitness of the cloned hypermutated population over the generations. For this analysis the squared grid instance S_SG9 has been considered, whose features, and weights range are shown in table 1. From this plot is possible to see how HYBRID-IA go down quickly in a very short generations to acceptable solutions for then oscillating between values close to each other. This oscillatory behavior, with main reference to the cloned hypermutated population curve, proves how the algorithm HYBRID-IA has good solutions diversity, which is surely helpful for the search space exploration. These fluctuations are instead less pronounced in the curve of the average fitness of the population, and this is due to the use of the local search that provides greater convergence stability. Last curve, the one of the best fitness, indicates how the algorithm takes the

best advantage of the generations, managing to still improve in the last generations. Since this is one of few instances where HYBRID-IA didn't reach the optimal solution, inspecting this curve we think that likely increasing the number of generations (even just a little), HYBRID-IA will be able to find the global best. Right plot of figure 1

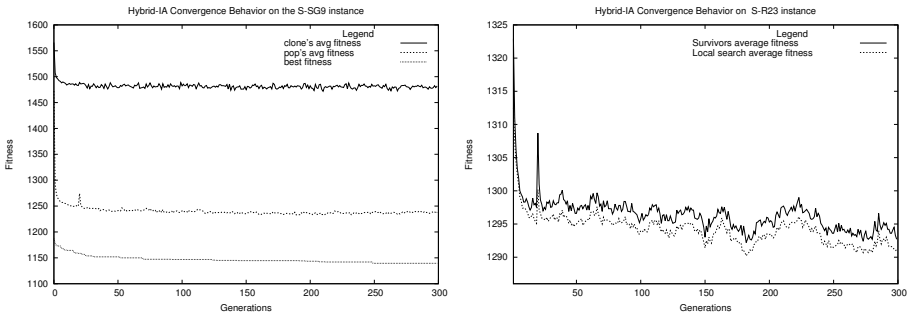

Fig. 1. Convergence behavior of the average fitness function values of $P^{(t)}$, $P^{(hyp)}$, and the best B cell versus generations on the grid S_SG9 instance (left plot). Average fitness function of $P^{(select)}$ vs. average fitness function $P^{(t)}$ on the random S_R23 instance (right plot).

shows the comparison between the average fitness values of the survivors' population ($P^{(select)}$), and the one produced by the local search. Although they show a very similar parallel behavior, this plot proves the usefulness and benefits produced by the local search, which is always able to improve the solutions generated by the stochastic immune operators. Indeed the curve of the average fitness produced by the local search is always below to the survivors one.

Besides the convergence analysis, it becomes important also to understand the *learning ability* of the algorithm, i.e. how many information it is able to gain during all evolutionary process, which affects the performances of any evolutionary algorithm in general. For analyzing the learning process we have then used the well-known entropy function, *Information Gain*, which measures the quantity of information the system discovers during the learning phase [6, 7, 18]. Let B_m^t 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 $f_m^{(t)}$ as the ratio between the number B_m^t and the total number of candidate solutions:

$$f_m^{(t)} = \frac{B_m^t}{\sum_{m=0}^{h} B_m^t} = \frac{B_m^t}{d}. \tag{3}$$

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

$$K(t, t_0) = \sum_m f_m^{(t)} \log(f_m^{(t)} / f_m^{(t_0)}), \tag{4}$$

$$E(t) = \sum_m f_m^{(t)} \log f_m^{(t)}. \tag{5}$$

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.

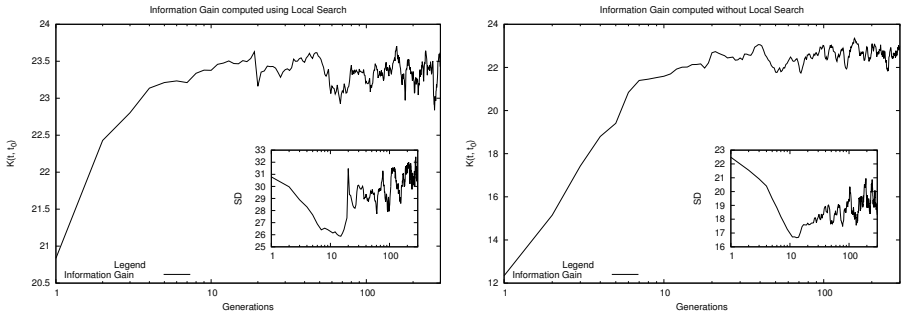

Fig. 2. Learning during the evolutionary process. Information gain curves of Hybrid-IA with local search strategy (left plot) and without it (right plot). The inset plots, in both figures, display the relative standard deviations.

Figure 2 shows the information gain obtained with the use of the local search mechanism (left plot), and without it (right plot) when HYBRID-IA is applied to the random S_R23 instance. In both plots we report also the standard deviation, showed as inset plot. Inspecting both plots is possible to note that with the using of the local search, the algorithm quickly gains high information until to reach the higher peak within the first generations, which exactly corresponds to the achievement of the optimal solution. It is important to highlight that the higher peak of the information gain corresponds to the lower point of the standard deviation, and then this confirm that HYBRID-IA reaches its maximum learning at the same time as it show less uncertainty. Due to the deterministic approach of the local search, once reached the highest peak, and found the global optimal solution, the algorithm begins to lost information and starts to show an oscillatory behavior. The same happens for the standard deviation. From the right plot, instead, without the use of the local search, HYBRID-IA gains information more slowly. Also for this version, the higher peak of information learned corresponds to the lowest uncertainty degree, and in this temporal window HYBRID-IA reaches the best solution. Unlike the curve produced with the use of the local search, once found the optimal solution and reached the highest information peak, the algorithm starts to lost information as well but its behavior seems to be more steady-state.

The two curves of the information gain have been compared and showed in figure 3 (left plot) over all generations. The relative inset plot is a zoom of the information gain behavior over the first 25 generations. From this plot is clear how the local search approach developed helps the algorithm to learn more information already from the first generations, and in a quickly way; the inset plot, in particular, highlight the important existing distance between the two curves. From an overall view over all 300 generations, it is possible to see how the local search gives more steady-state even after reaching

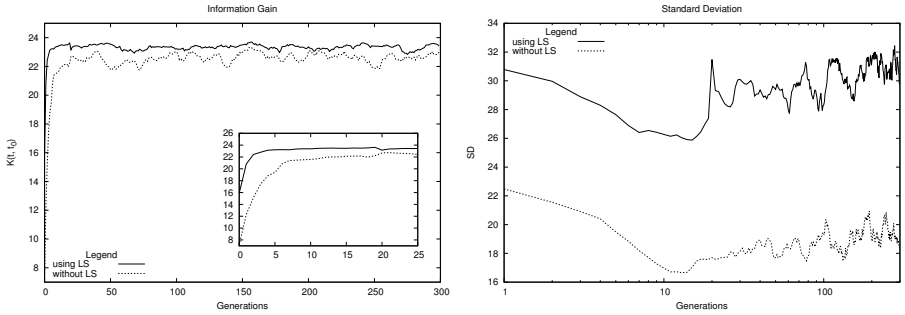

Fig. 3. Information Gain and standard deviation.

the global optimal. In the right plot of the figure 3 is instead shown the comparison between the two standard deviations produced by HYBRID-IA with and without the use of the local search. Of course, as we expected, the curve of the standard deviation using the local search is taller as the algorithm gains higher amount of information than the version without local search.

Finally, from the analysis of all these figures – from convergence speed to learning ability – clearly emerges how the local search designed and developed helps HYBRID-IA in having an appropriate and correct convergence towards the global optimum, and a greater amount of information gained.

4.2 Experiments and Comparisons

In this subsection all experimental results and comparisons done with the state-of-the-art are presented in order to evaluate the efficiency, reliability and robustness of the HYBRID-IA performances. Many experiments have been performed, and the benchmark instances proposed in [2] have been considered for our tests and comparisons. In particular, HYBRID-IA has been compared with three different metaheuristics: ITS [3], XTS [9, 1], and MA [2]. As written in section 4, the same experimental protocol proposed in [2] has been used. Tables 1 and 2 show the results obtained by HYBRID-IA on different sets of instance: *squared grid graph*, *no squared grid graph*, and *toroidal graph* in table 1; *hypercube graph* and *random graphs* in table 2. In both tables are reported: the name of the instance in $1st$ column; number of vertices in $2nd$; number of edges in $3rd$; lower and upper bounds of the vertex weights in $4th$ and $5th$; and optimal solution K^* in $6th$. In the next columns are reported the results of HYBRID-IA ($7th$) and of the other three algorithms compared. It is important to clarify that for the grid graphs (squared and not squared) n and m indicate the number of the rows and columns of the grid. The last column of both tables shows the difference between the result obtained by HYBRID-IA and the best result among the three compared algorithms. Further, in each line of the tables the best results among all are reported in boldface.

On the *squared grid graph* instances (top of table 1) is possible to see how HYBRID-IA is able to reach the optimal solution in 8 instances over 9, unlike of MA that instead reaches it in all instances. However, in this instance (S_SG7) the performances showed

Table 1. HYBRID-IA versus MA, ITS and XTS on the set of the instances: *squared grid*; *not squared grid* and *toroidal* graphs.

	INSTANCE									
	n	m	Low	Up	K^*	HYBRID-IA	ITS	XTS	MA	$\pm$
SQUARED GRID GRAPHS										
S_SG1	5	5	10	25	114.0	**114.0**	**114.0**	**114.0**	**114.0**	=
S_SG2	5	5	10	50	199.8	**199.8**	**199.8**	**199.8**	**199.8**	=
S_SG3	5	5	10	75	312.4	**312.4**	312.6	**312.4**	**312.4**	=
S_SG4	7	7	10	25	252.0	**252.0**	252.4	**252.0**	**252.0**	=
S_SG5	7	7	10	50	437.6	**437.6**	439.8	**437.6**	**437.6**	=
S_SG6	7	7	10	75	713.6	**713.6**	718.4	717.4	**713.6**	=
S_SG7	9	9	10	25	442.2	442.4	444.2	442.8	**442.2**	+0.2
S_SG8	9	9	10	50	752.2	**752.2**	754.6	753.0	**752.2**	=
S_SG9	9	9	10	75	1134.4	**1134.4**	1138.0	**1134.4**	**1134.4**	=
NOT SQUARED GRID GRAPHS										
S_NG1	8	3	10	25	96.8	**96.8**	**96.8**	**96.8**	**96.8**	=
S_NG2	8	3	10	50	157.4	**157.4**	**157.4**	**157.4**	**157.4**	=
S_NG3	8	3	10	75	220.0	**220.0**	**220.0**	**220.0**	**220.0**	=
S_NG4	9	6	10	25	295.6	**295.6**	295.8	295.8	**295.6**	=
S_NG5	9	6	10	50	488.6	**488.6**	489.4	**488.6**	**488.6**	=
S_NG6	9	6	10	75	755.0	**755.0**	**755.0**	755.2	**755.0**	=
S_NG7	12	6	10	25	398.2	**398.2**	399.8	398.8	398.4	−0.2
S_NG8	12	6	10	50	671.8	**671.8**	673.4	**671.8**	**671.8**	=
S_NG9	12	6	10	75	1015.2	**1015.2**	1017.4	1015.4	**1015.2**	=
TOROIDAL GRAPHS										
S_T1	5	5	10	25	101.4	**101.4**	**101.4**	**101.4**	**101.4**	=
S_T2	5	5	10	50	124.4	**124.4**	**124.4**	**124.4**	**124.4**	=
S_T3	5	5	10	75	157.8	**157.8**	**157.8**	158.8	**157.8**	=
S_T4	7	7	10	25	195.4	**195.4**	197.4	**195.4**	**195.4**	=
S_T5	7	7	10	50	234.2	**234.2**	**234.2**	**234.2**	**234.2**	=
S_T6	7	7	10	75	269.6	**269.6**	**269.6**	**269.6**	**269.6**	=
S_T7	9	9	10	25	309.6	**309.8**	310.4	**309.8**	**309.8**	=
S_T8	9	9	10	50	369.6	**369.6**	370.0	**369.6**	**369.6**	=
S_T9	9	9	10	75	431.8	**431.8**	432.2	432.2	**431.8**	=

by HYBRID-IA are very close to the optimal solution (+0.2), and anyway better than the other two compared algorithms. On the *not square grid graphs* (middle of table 1) instead HYBRID-IA reaches the optimal solution on all instances (9 over 9), outperforming all three algorithms on the instance S_NG7, where none of the three compared algorithms is able to find the optimal solution. Also on the *toroidal graphs* (bottom of table 1) HYBRID-IA is able to reaches the global optimal solution in 9 over 9 instances. In the overall, inspecting all results in table 1, is possible to see how HYBRID-

IA shows competitive and comparable performances to MA algorithm, except in the instance S_SG7 where it shows slight worst results, whilst instead it is able to outperform MA in the instance S_NG7 where it reaches the optimal solution unlike of MA. Analysing the results with respect the other two algorithms, is clear how HYBRID-IA outperform ITS and XTS on all instances.

In table 2 are presented the comparisons on the hypercube, and random graphs, which present larger problem dimensions with respect the previous ones. Analysing the results obtained on the *hypercube graph* instances, it is very clear how HYBRID-IA outperform all three algorithms in all instances (9 over 9), reaching even the optimal solution on the S_H7 instance where instead the three algorithms fail. On the *random graphs* HYBRID-IA still shows comparable results to MA on all instances, even reaching the optimum on the instances S_R20 and S_R23 where instead MA, and the other two algorithms fail. In the overall, analyzing all results of this table is easy to assert that HYBRID-IA is comparable, and sometime the best, with respect to MA also on this set of instances, winning even on three instances the comparison with it. Extending the analysis to the comparison with ITS and XTS algorithms, it is quite clear that HYBRID-IA outperforms them in all instances, finding always better solutions than these two compared algorithms.

Finally, from the analysis of the convergence behavior, learning ability, and comparisons performed it is possible to clearly assert how HYBRID-IA is comparable with the state-of-the-art for the $MWFVS$ problem, showing reliable and robustness performances, and good ability in information learning. These efficient performances are due to the combination of the immunological operators, which introduce enough diversity in the search phase, and the local search designed, which instead refine the solution via more appropriate and deterministic choices.

5 Conclusion

In this paper we introduce a hybrid immunological algorithm, simply called HYBRID-IA, which takes advantage by the immunological operators (cloning, hypermutation and aging) for carefully exploring the search space, introducing diversity, and avoiding to get trapped into a local optima, and by a Local Search whose aim is to refine the solutions found through appropriate choices and based on a deterministic approach.

The algorithm was designed and developed for solving one of the most challenging combinatorial optimization problems, such as the *Weighted Feedback Vertex Set*, which simply consists, given an undirected graph, in finding the subset of vertices of minimum weight such that their removal produce an acyclic graph. In order to evaluate the goodness and reliability of HYBRID-IA, many experiments have been performed and the results obtained have been compared with three different metaheuristics (ITS, XTS, and MA), which represent nowadays the state-of-the-art. For these experiments a set of graph benchmark instances has been considered, based on different topologies: *grid* (squared and no squared), *toroidal*, *hypercube*, and *random* graphs.

An analysis on the convergence speed and learning ability of HYBRID-IA has been performed in order to evaluate, of course, its efficiency but also the computational impact and reliability provided by the developed local search. From this analysis, clearly

Table 2. HYBRID-IA versus MA, ITS and XTS on the set of the instances: *hypercube* and *random* graphs.

	INSTANCE									
	n	m	Low	Up	K^*	HYBRID-IA	ITS	XTS	MA	$\pm$
HYPERCUBE GRAPHS										
S_H1	16	32	10	25	72.2	**72.2**	**72.2**	**72.2**	**72.2**	=
S_H2	16	32	10	50	93.8	**93.8**	**93.8**	**93.8**	**93.8**	=
S_H3	16	32	10	75	97.4	**97.4**	**97.4**	**97.4**	**97.4**	=
S_H4	32	80	10	25	170.0	**170.0**	**170.0**	**170.0**	**170.0**	=
S_H5	32	80	10	50	240.6	**240.6**	241.0	**240.6**	**240.6**	=
S_H6	32	80	10	75	277.6	**277.6**	**277.6**	**277.6**	**277.6**	=
S_H7	64	192	10	25	353.4	**353.4**	354.6	353.8	353.8	−0.4
S_H8	64	192	10	50	475.6	**475.6**	476.0	**475.6**	**475.6**	=
S_H9	64	192	10	75	503.8	**503.8**	**503.8**	504.8	**503.8**	=
RANDOM GRAPHS										
S_R1	25	33	10	25	63.8	**63.8**	**63.8**	**63.8**	**63.8**	=
S_R2	25	33	10	50	99.8	**99.8**	**99.8**	**99.8**	**99.8**	=
S_R3	25	33	10	75	125.2	**125.2**	**125.2**	**125.2**	**125.2**	=
S_R4	25	69	10	25	157.6	**157.6**	**157.6**	**157.6**	**157.6**	=
S_R5	25	69	10	50	272.2	**272.2**	**272.2**	**272.2**	**272.2**	=
S_R6	25	69	10	75	409.4	**409.4**	**409.4**	**409.4**	**409.4**	=
S_R7	25	204	10	25	273.4	**273.4**	**273.4**	**273.4**	**273.4**	=
S_R8	25	204	10	50	507.0	**507.0**	**507.0**	**507.0**	**507.0**	=
S_R9	25	204	10	75	785.8	**785.8**	**785.8**	**785.8**	**785.8**	=
S_R10	50	85	10	25	174.6	**174.6**	175.4	176.0	**174.6**	=
S_R11	50	85	10	50	280.8	**280.8**	**280.8**	281.6	**280.8**	=
S_R12	50	85	10	75	348.0	**348.0**	**348.0**	349.2	**348.0**	=
S_R13	50	232	10	25	386.2	**386.2**	389.4	386.8	**386.2**	=
S_R14	50	232	10	50	708.6	**708.6**	**708.6**	**708.6**	**708.6**	=
S_R15	50	232	10	75	951.6	**951.6**	**951.6**	**951.6**	**951.6**	=
S_R16	50	784	10	25	602.0	**602.0**	602.2	**602.0**	**602.0**	=
S_R17	50	784	10	50	1171.8	**1171.8**	1172.2	1172.0	**1171.8**	=
S_R18	50	784	10	75	1648.8	**1648.8**	1649.4	**1648.8**	**1648.8**	=
S_R19	75	157	10	25	318.2	**318.2**	321.0	320.0	**318.2**	=
S_R20	75	157	10	50	521.6	**522.2**	526.2	525.0	522.6	−0.4
S_R21	75	157	10	75	751.0	**751.0**	757.2	754.2	**751.0**	=
S_R22	75	490	10	25	635.8	**635.8**	638.6	**635.8**	**635.8**	=
S_R23	75	490	10	50	1226.6	**1226.6**	1230.6	1228.6	1227.6	−1.0
S_R24	75	490	10	75	1789.4	**1789.4**	1793.6	**1789.4**	**1789.4**	=
S_R25	75	1739	10	25	889.8	**889.8**	891.0	**889.8**	**889.8**	=
S_R26	75	1739	10	50	1664.2	**1664.2**	1664.8	**1664.2**	**1664.2**	=
S_R27	75	1739	10	75	2452.2	**2452.2**	2452.8	**2452.2**	**2452.2**	=

emerges how the developed local search helps the algorithm in having a correct convergence towards the global optima, as well as a high amount of information gained during the learning process.

Finally, from the results obtained, and the comparisons done, it is possible to assert how HYBRID-IA is competitive and comparable with the $WFVS$ state-of-the-art, showing efficient and robust performances. In all instances tested it was able to reach the optimal solutions, except in only one (S_SG7). It is important to point out how HYBRID-IA has been instead the only one to reach the global optimal solutions on 4 instances, unlike the three compared algorithms.

References

1. L. Brunetta, F. Maffioli, M. Trubian: "*Solving the Feedback Vertex Set Problem on Undirected Graphs*", Discrete Applied Mathematics, vol. 101, pp. 37-51, 2000.
2. 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.
3. 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.
4. 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.
5. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein: "*Introduction to Algorithms, Third Edition*", MIT Press, July 2009.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. 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.
12. M. R. Garey, D. S. Johnson: "*Computers and Intractability: A Guide to Theory of NP-Completeness*", Freeman, New York, 1979.
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. T. Jansen, P. S. Oliveto, C. Zarges: "*On the Analysis of the Immune-Inspired B-Cell Algorithm for the Vertex Cover Problem*", Proc. of $10th$ International Conference on Artificial Immune Systems (ICARIS), LNCS 6825, pp. 117-131, 2011.
16. D. B. Johnson: "*Finding All the Elementary Circuits of a Directed Graph*", SIAM Journal on Computing, vol. 4, pp. 77–84, 1975.
17. R. M. Karp: "*Reducibility among Combinatorial Problems*", Complexity of Computer Computations, The IBM Research Symposia Series, Springer, pp. 85–103, 1972
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. 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.
21. 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.
22. 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.
23. X. Xia, Z. Yuren: "*On the Effectiveness of Immune Inspired Mutation Operators in Some Discrete Optimization Problems*", Information Sciences, vol. 426, pp. . 87–100.
24. M. Yannakakis: "*Node-Deletion Problem on Bipartite Graphs*", SIAM Journal on Computing, vol. 10, no. 2, pp. 310–327, 1981.
25. 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.