A Comparative Study of Three Evolutionary Algorithms Incorporating Different Amounts of Domain Knowledge for Node Covering Problem

Jun He, Xin Yao, and Jin Li

***Abstract*—This paper compares three different evolutionary algorithms for solving the node covering problem: EA-I relies on the definition of the problem only without using any domain knowledge, while EA-II and EA-III employ extra heuristic knowledge. In theory, it is proven that all three algorithms can find an optimal solution in finite generations and find a feasible solution efficiently; but none of them can find the optimal solution efficiently for all instances of the problem. Through experiments, it is observed that all three algorithms can find a feasible solution efficiently, and the algorithms with extra heuristic knowledge can find better approximation solutions, but none of them can find the optimal solution to the first instance efficiently. This paper shows that heuristic knowledge is helpful for evolutionary algorithms to find good approximation solutions, but it contributes little to search for the optimal solution in some instances.**

***Index Terms*—Algorithm design, heuristic knowledge, optimization methods, performance analysis.**

I. Introduction

According to the no free lunch theorem [1]–[4], there is no single evolutionary algorithm (EA) which can solve all possible optimization problems efficiently. Over the entire problem space, the performance of all EAs is equivalent. For a given optimization problem, we need to focus on designing problem-oriented algorithms in order to improve the algorithm's performance. The domain knowledge regarding a problem is important in designing efficient EAs. For the same problem, different EAs can be designed based on different knowledge. In order to understand the fundamental role knowledge plays in EAs, it would be useful to analyze and compare the performances of EAs using different amounts of domain knowledge.

For any NP-hard problem, we only have limited heuristic knowledge to use when designing EAs that solve it. In this paper, we aim to discover the role of heuristic knowledge in EAs. Through a comparative study of three different EAs for solving the node covering problem, we investigate the role of heuristic knowledge in EAs finding a feasible solution and the optimal solution.

In this paper, the node covering problem [5], [6] is selected as a case study, as it is a well-known combinatorial optimization problem with practical applications, for example, computer networking and scheduling. The node covering problem is known to be NP-hard and there is no efficient algorithm for it unless $P = \mathrm{NP}$. It is certainly a challenging problem for EAs. EAs have been used to solve this problem [7], [8], and it is claimed that they produced high-quality approximate solutions.

Manuscript received September 3, 2003; revised February 6, 2004. This work was supported in part by the Engineering and Physical Sciences Research Council under Grant GR/S63847/01, in part by the State Key Lab of Software Engineering at Wuhan University, and in part by the National Natural Science Foundation under Grant 60443003. This paper was recommended by Guest Editor Y. Jin.

The authors are with the Center of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, Birmingham, B15 2TT, U.K. (e-mail: j.he@cs.bham.ac.uk).

Digital Object Identifier 10.1109/TSMCC.2004.841903

Rather than designing more efficient new EAs for the node cover problem [7], [8], we focus on comparing the performance of three different EAs incorporating different amounts of knowledge and investigating the role of heuristic knowledge.

The rest of this paper is organized as follows: Section II introduces the node covering problem and describes three different EAs. Section III compares these three EAs both theoretically and experimentally. Section IV concludes the paper with a brief summary and future work.

II. Three Evolutionary Algorithms for Node Covering Problem

A. Node Covering Problem

An instance of the node covering problem [5], [6] can be described as follows: given an undirected graph $G = (V, E)$ where V is the set of nodes and E the set of edges, find a minimum subset C of V such that for any edge, at least one of its endpoints must be in the subset C.

In the node covering problem, a *feasible* solution is a subset C of V such that for any edge, one or both of its endpoints is in C. Such a subset is called a node cover. Other subsets of V are called *infeasible* solution.

Denote the node set $V = \{v_1, \ldots, v_n\}$ and the edge set $E = \{e_{i,j}; i, j = 1, \ldots, n\}$ where $e_{i,j} = 1$ if two nodes v_i, v_j are connected by an edge and $e_{i,j} = 0$ if they are not connected by any edge. Now we can represent a subset C by a binary string $(s_1 \ldots s_n)$ where $s_i = 1$ if the node v_i appears in the set C; $s_i = 0$ if the node v_i does not appear in C.

In our EAs, an individual is a string $(s_1 \ldots s_n)$ (i.e., a subset C of V). A population consists of a number of individuals. There are other advanced genetic representation to the node covering problem [8], which utilize more domain knowledge about the node cover problem.

An objective function for the node covering problem is introduced in [7]: given an individual $x = (s_1 \ldots s_n)$ (i.e., a subset C)

$$f(C) = \sum_{i=1}^{n} \left(s_i + n(1 - s_i) \sum_{j=1}^{n} (1 - s_j) e_{i,j} \right) \tag{1}$$

where the first part of the objective function is the number of nodes in the subset C, and the second part gives a penalty to the edges uncovered by the set C.

According to the above definition, the objective function value of a feasible solution is always smaller than that of an infeasible solution. The objective function $f(C)$ reaches its minimum if and only if C is the minimum node cover. The node covering problem is equivalent to search for a subset C of V such that $f(C)$ is minimized.

In order to use the proportional selection, we define a new fitness function as follows and transform the minimum node covering problem into a maximum optimization problem:

$$f'(C) = \sum_{i=1}^{n} n \sum_{j=1}^{n} e_{i,j} - f(C) + 1. \tag{2}$$

B. Three EAs

EA-I is a simple EA which uses little domain knowledge except the problem definition, that is

- the minimum node cover must satisfy

$$\min\{f(C); C \subset V\}. \quad (3)$$

A small difference of this simple EA from the EA in [7] is the crossover: here, we use a simple one-point crossover.

EA-I can be described as follows.

1) *Initialization*: set m individuals to be empty set (i.e., for $k = 1, \ldots, m$, set $x_k = (0 \cdots 0)$, where m is the population size).
2) *Crossover*: one-point crossover.
3) *Mutation*: for each individual $(s_1 \ldots s_n)$, flip each bit $s_i, i = 1, \ldots, n$ with probability $1/n$.
4) *Fitness evaluation*: compute the fitness function value for each individual.
5) *Selection*: always keep the best individual in the parent and offspring population. Select other $m-1$ individuals from offspring and parent populations according to the proportional selection strategy. These m individuals form the next-generation population.
6) If stop criterion is satisfied, stop; otherwise, return to step 2.
7) Output C with the minimum objective function value.

In addition to the definition of the node covering problem used in EA-I, EA-II incorporates an extra heuristic knowledge as indicated below.

- Since the node covering problem uses the minimum number of nodes to cover all edges, so a node with a larger degree is more likely to appear in the optimal node covering. However, this is only a heuristic, rather than accurate knowledge.

In EA-II, we assign a larger probability to the node with higher degree for its appearance in the node cover. This is a kind of greedy strategy used in a few deterministic greedy algorithm [5], [6]. EA-II is described as follows:

1) Compute the degree of each node $v_i (i = 1, \ldots, n)$ and denoted it by $\deg(i)$.
2) *Initialization*: set m individuals $x_k = (0 \cdots 0)$, for $k = 1, \ldots, m$.
3) *Crossover:* one-point crossover.
4) *Mutation:* for each individual $x_k, k = 1, \ldots, m$,
   a) Pick an edge $e_{i,j}$ at random.
   b) If $s_i = 0$ and $s_j = 0$ (i.e., neither nodes v_i nor v_j is in the set C_k), then generate a random number p in [0, 1], if p is less than

   $$\frac{\deg(i)}{\deg(i) + \deg(j)}$$

   then let $s_i = 1$ (i.e., add node v_i into the subset C_k); else let $s_j = 1$ (i.e., add node v_j into the subset C_k).
   c) Else if $s_i = 1$ and $s_j = 1$ (both nodes v_i and v_j are in C_k), then let $s_i = 0$ (i.e., remove node v_i from the subset C_k) with probability

   $$1 - \frac{\deg(i)}{\deg(i) + \deg(j)}$$

   and let $s_j = 0$ (i.e., remove node v_j from C_k) with probability

   $$1 - \frac{\deg(j)}{\deg(i) + \deg(j)}.$$

   d) Else if $s_i = 1$ and $s_j = 0$ (i.e., node v_i in C_k and v_j not in C_k), then let $s_i = 0$ (i.e., remove v_i from C_k) with probability

   $$1 - \frac{\deg(i)}{\deg(i) + \deg(j)}$$

   and let $s_j = 1$ (i.e., add v_j into C_k) with probability

   $$\frac{\deg(j)}{\deg(i) + \deg(j)}.$$

   e) Else if $s_i = 0$ and $s_j = 1$, do similar action as the above step.
5) *Fitness Evaluation*: Compute the fitness function.
6) *Selection*: Always keep the best individual in the parent and offspring population. Select $m - 1$ individuals from the offspring and parent populations according to the proportional selection strategy, and then these m individuals form the next generation population.
7) If stop criterion is satisfied, stop; otherwise return to step 3.
8) Output C with the minimum objective function value.

EA-III incorporates more knowledge than EA-II:

- For each edge, at least one endpoint of the edge should be in the minimum node cover. However, we do not know which endpoint is in the optimal node cover. Sometimes both are in it, sometimes only one of them. This knowledge is useful to produce a high quality 2-approximation solution [5], [6].

EA-III is decomposed into two phases.

1) In the first phase, we aim to produce a feasible solution using the heuristic knowledge. Given an edge, we know that at least one endpoint should belong to the optimal node cover. We choose an edge uncovered by the subset C (an edge whose endpoints are not in C) and then put both endpoints into C until all edges are covered.
2) In the second phase, we try to optimize the approximation node cover C by using EA-II.

EA-III is described as follows:

1) Compute the degree of each node $v_i (i = 1, \ldots, n)$ and denote it by $\deg(i)$.
2) *Initialization*: Set m individuals $x_k = (0 \cdots 0)$ for $k = 1, \ldots, m$.
   For individuals $x_k, k = 1, \ldots, m$:
   a) Choose an edge $e_{i,j}$ uncovered by C_k at random; let $s_i = 1$ and $s_j = 1$ (i.e., add both endpoints of $e_{i,j}$ into C_k.
   b) Repeat the above procedure until every edge is covered. Then produce m 2-approximation node covers.
3) *Crossover:* one-point crossover.
4) *Mutation:*
   a) Pick an edge $e_{i,j}$ at random.
   b) If $s_i = 0$ and $s_j = 0$ (i.e., neither nodes v_i nor v_j is in the set C_k), then generate a random number p in [0, 1], if p is less than

   $$\frac{\deg(i)}{\deg(i) + \deg(j)}$$

   then let $s_i = 1$ (i.e., add v_i into the subset C_k); else let $s_j = 1$ (i.e., add node v_j into the subset C_k).

c) Else if $s_i = 1$ and $s_j = 1$, then let $s_i = 0$ with probability

$$1 - \frac{\deg(i)}{\deg(i) + \deg(j)}$$

and $s_j = 0$ with probability

$$1 - \frac{\deg(j)}{\deg(i) + \deg(j)}.$$

d) Else, if $s_i = 1$ and $s_j = 0$, then let $s_i = 0$ with probability

$$1 - \frac{\deg(i)}{\deg(i) + \deg(j)}$$

and let $s_j = 1$ with probability

$$\frac{\deg(j)}{\deg(i) + \deg(j)}.$$

e) Else if $s_i = 0$ and $s_j = 1$, do similarly as the above step.

5) *Fitness Evaluation*: Compute the fitness function.
6) *Selection*: Always keep the best individual in the parent and offspring population. Select $m - 1$ individuals from offspring and parent populations according to the proportional selection, and these m individuals form the next-generation population.
7) If stop criterion is satisfied, stop; otherwise, return to step 3).
8) Output C with the minimum objective function value.

III. Comparison Study of Three EAs

We compare the above three evolutionary algorithms on the following topics.

1) Can the EAs find out the optimal solution in finite generations?
2) How many generations will the EAs take to find out the optimal solution?
3) How many generations will the EAs take to find out an approximation solution?

A. Theoretical Comparison

The first question is equivalent to the convergence of EAs. We prove that all these three EAs can find out the optimal node cover.

Denote τ to be the first hitting time for an EA to find the global optimal solution. We say the EA can find the optimal solution if the mean first hitting time $\mathbf{E}[\tau] < +\infty$.

In the following, we prove the convergence of EA-I by drift analysis [9], [12], [10].

Proposition 1: Let τ be the first hitting time for EA-I to find a minimum node cover. Then for EA-I, the mean first hitting time $\mathbf{E}[\tau] < +\infty$.

Proof: Let C^* be an optimal node cover. Define a distance between a population ξ and C^* as follows:

$$d(\xi, C^*) = \begin{cases} 0, & \text{if } C^* \in \xi \\ 1, & \text{if } C^* \notin \xi \end{cases} \tag{4}$$

which is written by $d(\xi)$ in short.

Assume at tth generation, the population is $\xi_t = \{C_1, \ldots, C_m\}$, where none of individuals are an optimal solution. Given an individual C in the population, it will become an optimal solution C^* if the following event happens during the mutation:

- all nodes in the sets $C \setminus C^*$ and $C^* \setminus C$ are flipped;
- all other nodes are kept unchanged.

Denote $k = |C \setminus C^*| + |C^* \setminus C|$, then the above happens with the following probability:

$$\mathbf{P}(C^* \in \xi_{t+1} \mid C \in \xi_t) = \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k}.$$

So the mean drift is

$$\Delta(\mathbf{E}) = \mathbf{E}[d(\xi_t) - d(\xi_{t+1})] \geq n^{-n}.$$

Since EA-I always keeps the best individual found during the evolution, the mean first hitting time of EA-I satisfies

$$\mathbf{E}[\tau] \leq d(\xi_0)/\Delta(\mathbf{E}) \leq n^n$$

which implies that EA-I can find an optimal solution in finite generations. ■.

The second evolutionary algorithm, EA-II, can find the global optimal solution too.

Proposition 2: Let τ be the first hitting time of EA-II to find a minimum node cover, then $\mathbf{E}[\tau] < +\infty$.

Proof: Let C^* be the optimal node cover. Define a distance between a population ξ and C^* as follows:

$$d(\xi, C^*) = \begin{cases} 0, & \text{if } C^* \in \xi \\ 1, & \text{if } C^* \notin \xi \end{cases}$$

which is written as $d(\xi)$ in short.

If we could prove that there is a positive $\rho > 0$ such that for any generation $t \geq 0$

$$\Delta(\mathbf{E}) = \mathbf{E}[d(\xi_t) - d(\xi_{t+n}) \mid d(\xi_t) > 0] \geq \rho \tag{5}$$

then we would have that

$$\mathbf{E}[\tau] \leq n d(\xi_0)/\Delta(\mathbf{E}) < +\infty.$$

In the following, we only need to prove that (5) holds.

At tth generation, assume the population does not include an optimal solution. Let C be the best individual in the population ξ_t. Assume that there are k_1 nodes in the set $C \setminus C^*$, k_2 nodes in the set $C^* \setminus C$, and k_3 nodes in the set $C \cap C^*$.

Now we estimate the probability of the following event of removing one node in $C \setminus C^*$, and not adding any node in the set $C \setminus C^*$.

For simplicity, we assume C is unchanged during the crossover (with probability $1 - p_c$), and only need to consider the effect of mutation.

During the mutation, the probability of removing one node u in $C \setminus C^*$ from C and not adding any node in $C \setminus C^*$ is always positive ($\geq p_m > 0$). In this way, we will generate a new individual C', which has removed one node in $C \setminus C^*$, and not added more nodes in $C \setminus C^*$ into it.

The new individual C' will survive in the next generation with a positive survive probability ($\geq p_s > 0$).

At most k_1 generations, it is possible for us to delete all of the nodes in $C \setminus C^*$. The probability of this event happening is

$$\geq ((1 - p_c) p_m p_s)^{k_1}.$$

Now we estimate the probability of the following event of adding one node in $C^* \setminus C$, and not adding one node in the set $C \setminus C^*$.

For simplicity, we assume C is unchanged during the crossover (with probability $1 - p_c$), and only need to consider the effect of mutation.

During the mutation, the probability of adding one node in $C^* \setminus C$ and not adding one node in $C \setminus C^*$ is always positive ($\geq p'_m > 0$). In this way, we will generate a new individual C', which has added one node in $C^* \setminus C$, and not added any node in $C \setminus C^*$ into it.

The new individual C' will survive in the next generation with a positive probability ($\geq p'_s > 0$).

At most k_2 generations, it is possible for us to add all of the nodes in $C^* \setminus C$ and not to add any node in $C \setminus C^*$. The probability of this event happening is

$$\geq ((1 - p'_c)p'_m p'_s)^{k_2}.$$

Since $0 < k_1 + k_2 \leq n$, let ρ be

$$\min\{((1 - p_c)p_m p_s)^{k_1}((1 - p_c)p'_m p'_s)^{k_2}; \quad 0 < k_1 + k_2 \leq n\}$$

then

$$\mathbf{E}[d(\xi_t) - d(\xi_{t+n})] \geq \rho.$$

We have shown that (5) holds and then $\mathbf{E}[\tau] < +\infty$. ■

Since EA-III is different from EA-II only on its initialization phase, from Proposition 2, we obtain that

Proposition 3: Let τ be the first hitting time of EA-III to find a minimum node cover. Then, $\mathbf{E}[\tau] < +\infty$.

The second question is intended to estimate the bound on the mean first hitting time for the EAs to find the optimal solution. Since the node covering problem is NP-hard, it is not expected that any of these three EAs can find the optimal node covering efficiently for all instances of the node covering problem, no matter how much heuristic knowledge is used. But in theory, it is still a challenging task to give such estimations on this time bound, and it needs further work at this point.

Next, we will offer some answers to the third question, and roughly estimate the mean first hitting time for the EAs to find a feasible solution.

EA-I can find a feasible solution quickly.

Proposition 4: Let τ' be the first hitting time when EA-I finds a feasible solution, then $\mathbf{E}[\tau'] = O(n|E|)$.

Proof: For an individual x (i.e., a subset C), denote $d(x)$ to be the number of edges in E whose endpoints are not in C. From (1), we see that $d(x_1) > d(x_2) > 0$ holds if and only if $f(x_1) > f(x_2) > n$ for any two given individuals x_1 and x_2.

Define a distance between the population ξ and the set of feasible solutions as

$$d(\xi) = \min\{d(x); x \in \xi\}. \tag{6}$$

At the beginning, $d(\xi_0) = |E|$. At tth generation, assume that population ξ_t does not include a feasible solution. Let C be the best individual in the population. If we flip one node of two endpoints in an uncovered edge and keep all other nodes unchanged, then a new edge is covered, the fitness is decreased.

The probability of this event is at least

$$\frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1}.$$

Denote the new individual to be C'. According to our elitist selection strategy, $f(\xi_{t+1}) \leq f(C')$ and $d(\xi_{t+1}) \leq d(C')$. The mean drift satisfies

$$\Delta(\mathbf{E}) = \mathbf{E}[d(\xi_t) - d(\xi_{t+1})] \geq 1 \times \frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1}. \tag{7}$$

So the mean first hitting time satisfies

$$\mathbf{E}[\tau'] \leq d(\xi_0)/\Delta(\mathbf{E}) = O(n|E|) \tag{8}$$

which means that at most $O(n|E|)$ generations are needed by EA-I to find a feasible solution. ■

Similarly, EA-II can find a feasible solution quickly too.

Proposition 5: Let τ' be the first hitting time of EA-II to find a feasible solution, then $\mathbf{E}[\tau'] = O(|E|^2)$.

Proof: For an individual x (i.e., a subset C), denote $d(x)$ to be the number of edges in V whose endpoints are not in C. Given two individuals x_1 and x_2, from (1), we know $d(x_1) > d(x_2) > 0$ holds if and only if $f(x_1) > f(x_2) > n$.

Define a distance between the population ξ and the set of feasible solutions to be

$$d(\xi) = \min\{d(x); x \in \xi\}.$$

After initialization, all individuals are empty sets and $d(\xi_0) = |E|$. At tth generation, assume the population still does not include any feasible solution. Let C be the best individual in the population. Thus, the probability of choosing an uncovered edge in C is not less than $|E|$ and one of this edge's endpoints will be added into C. The fitness of the offspring will be decreased and $d(\xi_{t+1})$ will be decreased by at least 1. Then the mean drift is

$$\Delta(\mathbf{E}) = \mathbf{E}[d(\xi_t) - d(\xi_{t+1})] \geq |E|^{-1}.$$

So we get that $\mathbf{E}[\tau] \leq d(\xi_0)/\Delta(\mathbf{E}) = O(|E|^2)$. ■

EA-III is the fastest among these three EAs to find a high-quality feasible solution. From a well-known fact from [5] and [6], we have the following.

Proposition 6: Let τ' be the first hitting time of EA-III to find a feasible solution. Then at the initialization phase, the EA could find a feasible solution (i.e., $\mathbf{E}[\tau'] = 0$). And for all instances of the node covering problem, the feasible solution is a 2-approximation solution to the optimal node cover, that is

$$\frac{f(C_{\tau'})}{f(C^*)} \leq 2.$$

From the above theoretical analysis, we predict the following.

1) All three EAs can find a feasible solution very quickly. EA-II and EA-III with heuristic knowledge may perform better than EA-I when searching a higher quality feasible solution. Among them, EA-III performed the best.
2) For all three EAs, it will take a very long time to find the optimal solution for some instances of the node covering problem.

B. Experimental Comparison

Experimental results give more details on the performance of these three EAs and confirm the above prediction.

TABLE I
TIME τ_1 FOR THE FIRST INSTANCE, WHERE THE FIGURES IN THE PARENTHESES ARE THE OBJECTIVE FUNCTION VALUE OF THE BEST INDIVIDUAL FOR EACH RUN

	1	2	3	4	5
EA-I	101 (64)	115 (62)	120 (66)	129 (63)	124 (65)
EA-II	119 (60)	121 (61)	120 (60)	107 (60)	126 (61)
EA-III	0 (84)	0 (88)	0 (84)	0 (86)	0 (84)
	6	7	8	9	10
EA-I	139 (63)	104 (63)	128 (61)	119 (61)	123 (62)
EA-II	118 (60)	141 (61)	117 (59)	131 (62)	129 (61)
EA-III	0 (88)	0 (86)	0 (92)	0 (82)	0 (86)

TABLE II
TIME τ_2 FOR THE FIRST INSTANCE, WHERE THE FIGURES IN THE PARENTHESES ARE THE OBJECTIVE FUNCTION VALUE OF THE BEST INDIVIDUAL FOR EACH RUN

	1	2	3	4	5
EA-I	10000 (54)	10000 (57)	10000 (59)	10000 (57)	10000 (59)
EA-II	10000 (59)	10000 (57)	10000 (57)	10000 (56)	10000 (55)
EA-III	10000 (56)	10000 (56)	10000 (56)	10000 (57)	10000 (58)
	6	7	8	9	10
EA-I	10000 (56)	10000 (57)	10000 (56)	10000 (56)	10000 (57)
EA-II	10000 (58)	10000 (58)	10000 (58)	10000 (58)	10000 (56)
EA-III	10000 (56)	10000 (57)	10000 (57)	10000 (59)	10000 (58)

Two different instances of the node covering problem are chosen in the experiment. The first instance is the following simple graph (let n be an odd):

$$V = \{v_1, \ldots, v_n\}$$
$$E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_2\}\}$$

where the minimum node cover is $\{v_2, v_4, \ldots, v_{n-1}\}$ or $(01010 \cdots 01\,010)$ and the value of its objective function is $(n-1)/2$.

This instance seems to be a hard problem because the EAs often fall into some local optima (e.g., $(10101 \cdots 10\,101)$), with the objective function value $(n+1)/2$, which is very close to the minimum value $(n-1)/2$.

We predict that EA-II and EA-III will not play more efficiently than EA-I because the degree of almost all nodes is 2. This provides little information to EA-II and EA-III because their search is based on the heuristic knowledge about node degree.

In the experiment, we choose $n = 101$, so the minimum node cover has 50 nodes inside.

The second example is the bipartite graph $B(L, R, E)$ given by Fig. 4.1 in [6], whose problem is hard for deterministic greedy search algorithm. We choose $|L| = 50$, so the minimum node cover has 50 nodes inside.

The population size in all of our experiments within the three EAs is 50. The stop criterion for these three EAs is that if the minimum node cover is found, then stop; otherwise, run up to 10 000 generations. We run each EA ten times independently and present two measures to the performance of the EAs.

- τ_1 is the first hitting time for the best individual in the population to find a 2-approximation solution.
- τ_2 is the first hitting time for the EA to find the minimum node cover.

We have restricted the total generation up to 10 000. If an EA can not find the optimal solution during this running time, we say the first hitting time is greater than 10 000.

Tables I and II show the experimental results of these three EAs on the first instance of node covering problem for ten runs.

Tables III and IV show the experimental results for the second instance.

From the figures given in the above tables, we observe the following.

1) All three EAs found a feasible solution very quickly. Among them, EA-III performed the best as we expected. EA-II and EA-III performed better than EA-I to find the feasible solutions. This confirms that EAs with knowledge perform better than the EAs without any domain knowledge in finding the approximation solutions.

TABLE III
TIME τ_1 FOR THE SECOND INSTANCE, WHERE THE FIGURES IN THE PARENTHESES ARE THE OBJECTIVE FUNCTION VALUE OF THE FOUND BEST INDIVIDUAL FOR EACH RUN

	1	2	3	4	5
EA-I	355 (99)	377 (100)	278 (100)	397 (99)	288 (100)
EA-II	127 (64)	158 (68)	154 (67)	160 (73)	188 (73)
EA-III	0 (100)	0 (100)	0 (100)	0 (100)	0 (100)
	6	7	8	9	10
EA-I	394 (100)	251 (100)	215 (100)	208 (92)	342 (100)
EA-II	192 (73)	164 (70)	164 (65)	145 (73)	185 (67)
EA-III	0 (100)	0 (100)	0 (100)	0 (100)	0 (100)

TABLE IV
TIME τ_2 FOR THE SECOND INSTANCE, WHERE THE FIGURES IN THE PARENTHESES ARE THE OBJECTIVE FUNCTION VALUE OF THE FOUND INDIVIDUAL FOR EACH RUN

	1	2	3	4	5
EA-I	762 (50)	10000 (57)	670 (50)	755 (50)	630 (50)
EA-II	231 (50)	263 (50)	281 (50)	267 (50)	408 (50)
EA-III	215 (50)	195 (50)	284 (50)	331 (50)	265 (50)
	6	7	8	9	10
EA-I	766 (50)	680 (50)	482 (50)	571 (50)	677 (50)
EA-II	352 (50)	304 (50)	428 (50)	264 (50)	291 (50)
EA-III	419 (50)	181 (50)	278 (50)	300 (50)	194 (50)

2) None of the EAs found the optimal solution efficiently for the first instance. Even EA-I, the simple evolutionary algorithm, is as good as EA-II and EA-III incorporating extra domain knowledge.

3) For the second instance, EA-II and EA-III found the optimal node cover in all ten runs, but EA-I failed in its second run. And the first hitting time is much longer than EA-II and EA-III. In this instance, domain heuristic knowledge seems to be helpful for the EAs to search the optimal node cover.

IV. CONCLUSION

In this paper, we have analyzed three EAs with varying amounts of domain knowledge incorporated for solving the node cover problem. The theoretical work has shown that all three EAs can find the optimal solution in finite generations, find a feasible solution efficiently, and EA-III with more heuristic knowledge can find some high-quality approximation solutions.

In the experiments, we observe that the heuristic knowledge is useful for the EAs to find the optimal solutions for the second instance, but makes little contribution for the first instance of the node covering problem. Even the performance of the simple EA is no worse than that of EA-II and EA-III on the first instance.

It should be pointed out that our experimental approach has some limitations in comparing different EAs. That is, it is impossible for us to test problems with very large sizes or to test all instances. We need more theoretical analysis in studying the performance of EAs [11]. Unfortunately, this is challenging work and is difficult to estimate the computation time bounds (the number of generations) of EAs.

Although our study has been carried out using the node covering problem as an example, the same research questions, analytical techniques, and results could be generalized to other problems.

References

[1] T. M. English, "Evaluation of evolutionary and genetic optimizer: No free lunch," in *Proc. 5th Annu. Conf. Evolutionary Programming*, L. J. Fogel, P. J. Angeline, and T. Bäck, Eds. Cambridge, MA, 1996, pp. 163–169.

[2] D. H. Wolpert and W. G. Macready, "No free lunch theorem for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[3] S. Droste, T. Jansen, and I. Wegener, "Perhaps not a free lunch but at least a free appetizer," in *Proc. Genetic Evolutionary Computation Conf.*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds. San Mateo, CA, 1999, pp. 833–839.

[4] C. C. Schumacher, M. D. Vose, and L. D. Whitley, "The no free lunch and problem description length," in *Proc. Genetic Evolutionary Computation Conf.*, L. Spector, E. D. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, Eds. San Mateo, CA, 2001, pp. 565–570.

[5] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. New York: Dover, 1998.

[6] R. Motwani, "Lecture Notes on Approximation Algorithms: Volume I," Dept. Comput. Sci., Stanford Univ., Stanford, CA, Tech. Rep. CS-TR-92-1435, 1992.

[7] S. Khuri and T. Bäck, "An evolutionary heuristic for the minimum vertex cover problem," in *Proc. Workshop 18th Annu. German Conf. Artificial Intelligence*, J. Hopf, Ed., Saarbrücken, Germany, 1994, pp. 86–90.

[8] I. K. Evans, "Evolutionary algorithms for vertex cover," in *Proc. 7th Annu. Conf. Evolutionary Programming*, V. W. Porto, N. Saravanan, D. E. Waagen, and A. E. Eiben, Eds. New York, 1998, pp. 377–386.

[9] J. He and X. Yao, "Drift analysis and average time complexity of evolutionary algorithms," *Artif. Intell.*, vol. 127, no. 1, pp. 57–85, 2001.

[10] ——, "Toward an analytic framework for analysing the computation time of evolutionary algorithms," *Artif. Intell.*, vol. 145, no. 1–2, pp. 59–97, 2003.

[11] H.-G. Beyer, H.-P. Schwefel, and I. Wegener, "How to analyze evolutionary algorithms," *Theor. Comput. Sci.*, vol. 287, no. 1, pp. 101–130, 2002.

[12] J. He and X. Yao, "Erratum to: Drift analysis and average time complexity of evolutionary algorithms," *Artif. Intell.*, vol. 140, no. 1–2, pp. 57–85, 2001.