A Novel Integer-Coded Memetic Algorithm for the Set *k*-Cover Problem in Wireless Sensor Networks

Chien-Chih Liao and Chuan-Kang Ting, *Senior Member, IEEE*

***Abstract*—The Set *k*-Cover problem aims to partition a set of nodes for the maximal number of covers. This problem is crucial for extending the lifetime of wireless sensor networks (WSNs) under the constraint of covering all targets. More specifically, the Set *k*-Cover problem enables partitioning the set of sensors into several covers over all targets and activating the covers by turns to effectively extend the WSN lifetime. To resolve this problem, we propose a novel memetic algorithm (MA) based on integer-coded genetic algorithm and local search. This paper adapts the crossover and mutation operators to integer representation and, furthermore, designs a new fitness function that considers both the number of covers and the contribution of each sensor to covers. A local improvement method, called the recycling operator, is developed to enhance the performance on the Set *k*-Cover problem. Experimental results show that the proposed MA significantly outperforms five evolutionary algorithms in terms of the number of covers obtained, hit rate (HR), and running time. In particular, the new MA increases 38.1% HR and saves 78.7% running time of state-of-the-art MA on average. The preferable results validate the effectiveness and efficiency of the proposed MA for the Set *k*-Cover problem.**

***Index Terms*—Evolutionary computation, genetic algorithm (GA), lifetime extension, local search, memetic algorithm (MA), wireless sensor networks (WSNs).**

I. Introduction

Wireless sensor networks (WSNs) have been widely applied to home security, surveillance, and environmental monitoring [3], [4], [21], [34]. The WSNs are composed of tiny, cheap, and wirelessly connected sensors that are ordinarily powered by batteries. In WSNs, sensors are deployed in a random or systematical manner to gather information within their sensing range. The sensors can transmit and even process the gathered information wirelessly. A crucial consideration to WSNs is the energy efficiency for long network lifetime. The various aspects of energy efficiency in WSNs include sensor scheduling [7], [31], [33], sensor deployment [6], [23], [35], data aggregation [15], [18], [19], [24], and routing protocol [16], [25], [28]. Akyildiz *et al.* [3] conducted a survey on the related problems and applications of WSNs.

Coverage is another key design factor in WSNs, which requires the sensors to cover all targets [6], [38]. However, due to the limited battery-powered energy, extending the WSN lifetime with full coverage is greatly needed [7], [9], [37]. An effective way to extend the WSN lifetime under the coverage constraint is to partition the set of sensors into several covers over all targets and then activate the covers one by one, where a *cover* is defined as a set of sensors that includes all targets [27], [29], [32]. By this way, the WSN lifetime can ideally be extended k times through the sequential activation of k covers under the coverage constraint. The Set k-Cover problem is formulated to optimize the schedule of sensor activities for maximal lifetime extension [31]. This problem aims to find the optimal partition of sensor set for maximal number of covers so as to maximize the WSN lifetime. The Set k-Cover problem has been proved to be NP-complete. Heuristic and metaheuristic algorithms are therefore proposed to deal with this problem. For example, the most constrained-minimally constraining covering (MCMCC) algorithm [31] explores the covers according to a heuristic based on the coverage of sensors, while the maximum covers using mixed integer programming (MCMIP) [7] conducts an implicit exhaustive search for covers. These algorithms, nevertheless, encounter the tradeoff between solution quality and efficiency: the approaches like MCMCC take only polynomial time but the results are unsatisfactory; on the other hand, the algorithms like MCMIP can achieve the optimal solution but are computationally very expensive. Some evolutionary algorithms (EAs) are further developed to address this issue [10], [17], [20], [33], [39].

Memetic algorithm (MA) hybridizes EA with local search operator to enhance the search ability of EA. This hybridization has shown to be very effective for superior solution quality and convergence speed [11]–[14]. In this paper, we propose an MA for the Set k-Cover problem. Specifically, the MA is based on integer-coded genetic algorithm (GA) and a local search operator. This paper devises mix crossover and mutation operators for the MA and presents a new fitness function that considers both the number of covers obtained and the contribution of sensors to formation of covers. The recycling operator is further developed to enhance the local search capability of the MA. This paper carries out experiments and performance comparison to examine the advantages of the MA in solution quality and convergence speed.

Manuscript received November 14, 2016; revised April 21, 2017; accepted July 8, 2017. This paper was recommended by Associate Editor Y. S. Ong.
(Corresponding author: Chuan-Kang Ting.)

The authors are with the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan (e-mail: lcch97p@cs.ccu.edu.tw; ckting@cs.ccu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2731598

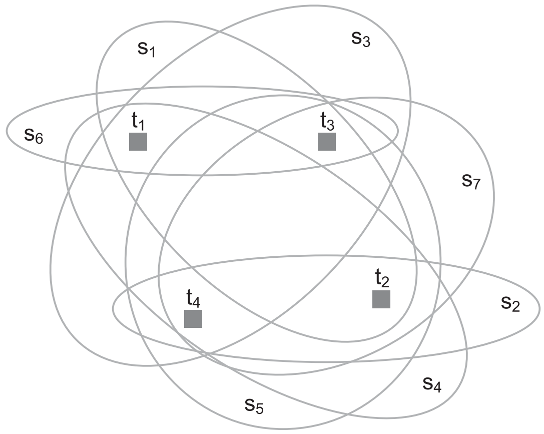


Fig. 1. Example WSN.

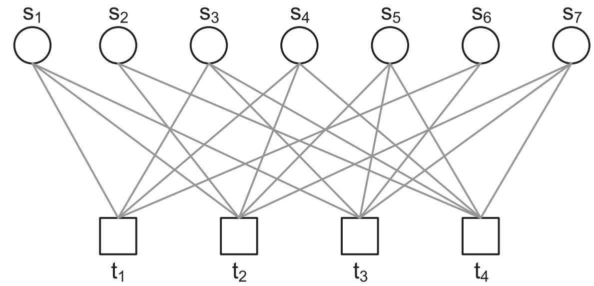


Fig. 2. Bipartite graph of a WSN.

The main contributions of this paper are summarized as follows.

1) We propose a novel integer-coded MA for the Set k-Cover problem. A tight upper bound is derived for the integer representation of sensor assignment. The crossover and mutation operators are designed to maintain the separation of critical sensors.
2) A new fitness function is devised considering the contribution of sensors in addition to the number of covers. The fitness evaluation can hence distinguish the utilization of sensors.
3) The recycling operator provides an effective local improvement by rearranging sensors to squeeze more covers without destroying the covers formed.
4) Comprehensive experiments are carried out to investigate the performance and advantages of the proposed MA in comparison with state-of-the-art methods. In addition, the experiments explore the effects of sensor range, the number of sensors, and the number of targets, upon the solution quality and running time.

The remainder of this paper is organized as follows. Section II gives the formulation of Set k-Cover problem and recapitulates its application to WSN lifetime extension. The proposed MA and its associated operators are described in Section III. Section IV presents the experimental results and performance comparison. Finally, the conclusions are drawn in Section V.

II. Set k-Cover Problem

Given n sensors $S = \{s_1, \ldots, s_n\}$ randomly distributed to monitor m targets $T = \{t_1, \ldots, t_m\}$, a target t_j is said to be *covered* if it locates in the sensing range of at least one sensor. Fig. 1 illustrates a WSN consisting of seven sensors and four targets. The relationship between sensors and targets can be represented by a bipartite graph $G = (V, E)$, where $V = S \cup T$ and $e_{ij} \in E$ if s_i covers t_j. For example, Fig. 2 demonstrates the bipartite graph of the WSN in Fig. 1, where $s_1 = \{t_1, t_2, t_3\}$, $s_2 = \{t_2, t_4\}$, and so on. In this example, the maximum number k of covers is three and the covers are $C_1 = \{s_1, s_2\}$, $C_2 = \{s_4, s_7\}$, and $C_3 = \{s_3, s_5, s_6\}$.

The Set k-Cover problem is to find the maximal number of covers from the sensors for WSN lifetime extension. That is, this problem aims to partition all sensors into several groups for the covers, where a *cover* is defined as a group that can sense all targets. The Set k-Cover problem has been proved to be NP-complete [7], [31]. The formal definition is given below.

Definition 1 (Set k-Cover Problem): Given a collection $S = \{s_1, \ldots, s_n\}$ of subsets of a finite set $T = \{t_1, \ldots, t_m\}$, find the maximum number, k, of covers $C_1, \ldots, C_k \subseteq S$ with $C_i \cap C_j = \emptyset$ for $i \neq j$, such that every element of T belongs to at least one element of C_i.

Slijepcevic and Potkonjak [31] formulated the Set k-Cover problem to extend WSN lifetime. They designed a polynomial-time heuristic method, called the MCMCC heuristic, for the Set k-Cover problem. However, the solutions obtained from MCMCC are often unsatisfactory. Cardei and Du [7] presented an equivalent problem, i.e., the disjoint set cover (DSC) problem, for WSN lifetime extension. They transformed the DSC problem into the maximum flow problem and solved it by the MCMIP. The MCMIP method guarantees the optima but requires exponential running time due to its exhaustive search, which is impractical for large-scale WSNs.

Furthermore, Cardei *et al.* [8] extended the DSC problem by allowing a sensor being assigned to multiple sensor covers. Berman *et al.* [5] considered the energy consumption rate and relaxed the constraint of disjoint covers in the scheduling of sensor activation. The problem is designated the sensor network life problem. This paper develops the $(1 + \ln(1 - q)^{-1})$-approximation algorithm guaranteeing that a q-portion of the monitored area is covered; for example, the lifetime obtained from the schedule must be at most 5.6 times shorter than the optimum lifetime for covering 99% of monitor area. Abrams *et al.* [1] considered maximization of network coverage rather than lifetime. They developed a randomized algorithm with an expected fraction $1 - 1/e$ to the optimum; moreover, they designed two greedy algorithms: one is a distributed $(1 + 1/2)$-approximation algorithm, and the other is a centralized $(1+(1-1/e))$-approximation algorithm. Ai *et al.* [2] modeled the problem of maximizing coverage under the consideration of lifetime guarantee. The authors proposed a distributed, robust, asynchronous algorithm to find the near-optimal coverage.

In the light of the success in resolving various optimization problems, some studies attempt applying EAs to deal with the Set k-Cover problem recently. Lai *et al.* [20] proposed a GA for maximum disjoint set covers (GAMDSC), which uses integer-coded representation to indicate the grouping of sensors. In addition, a scattering operator is performed on

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

each offspring to dispatch the critical sensors to different groups. The GAMDSC achieves significantly more covers than MCMCC does and requires less time than MCMIP. Liao and Ting [22] presented an order-based GA that conceptualizes partitioning the sensor set for covers as a process of collecting covers from sensors. Furthermore, Ting and Liao [33] designed an order-based MA for the Set k-Cover problem. The simulation results show that their MA outperforms MCMCC, integer-coded GA, order-coded GA, in the number of covers obtained, and is therefore viewed as a state-of-the-art EA for the Set k-Cover problem. Chen *et al.* [10] considered the dynamics of sensor activities and proposed a hybrid memetic framework to optimize disjoint set covers and dynamic coverage maintenance simultaneously. Zhang *et al.* [39] proposed a parallel GA using Kuhn–Munkres algorithm to address the Set k-Cover problem in large-scale WSNs.

The above studies show promising results on the Set k-Cover problem; however, there exist some issues at the approaches. Regarding the chromosome representation, integer-coded representation [20] provides a straightforward way to group sensors but cannot ensure that a group can form a cover. The required upper bound for the number of covers is another issue to address in the integer-coded representation. By contrast, the order-based representation proposed in [22] and [33] removes the need for the upper bound and can guarantee the completeness of covers. Nevertheless, order-based representation requires extra cost to examine every sensor to see if it achieves a cover. Aside from representation, the design and computational cost of fitness evaluation are essential to the performance. The hybrid memetic framework in [10] needs to repeat the MA several times to collect all covers, thereby aggravating the computational cost. The Kuhn–Munkres scheduled parallel genetic approach [39] applies the divide-and-conquer strategy to separate the target area into numerous subareas for the parallel GA to tackle, and then merges the feasible solutions from all subproblems. The evaluation on the number of covers obtained, however, requires several matching combinations with high time complexity.

This paper aims to design an effective and efficient MA to address the above issues and improve the performance on the Set k-Cover problem in order to extend WSN lifetime. More details about the problem formulation and the proposed algorithm are given below.

III. Proposed Memetic Algorithm

EAs have shown their effectiveness on various search and optimization problems. MA is an emerging dialect of EAs. To enhance the search efficiency, MA performs local improvement on the candidate solutions generated by EA in the evolution process [26], [30], [36]. Restated, MA combines EA and local search for performance enhancement.

This paper proposes a novel MA based on integer-coded GA and the recycling operator for the Set k-Cover problem. Following the framework of GA, the MA initializes the population by randomly generating chromosomes. Next, the selection operator chooses some chromosomes from the population to serve as parents for reproduction. The crossover operator then exchanges partial information of two parents to produce offspring. The MA determines a probability, named crossover rate, to perform the crossover operator. Similarly, the mutation operator is performed with a predetermined mutation rate to change some genes in the offspring. After crossover and mutation, the recycling operator is performed on the offspring to improve its quality. The reproduction process, to wit, selection, crossover, mutation, and recycling, repeats until the offspring population is filled. Acting on the Darwinian theory "survival of the fittest," the survivor operator subsequently chooses the fittest chromosomes to survive into the next generation. Algorithm 1 presents the framework of the proposed MA. The following sections elucidate the MA components in detail.

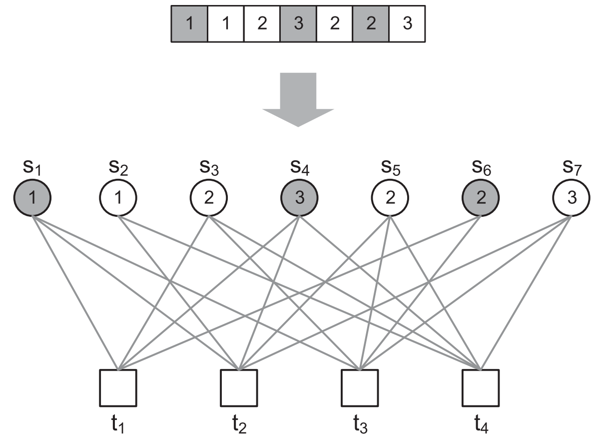


Fig. 3. Integer-coded chromosome. The number in the circle indicates the index of group. Gray circles mark the selected critical sensors.

Algorithm 1 Memetic Algorithm

```
Initialize population
Evaluate population
while (not terminated) do
    Select from population as parents
    Crossover the parents to generate offspring
    Mutate offspring
    Local improvement on offspring
    Evaluate offspring
    Survive from parents and offspring as population
end while
```

A. Representation

The candidate solutions in the MA are encoded as chromosomes to be evolved. This paper adopts integer representation for the grouping of sensors in the Set k-Cover problem. The gene value c at locus i indicates that sensor s_i is assigned to the group number G_c; a chromosome therefore designates the group composition. Fig. 3 illustrates an example chromosome with groups $G_1 = \{s_1, s_2\}$, $G_2 = \{s_3, s_5, s_6\}$, and $G_3 = \{s_4, s_7\}$.

The integer representation requires an upper bound ub for the valid range of the number of covers. A gene in a chromosome is therefore represented by $c \in \{1, 2, \ldots, ub\}$. A naive

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

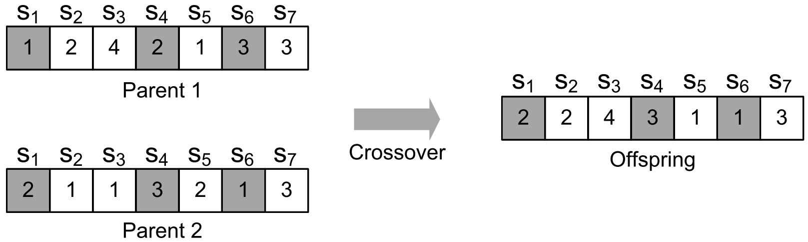


Fig. 4. Crossover operation. The critical sensors are marked in gray.

upper bound can be determined by the least covered targets, namely $\min_{t\in T}|S(t)|$, where $S(t)$ denotes the set of sensors covering target t. This upper bound, nonetheless, disregards the amount of sensors required to form a cover and thus results in overestimated number of covers. For example, the instance in Fig. 2 has $\min_{t\in T}|S(t)| = 4$ and yet can form only three covers.

To address this issue, we derive a tighter upper bound, which considers both the least covered targets and the minimal number of sensors needed. Let $T(s)$ denote the set of targets covered by sensor s. For calculation of the latter, we separate the sensor set S into two subsets S_f and S_p, where $S_f = \{s \in S | s \equiv T\}$ contains the sensors that can cover all targets and $S_p = S - S_f$. Dividing the total amount of targets $|T|$ by the maximum number of targets monitored by a sensor $s \in S_p$ assists in estimating the least number of sensors required for forming a cover, that is

$$|S_p^{\text{cover}}| = \left\lceil \frac{|T|}{\max_{s\in S_p}|T(s)|} \right\rceil. \tag{1}$$

Considering both the naive bound and the tighter bound presented in (1), the new upper bound is defined by

$$ub = \min\left\{ \min_{t\in T}|S(t)|, |S_f| + \left\lfloor \frac{|S_p|}{|S_p^{\text{cover}}|} \right\rfloor \right\}. \tag{2}$$

The example in Fig. 2 illustrates the utility of this new upper bound. According to (2), the upper bound is calculated by

$$ub = \min\left\{ 4, 0 + \left\lfloor \frac{7}{\left\lceil \frac{4}{3} \right\rceil} \right\rfloor \right\} = \min(4, \lfloor 3.5 \rfloor) = 3$$

which provides a bound tighter than $\min_{t\in T}|S(t)| = 4$, thereby reducing the search space.

The proposed MA randomly picks one of the least covered targets and regards its corresponding sensors as critical sensors. The optimal solution, in essence, must assign the sensors that monitor the same least covered target to distinct groups for more covers. Note that the number of critical sensors (i.e., $\min_{t\in T}|S(t)|$) is greater than or equal to the number of groups ub. Based on these observations, the MA randomly selects ub critical sensors and distributes them to the ub groups for each chromosome. To summarize, the proposed representation considers two characteristics of WSN lifetime extension through recycling redundant sensors and compulsorily separating critical sensors.

B. Variation Operators

The crossover operator exchanges partial information of the parents to produce offspring. Various crossover operators, such as 1-point, n-point, and uniform crossover, have been proposed for integer-coded representation. These operators, nevertheless, are unable to keep critical sensors separated. This paper designs a crossover operator to deal with this issue: an offspring inherits the group indexes of critical sensors from one parent and then fills the remaining loci with the genes of the other parent. By this way, the proposed mix crossover operator can recombine the parental information without destructing separation of critical sensors. Fig. 4 illustrates the crossover operation, where the offspring inherits the critical sensors from parent 2 and the other sensors from parent 1.

To introduce small variation, the mutation operator slightly alters the genes of chromosomes. An adequate mutation operator, as well as crossover operator, needs to change the group composition under the constraint of separating critical sensors. This paper presents the mix mutation that applies different mutation operators for the two types of sensors (genes). The mutation first changes the group indexes of either critical sensors or the noncritical ones with equal probability. Then the swap mutation and the random resetting mutation are conducted regarding critical and noncritical sensors, respectively. Restated, the swap mutation randomly exchanges two critical sensors, while the random resetting mutation reassigns the group index of noncritical sensors. Both mutation operators infuse diversity into group composition; in particular, the swap mutation caters to change the arrangement of critical sensors. Fig. 5 presents an example of these two mutation operators.

C. Recycling Operator

Considering the fact that some sensors are made redundant from the groups, moving these sensors to other groups is a promising task for gaining more covers. This paper develops the recycling operator to reallocate the redundant sensors to earn covers. The recycling operation is performed on each offspring after mutation.

The recycling operator includes two key steps. The first is to discover the redundant sensors from all groups. A redundant sensor is defined as a noncritical sensor that, once removed, will not shrink the coverage of its present group. The second key step is to randomly dispatch redundant sensors to the groups that have not formed covers yet. In this way, the recycling operator will not destroy a

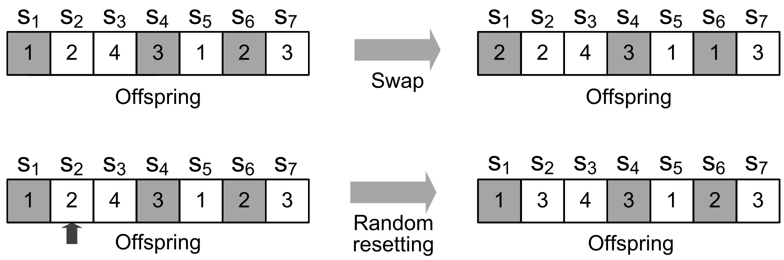


Fig. 5. Mutation operation including the swap mutation and the random resetting mutation. The critical sensors are marked in gray.

Algorithm 2 Recycling Operator

procedure RECYCLE
  for $i \in \{1, \ldots, ub\}$ **do**
    for $s \in G_i$ **do**
      if $T(G_i - \{s\}) = T(G_i)$ **then**
        $G_j \leftarrow \text{RandomPick}(U)$ $\triangleright$ U: collection of non-cover groups
        $G_j \leftarrow G_j + \{s\}$
        $G_i \leftarrow G_i - \{s\}$
      end if
    end for
  end for
end procedure

cover while extracting redundant sensors from it. In addition, reallocation of redundant sensors helps the groups to grow as covers. Algorithm 2 presents the procedure of recycling operator. Its time complexity is $\mathcal{O}(|S|)$ in that $\sum_{i=1}^{ub} |G_i| = |S|$.

D. Fitness Evaluation

Considering the objective of the Set k-Cover problem, an intuitive fitness evaluation is to count the number of covers formed by the group partition indicated in the chromosome. However, such fitness evaluation fails to distinguish the different arrangements of sensors that achieve an identical number of covers. Furthermore, it renders no information about the sensor utilization, which is closely associated with the room for improvement by the recycling operator. In this paper, we design a novel fitness function considering both the number of covers and the sensor utilization, where the latter is measured by the average contribution of all groups to the formation of covers. The fitness value of a chromosome $\boldsymbol{x}$ is defined by

$$f(\boldsymbol{x}) = k + \frac{1}{ub}\sum_{i=1}^{ub} c_i \tag{3}$$

where k denotes the number of covers formed by the groups determined in $\boldsymbol{x}$, ub stands for the upper bound determined by (2), and c_i represents the proportion of contribution calculated by

$$c_i = \frac{|T(G_i)|}{(1 + \sum_{s \in G_i} |T(s)|) + p} \tag{4}$$

in which $|T(G_i)|$ and $|T(s)|$ denote the numbers of targets covered by group G_i and sensor s, respectively; and penalty

$$p = \begin{cases} 0 & G_i \text{ forms a cover} \\ |T| & \text{otherwise.} \end{cases}$$

Note that the constant 1 in (4) ensures the decimal fraction $c_i \in [0, 1)$, where a larger c_i implies better sensor utilization. The proposed fitness evaluation can accordingly guide the evolutionary search toward maximization of the number of covers as well as sensor utilization.

IV. EXPERIMENTAL RESULTS

This paper conducts a series of experiments to evaluate the performance of the proposed MA (denoted by iMA) in comparison with state-of-the-art EAs, including integer-coded GA (iGA), GAMDSC [20], and order-based MA (oMA) [33], on the Set k-Cover problem for WSN lifetime extension. Additionally, to investigate the effects of the proposed fitness function and recycling operator, we examine three variants of integer-coded GA: iGA1, iGA2, and iGA3. The iGA1 uses the naive fitness function defined by the number of covers, whereas iGA2 and iGA3 adopt the proposed fitness function that additionally considers the average proportion of contribution (denoted by APoC). The iGA3 further uses the proposed mix crossover and mix mutation operators. Table I summarizes the parameter setting for the test EAs. Notably, the mutation rate for the oMA is defined by the parameter of Poisson-distributed random generator for the times of swapping the genes [33]. The mix mutation in iGA3 and iMA consists of swap and random resetting mutation. The mutation rate for swap mutation is 1.0 and that for random resetting is $1/l$, where l denotes the number of noncritical sensors. The termination criterion is set to 100 000 fitness evaluations for all test algorithms.

The test instances of the Set k-Cover problem are generated by simulation of WSNs, which randomly distributes sensors and targets over a 500×500 area. Each experimental setting includes 100 instances[1] and the test algorithms run once on each instance. The performance of test algorithms is evaluated according to three measures: the number of covers obtained, hit rate (HR), and running time. The number of covers obtained serves as the objective of the Set k-Cover problem and reflects the level of WSN lifetime extension. The experiments use the ub defined in Section III-A as the baseline since

[1] The test instances can be downloaded via http://cilab.cs.ccu.edu.tw/WSNdata.zip.

TABLE I
SETTING OF OPERATORS AND PARAMETERS FOR THE IGAS, GAMDSC, OMA, AND IMA

	iGA1	iGA2	iGA3	GAMDSC	oMA	iMA
Representation	Integer	Integer	Integer	Integer	Order	Integer
Fitness	#Covers	#Covers+APoC	#Covers+APoC	#Covers	Contribution	#Covers+APoC
Crossover	Uniform	Uniform	Mix	Uniform	Order	Mix
Crossover rate	1.0	1.0	1.0	1.0	1.0	1.0
Mutation	Random resetting	Random resetting	Mix	Random resetting	Swap	Mix
Mutation rate	$\frac{1}{\lvert S\rvert}$	$\frac{1}{\lvert S\rvert}$	1.0	$\frac{1}{\lvert S\rvert}$	1.0	1.0
Local search	–	–	–	Scattering	Compact	Recycling
Population size	100					
Selection	2-tournament					
Survival	$\mu + \lambda$					
Termination	1000 generations (100,000 evaluations)					

TABLE II
AVG AND HR OF COVERS OBTAINED FROM IGAS, GAMDSC, OMA, AND IMA FOR DIFFERENT SENSING RANGE r WITH $|S|$ SENSORS AND $|T|$ TARGETS. BOLDFACE INDICATES THE MOST COVERS OBTAINED AMONG THE COMPARING ALGORITHMS

$\lvert S\rvert$	$\lvert T\rvert$	r	ub	iGA1		iGA2		iGA3		GAMDSC		oMA		iMA	
				Avg	HR	Avg	HR	Avg	HR	Avg	HR	Avg	HR	Avg	HR
90	10	100	4.13	4.08	0.95	4.12	0.99	4.12	0.99	**4.13**	1.00	**4.13**	1.00	**4.13**	1.00
		150	9.84	9.58	0.81	**9.84**	1.00	**9.84**	1.00	9.83	0.99	**9.84**	1.00	**9.84**	1.00
		200	16.77	16.33	0.69	16.67	0.90	16.75	0.98	16.73	0.96	**16.77**	1.00	**16.77**	1.00
		250	25.18	24.67	0.63	25.02	0.85	**25.18**	1.00	25.16	0.98	**25.18**	1.00	**25.18**	1.00
		300	34.78	34.05	0.52	34.43	0.72	34.77	0.99	34.75	0.97	**34.77**	0.99	**34.77**	0.99
		350	45.34	44.49	0.48	44.77	0.57	45.32	0.98	45.28	0.94	45.31	0.97	**45.33**	0.99
		400	57.42	56.29	0.31	56.85	0.51	57.40	0.98	57.10	0.82	57.40	0.98	**57.41**	0.99
		450	69.39	68.26	0.22	68.96	0.61	**69.39**	1.00	69.08	0.83	**69.39**	1.00	**69.39**	1.00
		500	80.06	79.12	0.25	79.84	0.80	**80.06**	1.00	79.98	0.95	**80.06**	1.00	**80.06**	1.00
300	500	100	8.69	6.63	0.34	7.05	0.37	7.59	0.55	7.07	0.43	8.68	0.99	**8.69**	1.00
		150	20.20	16.70	0.10	17.01	0.09	18.42	0.26	17.96	0.30	20.19	0.99	**20.20**	1.00
		200	36.01	30.88	0.06	31.56	0.06	33.80	0.20	33.11	0.22	**36.01**	1.00	**36.01**	1.00
		250	58.16	50.47	0.00	51.32	0.00	53.82	0.02	53.19	0.07	58.13	0.97	**58.16**	1.00
		300	84.71	73.93	0.00	76.12	0.00	80.57	0.06	79.93	0.10	84.51	0.83	**84.71**	1.00
		350	115.57	101.92	0.00	110.74	0.00	114.54	0.50	111.26	0.10	114.92	0.68	**115.57**	1.00
		400	151.24	134.75	0.00	146.43	0.00	150.30	0.58	145.84	0.08	148.76	0.28	**151.08**	0.97
		450	179.26	166.20	0.00	176.62	0.02	178.12	0.20	155.90	0.01	178.87	0.64	**179.26**	1.00
		500	209.24	199.50	0.00	207.91	0.13	208.68	0.45	176.19	0.00	**209.24**	1.00	**209.24**	1.00

the maximal k is unknown. The HR represents the ratio of the number of runs achieving ub covers to that of all runs. The running time accounts for the average running time to the hits, which is measured on Intel Xeon E5-2620 machines. The following sections present and discuss the experimental results with different sensing range r, number of targets $|T|$, and number of sensors $|S|$.

A. Experiments With Different Sensing Ranges

The first experiment examines the performance of the proposed iMA with the five test EAs for different sensing ranges r on two problem scales: one with 90 sensors and 10 targets; the other with 300 sensors and 500 targets. Specifically, we look into the influences of iMA's components, including the fitness function using the average proportion of sensor contribution, mix crossover and mix mutation operators, and the recycling operator.

Table II compares the average number (Avg) of covers and HR obtained from the six EAs; furthermore, Table III presents the results of t-test with confidence level $\alpha = 0.05$. The tables show that iGA2 achieves significantly more covers and higher HR than iGA1 does on all test cases, validating the merit of the proposed fitness function for the Set k-Cover problem. The iGA3 surpasses iGA2 in both the number of covers obtained and HR on 16 out of 18 cases. These results show the advantages of mix crossover and mix mutation over uniform crossover and random resetting mutation. Meanwhile, iMA performs better or equally than iGA3 does on the small cases, and significantly betters on all the large cases. The superior performance of iMA demonstrates the effectiveness of the recycling operator on squeezing covers out of groups. Comparing the three MAs, iMA outperforms GAMDSC and oMA on 17 and 9 cases (14 and 5 cases with statistical significance), respectively.

Regarding the HR on the test instances of 90 sensors and 10 targets, iMA, iGA3, and oMA gain the highest HR, GAMDSC follows with small deficiency, and iGA1 has the lowest HR (see Fig. 6). As the problem scale increases to 300 sensors and 500 targets, the difference of test algorithms in HR becomes more significant. The iMA can maintain high HR; nonetheless, the others all result in a substantial decrease in HR on the large cases. These results show the robustness of iMA and confirm the utility of the proposed fitness function and recycling operator in performance improvement. In addition, although GAMDSC and iMA both apply the notion of scattering critical

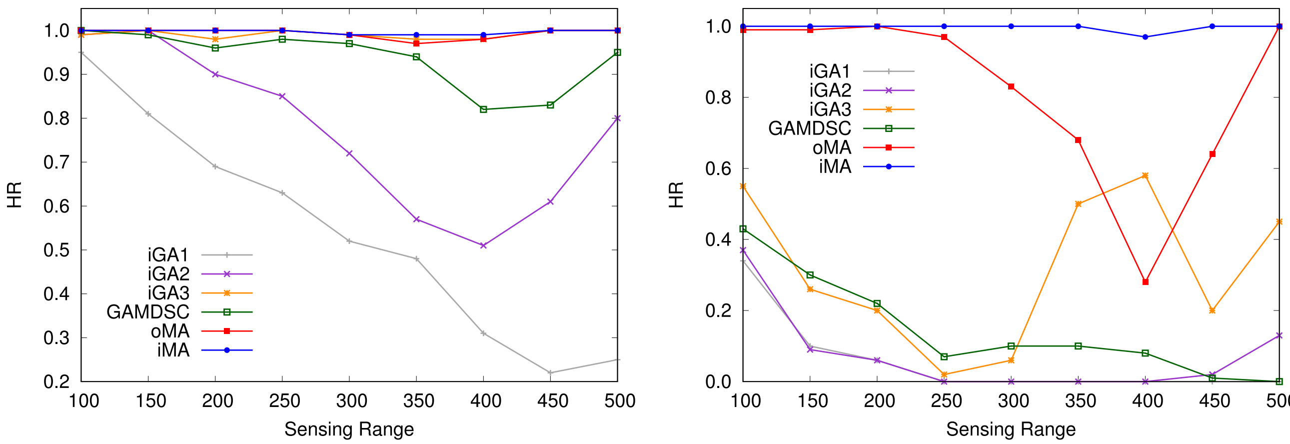


Fig. 6. HRs of iGAs, GAMDSC, oMA, and iMA for different sensing ranges with (left) $|S| = 90$, $|T| = 10$ and (right) $|S| = 300$, $|T| = 500$.

TABLE III

RESULTS OF ONE-TAILED PAIRED t-TEST ON THE NUMBERS OF COVERS OBTAINED FROM X AND Y ALGORITHMS (DENOTED BY X VERSUS Y) FOR DIFFERENT SENSING RANGES r WITH $|S|$ SENSORS AND $|T|$ TARGETS. SYMBOL "+" INDICATES THAT X IS SUPERIOR TO Y. BOLDFACE SIGNIFIES STATISTICAL SIGNIFICANCE WITH CONFIDENCE LEVEL $\alpha = 0.05$

$\|S\|$	$\|T\|$	r	iGA2 vs. iGA1	iGA3 vs. iGA2	oMA vs. GAMDSC	iMA vs. iGA3	iMA vs. GAMDSC	iMA vs. oMA
90	10	100	**2.25E-02 (+)**	–	–	1.60E-01 (+)	–	–
		150	**1.61E-05 (+)**	–	1.60E-01 (+)	–	1.60E-01 (+)	–
		200	**9.10E-07 (+)**	**5.35E-03 (+)**	**2.25E-02 (+)**	7.92E-02 (+)	**2.25E-02 (+)**	–
		250	**1.80E-07 (+)**	**5.07E-05 (+)**	7.92E-02 (+)	–	7.92E-02 (+)	–
		300	**6.20E-07 (+)**	**4.66E-08 (+)**	7.92E-02 (+)	–	7.92E-02 (+)	–
		350	**2.28E-03 (+)**	**1.17E-11 (+)**	9.05E-02 (+)	1.60E-01 (+)	**1.23E-02 (+)**	7.92E-02 (+)
		400	**2.35E-08 (+)**	**4.92E-13 (+)**	**9.21E-05 (+)**	1.60E-01 (+)	**6.06E-05 (+)**	1.60E-01 (+)
		450	**7.04E-17 (+)**	**1.37E-11 (+)**	**1.75E-04 (+)**	–	**1.75E-04 (+)**	–
		500	**5.00E-16 (+)**	**3.32E-06 (+)**	**1.59E-02 (+)**	–	**1.59E-02 (+)**	–
300	500	100	**1.21E-04 (+)**	**4.25E-06 (+)**	**9.26E-14 (+)**	**6.47E-10 (+)**	**1.27E-13 (+)**	1.60E-01 (+)
		150	**2.68E-02 (+)**	**1.42E-12 (+)**	**3.86E-15 (+)**	**4.47E-17 (+)**	**3.47E-15 (+)**	1.60E-01 (+)
		200	**3.84E-04 (+)**	**5.78E-20 (+)**	**7.97E-21 (+)**	**6.45E-22 (+)**	**7.97E-21 (+)**	–
		250	**1.91E-04 (+)**	**3.21E-19 (+)**	**5.83E-29 (+)**	**8.23E-33 (+)**	**7.58E-29 (+)**	**4.16E-02 (+)**
		300	**9.50E-14 (+)**	**1.18E-31 (+)**	**3.67E-27 (+)**	**4.02E-34 (+)**	**1.45E-26 (+)**	**2.49E-05 (+)**
		350	**1.23E-51 (+)**	**8.02E-44 (+)**	**8.43E-23 (+)**	**4.90E-13 (+)**	**1.64E-22 (+)**	**4.68E-07 (+)**
		400	**2.67E-60 (+)**	**3.55E-45 (+)**	**7.11E-16 (+)**	**1.13E-09 (+)**	**2.28E-21 (+)**	**1.87E-18 (+)**
		450	**1.79E-74 (+)**	**2.50E-14 (+)**	**4.13E-53 (+)**	**7.25E-26 (+)**	**2.02E-54 (+)**	**9.16E-11 (+)**
		500	**1.56E-67 (+)**	**4.09E-12 (+)**	**5.11E-80 (+)**	**1.01E-18 (+)**	**5.11E-80 (+)**	–

TABLE IV

RUNNING TIME OF IGAS, GAMDSC, OMA, AND IMA FOR DIFFERENT SENSING RANGE r WITH $|S|$ SENSORS AND $|T|$ TARGETS. DASH DENOTES THAT THE ALGORITHM FAILS TO ACHIEVE ub COVERS

r	$\|S\| = 90,\ \|T\| = 10$						$\|S\| = 300,\ \|T\| = 500$					
	iGA1	iGA2	iGA3	GAMDSC	oMA	iMA	iGA1	iGA2	iGA3	GAMDSC	oMA	iMA
100	0.030	0.011	0.005	0.011	0.002	0.002	0.989	0.414	0.816	0.656	0.208	0.060
150	0.110	0.042	0.016	0.052	0.003	0.004	5.984	2.047	2.687	4.301	0.765	0.132
200	0.245	0.088	0.035	0.086	0.007	0.006	11.063	7.802	6.086	7.902	1.992	0.219
250	0.339	0.128	0.054	0.124	0.014	0.010	–	–	8.057	16.241	7.734	0.398
300	0.472	0.187	0.074	0.129	0.029	0.016	–	–	16.983	16.460	19.815	1.267
350	0.599	0.268	0.104	0.169	0.039	0.024	–	–	23.718	28.833	35.238	2.863
400	0.921	0.314	0.183	0.234	0.051	0.036	–	–	24.074	24.250	50.179	7.078
450	0.893	0.482	0.137	0.262	0.036	0.035	–	29.062	29.930	18.876	51.294	9.898
500	1.154	0.581	0.082	0.150	0.013	0.025	–	28.051	26.205	–	14.012	13.134

sensors, the advanced design of iMA in the variation operators leads to its significant improvement in the number of covers obtained and HR.

Table IV lists the running time of the six algorithms for different sensing ranges. The table shows that iMA generally requires much less time than the other five algorithms. The results also reflect that the local search operators enhance the efficiency of iMA and oMA, in comparison with GAs. Comparing the three MAs, the proposed iMA shortens 70.8%–97.5% running time of GAMDSC and 71.2%–94.9% of oMA on the large cases. Fig. 7 shows the variation of the number of covers obtained from the six test EAs against

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

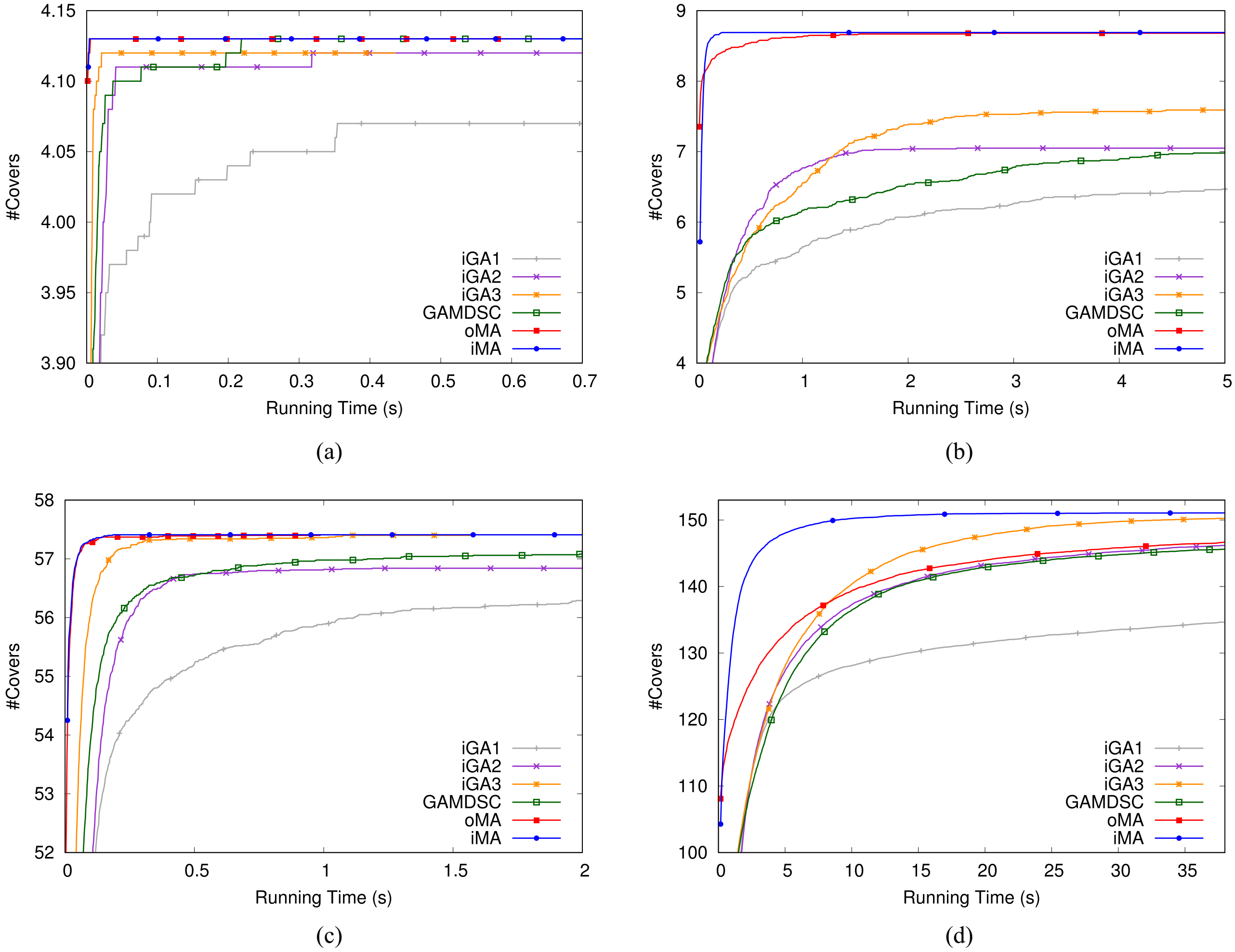


Fig. 7. Anytime behavior of iGAs, GAMDSC, oMA, and iMA for different sensing range $r \in \{100, 400\}$ with (left) $|S| = 90$, $|T| = 10$ and (right) $|S| = 300$, $|T| = 500$. (a) $r = 100$, $|S| = 90$, $|T| = 10$. (b) $r = 100$, $|S| = 300$, $|T| = 500$. (c) $r = 400$, $|S| = 90$, $|T| = 10$. (d) $r = 400$, $|S| = 300$, $|T| = 500$.

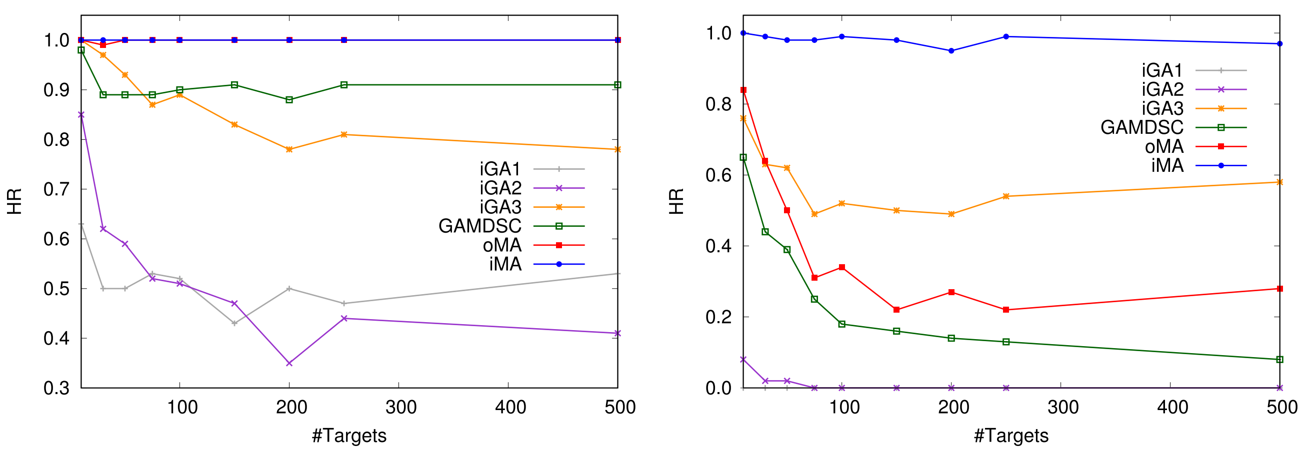


Fig. 8. HRs of iGAs, GAMDSC, oMA, and iMA for different numbers of targets with (left) $|S| = 90$, $r = 250$ and (right) $|S| = 300$, $r = 400$.

running time for different sensing ranges. The results indicate that iMA converges fastest, oMA follows, then iGA3, GAMDSC~iGA2, and lastly iGA1. The advantage of iMA in convergence speed is especially apparent on the large cases. These outcomes validate the effectiveness of the proposed representation, fitness function, and recycling operator on improving evolutionary optimization for the Set k-Cover problem.

B. Experiments With Different Numbers of Targets

The second experiment evaluates the performance of the test algorithms for different numbers of targets on two scales of

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE V
AVG AND HR OF COVERS OBTAINED FROM IGAs, GAMDSC, OMA, AND IMA FOR DIFFERENT NUMBERS OF TARGETS $|T|$ WITH $|S|$ SENSORS AND SENSING RANGE r. BOLDFACE INDICATES THE MOST COVERS OBTAINED AMONG THE COMPARING ALGORITHMS

$\|S\|$	r	$\|T\|$	ub	iGA1		iGA2		iGA3		GAMDSC		oMA		iMA	
				Avg	HR	Avg	HR	Avg	HR	Avg	HR	Avg	HR	Avg	HR
90	250	10	25.18	24.67	0.63	25.02	0.85	**25.18**	1.00	25.16	0.98	**25.18**	1.00	**25.18**	1.00
		30	21.13	20.34	0.50	20.70	0.62	21.10	0.97	20.99	0.89	21.12	0.99	**21.13**	1.00
		50	19.41	18.67	0.50	18.92	0.59	19.34	0.93	19.29	0.89	**19.41**	1.00	**19.41**	1.00
		75	18.58	17.84	0.53	17.95	0.52	18.43	0.87	18.44	0.89	**18.58**	1.00	**18.58**	1.00
		100	17.85	17.18	0.52	17.20	0.51	17.74	0.89	17.70	0.90	**17.85**	1.00	**17.85**	1.00
		150	17.26	16.44	0.43	16.50	0.47	17.07	0.83	17.16	0.91	**17.26**	1.00	**17.26**	1.00
		200	16.55	15.86	0.50	15.70	0.35	16.31	0.78	16.41	0.88	**16.55**	1.00	**16.55**	1.00
		250	16.20	15.52	0.47	15.40	0.44	15.97	0.81	16.10	0.91	**16.20**	1.00	**16.20**	1.00
		500	15.54	14.96	0.53	14.71	0.41	15.29	0.78	15.45	0.91	**15.54**	1.00	**15.54**	1.00
300	400	10	191.36	181.82	0.00	188.11	0.08	190.96	0.76	187.87	0.65	191.14	0.84	**191.36**	1.00
		30	175.13	163.35	0.00	171.32	0.02	174.52	0.63	168.57	0.44	174.56	0.64	**175.11**	0.99
		50	165.77	152.86	0.00	161.25	0.02	165.13	0.62	159.78	0.39	164.79	0.50	**165.73**	0.98
		75	165.09	150.45	0.00	160.69	0.00	164.11	0.49	157.59	0.25	163.28	0.31	**165.07**	0.98
		100	161.61	146.62	0.00	156.83	0.00	160.62	0.52	155.53	0.18	159.78	0.34	**161.59**	0.99
		150	159.13	143.20	0.00	153.85	0.00	158.04	0.50	152.68	0.16	156.42	0.22	**159.03**	0.98
		200	157.09	140.93	0.00	152.10	0.00	155.91	0.49	150.30	0.14	154.55	0.27	**157.00**	0.95
		250	155.56	139.26	0.00	150.62	0.00	154.66	0.54	149.66	0.13	152.89	0.22	**155.51**	0.99
		500	151.24	134.75	0.00	146.43	0.00	150.30	0.58	145.84	0.08	148.76	0.28	**151.08**	0.97

TABLE VI
RESULTS OF ONE-TAILED PAIRED t-TEST ON THE NUMBERS OF COVERS OBTAINED FROM X AND Y ALGORITHMS (DENOTED BY X VERSUS Y) FOR DIFFERENT NUMBERS OF TARGETS $|T|$ WITH $|S|$ SENSORS AND SENSING RANGE r. SYMBOLS + AND "–" INDICATE THAT X IS SUPERIOR AND INFERIOR TO Y, RESPECTIVELY. BOLDFACE SIGNIFIES STATISTICAL SIGNIFICANCE WITH CONFIDENCE LEVEL $\alpha = 0.05$

$\|S\|$	r	$\|T\|$	iGA2 vs. iGA1	iGA3 vs. iGA2	oMA vs. GAMDSC	iMA vs. iGA3	iMA vs. GAMDSC	iMA vs. oMA
90	250	10	**1.80E-07 (+)**	**5.07E-05 (+)**	7.92E-02 (+)	–	7.92E-02 (+)	–
		30	**1.92E-05 (+)**	**8.78E-10 (+)**	**6.60E-04 (+)**	**4.16E-02 (+)**	**7.18E-04 (+)**	1.60E-01 (+)
		50	**4.81E-03 (+)**	**2.37E-09 (+)**	**5.38E-04 (+)**	**3.75E-03 (+)**	**5.38E-04 (+)**	–
		75	1.03E-01 (+)	**3.55E-10 (+)**	**7.18E-04 (+)**	**2.13E-04 (+)**	**7.18E-04 (+)**	–
		100	3.99E-01 (+)	**1.17E-11 (+)**	**1.71E-03 (+)**	**3.52E-04 (+)**	**1.71E-03 (+)**	–
		150	2.24E-01 (+)	**1.16E-10 (+)**	**1.71E-03 (+)**	**4.44E-05 (+)**	**1.71E-03 (+)**	–
		200	3.31E-02 (–)	**4.72E-12 (+)**	**7.18E-04 (+)**	**9.53E-07 (+)**	**7.18E-04 (+)**	–
		250	5.14E-02 (–)	**1.93E-10 (+)**	**1.71E-03 (+)**	**1.67E-05 (+)**	**1.71E-03 (+)**	–
		500	8.01E-04 (–)	**1.07E-11 (+)**	**1.15E-03 (+)**	**2.71E-06 (+)**	**1.15E-03 (+)**	–
300	400	10	**4.43E-34 (+)**	**8.13E-25 (+)**	**1.13E-06 (+)**	**1.38E-06 (+)**	**2.37E-07 (+)**	**5.03E-05 (+)**
		30	**8.14E-45 (+)**	**1.34E-28 (+)**	**1.11E-09 (+)**	**1.26E-09 (+)**	**9.50E-11 (+)**	**7.61E-09 (+)**
		50	**1.85E-44 (+)**	**2.13E-31 (+)**	**8.20E-09 (+)**	**5.86E-09 (+)**	**1.06E-10 (+)**	**1.38E-12 (+)**
		75	**1.83E-55 (+)**	**3.62E-38 (+)**	**1.58E-10 (+)**	**3.16E-12 (+)**	**1.93E-14 (+)**	**4.01E-18 (+)**
		100	**1.42E-53 (+)**	**2.13E-40 (+)**	**2.48E-11 (+)**	**2.43E-12 (+)**	**4.20E-16 (+)**	**4.35E-17 (+)**
		150	**2.70E-58 (+)**	**4.14E-41 (+)**	**2.91E-14 (+)**	**1.43E-12 (+)**	**7.94E-20 (+)**	**1.94E-22 (+)**
		200	**5.16E-55 (+)**	**9.66E-46 (+)**	**2.66E-14 (+)**	**1.99E-13 (+)**	**1.82E-19 (+)**	**2.26E-20 (+)**
		250	**1.48E-58 (+)**	**4.11E-46 (+)**	**3.01E-15 (+)**	**5.08E-11 (+)**	**5.52E-22 (+)**	**7.34E-23 (+)**
		500	**2.67E-60 (+)**	**3.55E-45 (+)**	**7.11E-16 (+)**	**1.13E-09 (+)**	**2.28E-21 (+)**	**1.87E-18 (+)**

TABLE VII
RUNNING TIME OF IGAs, GAMDSC, OMA, AND IMA FOR DIFFERENT NUMBERS OF TARGETS $|T|$ WITH $|S|$ SENSORS AND SENSING RANGE r. DASH DENOTES THAT THE ALGORITHM FAILS TO ACHIEVE ub COVERS

$\|T\|$	$\|S\| = 90,\ r = 250$						$\|S\| = 300,\ r = 400$					
	iGA1	iGA2	iGA3	GAMDSC	oMA	iMA	iGA1	iGA2	iGA3	GAMDSC	oMA	iMA
10	0.339	0.128	0.054	0.124	0.014	0.010	–	4.165	1.608	1.680	0.887	0.358
30	0.520	0.203	0.102	0.246	0.038	0.018	–	5.897	2.527	2.750	2.363	0.726
50	0.537	0.230	0.101	0.224	0.039	0.017	–	6.131	3.543	3.329	4.374	0.896
75	0.683	0.215	0.120	0.283	0.063	0.022	–	–	4.895	5.522	7.597	1.812
100	0.804	0.230	0.216	0.320	0.078	0.021	–	–	5.918	5.986	9.860	1.934
150	0.754	0.287	0.242	0.484	0.120	0.027	–	–	8.432	6.901	12.680	3.039
200	1.101	0.476	0.335	0.419	0.109	0.032	–	–	10.384	9.684	19.683	4.003
250	1.239	0.473	0.388	0.672	0.113	0.030	–	–	12.746	13.339	20.511	4.187
500	2.370	0.701	0.612	0.977	0.177	0.051	–	–	24.074	24.250	50.179	7.078

problem instances: the small one with 90 sensors and sensing range $r = 250$; the large one with 300 sensors and sensing range $r = 400$. According to Table V, iGA2 obtains more covers than iGA1 does on 15 out of 18 cases; moreover, iGA3 betters iGA2 on all test cases. These results verify the efficacy of the proposed fitness evaluation as well as crossover

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

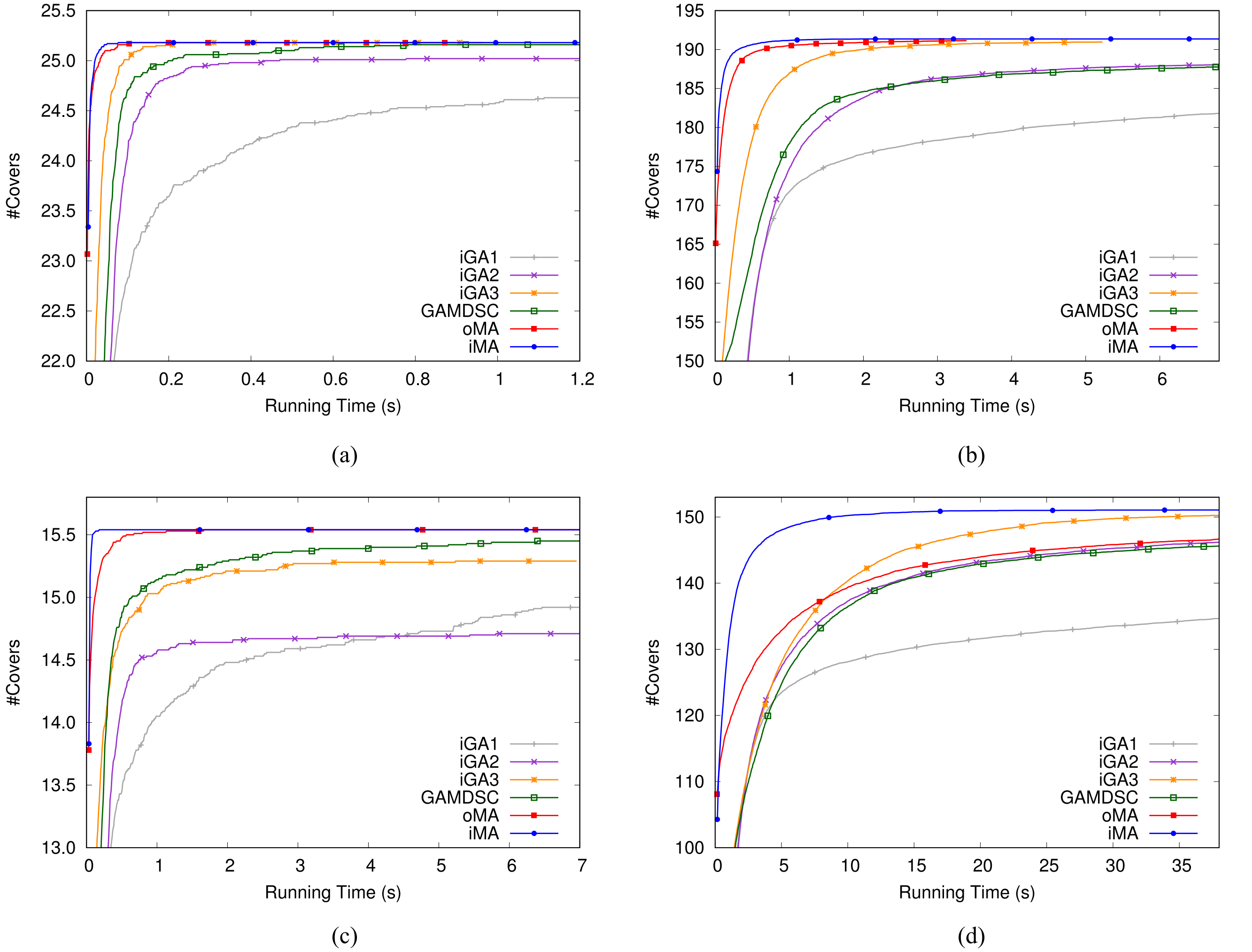


Fig. 9. Anytime behavior of iGAs, GAMDSC, oMA, and iMA for different numbers of targets $|T| \in \{10, 500\}$ with (left) $|S| = 90$, $r = 250$ and (right) $|S| = 300$, $r = 400$. (a) $|T| = 10$, $|S| = 90$, $r = 250$. (b) $|T| = 10$, $|S| = 300$, $r = 400$. (c) $|T| = 500$, $|S| = 90$, $r = 250$. (d) $|T| = 500$, $|S| = 300$, $r = 400$.

and mutation operators. Additionally, iMA obtains *ub* covers on all small instances; oMA achieves similar solution quality on the small instances but performs worse than iMA on the large instances. Table VI further indicates that iGA3 gains significantly more covers than iGA2 does in both problem scales. Notably, iMA performs significantly better than all test EAs on the large instances ($|S| = 300$, $r = 400$), validating that iMA benefits from the recycling operator in collection of covers.

Fig. 8 compares the HRs of the six test algorithms for different numbers of targets. The figure exhibits that iMA and oMA have ≥ 0.99 HRs for all test numbers of targets on the small instances. However, as the problem scale increases, the HRs of oMA and GAMDSC deteriorate seriously and are even lower than that of iGA3, which reveals their sensitivity to the problem scale. The proposed iMA, on the other hand, can retain high HRs and increases 16%–77% that of oMA on the large instances. This outcome shows the robustness of iMA.

Regarding the computational efficiency, Table VII and Fig. 9 demonstrate that iMA converges faster than the other five EAs in both problem scales, which validates the advantages of the proposed fitness function and recycling operator. The results of oMA reconfirm its sensitivity to the number of targets: although it requires less time on the small instances, oMA needs much more time than the integer-coded EAs (i.e., iGAs, GAMDSC, and iMA) and converges slower than iMA and iGA3 on the large instances. Precisely, iMA reduces 59.6%–85.9% running time of oMA. As stated in Section II, the higher complexity of order-based representation is a potential reason for the inefficiency of oMA in comparison to iMA. In addition, iGA3 converges faster than iGA2 does, while iGA2 surpasses iGA1. This outcome reflects the utility of the proposed variation operators and the contribution-based fitness evaluation in improving search efficiency for the Set k-Cover problem.

C. Experiments With Different Numbers of Sensors

Finally, we look into the performance of the test algorithms for different numbers of sensors. Likewise, the experiment includes two problem scales: 1) the small one with $|T| = 10$ targets and sensing range $r = 250$ and 2) the large one with $|T| = 500$ targets and sensing range $r = 400$. Table VIII

TABLE VIII
AVG AND HR OF COVERS OBTAINED FROM IGAS, GAMDSC, OMA, AND IMA FOR DIFFERENT NUMBERS OF TARGETS $|S|$ WITH $|T|$ TARGETS AND SENSING RANGE r. BOLDFACE INDICATES THE MOST COVERS OBTAINED AMONG THE COMPARING ALGORITHMS

$\|T\|$	r	$\|S\|$	ub	iGA1		iGA2		iGA3		GAMDSC		oMA		iMA	
				Avg	HR	Avg	HR	Avg	HR	Avg	HR	Avg	HR	Avg	HR
10	250	90	25.18	24.67	0.63	25.02	0.85	**25.18**	1.00	25.16	0.98	**25.18**	1.00	**25.18**	1.00
		100	28.76	28.07	0.56	28.55	0.82	**28.76**	1.00	28.68	0.94	28.75	0.99	**28.76**	1.00
		200	57.98	55.10	0.22	57.28	0.57	**57.98**	1.00	57.57	0.79	57.97	0.99	**57.98**	1.00
		300	88.92	83.04	0.05	87.78	0.44	88.73	0.89	87.63	0.67	88.86	0.97	**88.92**	1.00
500	400	90	43.78	41.59	0.13	42.82	0.28	43.72	0.96	43.51	0.79	43.61	0.86	**43.74**	0.97
		100	48.58	46.04	0.07	47.62	0.25	48.48	0.92	48.18	0.73	48.36	0.81	**48.53**	0.97
		200	100.87	91.45	0.00	98.10	0.00	100.53	0.77	98.12	0.24	99.79	0.52	**100.80**	0.94
		300	151.24	134.75	0.00	146.43	0.00	150.30	0.58	145.84	0.08	148.76	0.28	**151.08**	0.97

TABLE IX
RESULTS OF ONE-TAILED PAIRED t-TEST ON THE NUMBERS OF COVERS OBTAINED FROM X AND Y ALGORITHMS (DENOTED BY X VERSUS Y) FOR DIFFERENT NUMBERS OF SENSORS $|S|$ WITH $|T|$ TARGETS AND THE SENSING RANGE r. SYMBOLS + AND – INDICATE THAT X IS SUPERIOR AND INFERIOR TO Y, RESPECTIVELY. BOLDFACE SIGNIFIES STATISTICAL SIGNIFICANCE WITH CONFIDENCE LEVEL $\alpha = 0.05$

$\|T\|$	r	$\|S\|$	iGA2 vs. iGA1	iGA3 vs. iGA2	oMA vs. GAMDSC	iMA vs. iGA3	iMA vs. GAMDSC	iMA vs. oMA
10	250	90	**1.80E-07 (+)**	**5.07E-05 (+)**	7.92E-02 (+)	–	7.92E-02 (+)	–
		100	**2.27E-08 (+)**	**2.78E-05 (+)**	**9.43E-03 (+)**	–	**1.01E-02 (+)**	1.60E-01 (+)
		200	**3.60E-17 (+)**	**6.86E-12 (+)**	**1.70E-05 (+)**	–	**1.69E-05 (+)**	1.60E-01 (+)
		300	**3.74E-24 (+)**	**1.34E-10 (+)**	**4.07E-06 (+)**	**2.51E-03 (+)**	**5.84E-06 (+)**	5.46E-02 (+)
500	400	90	**1.97E-16 (+)**	**3.60E-20 (+)**	**2.03E-02 (+)**	7.92E-02 (+)	**8.80E-06 (+)**	**3.00E-04 (+)**
		100	**7.01E-23 (+)**	**1.60E-21 (+)**	**2.46E-03 (+)**	**1.23E-02 (+)**	**8.42E-07 (+)**	**6.70E-05 (+)**
		200	**5.99E-47 (+)**	**1.79E-42 (+)**	**1.33E-16 (+)**	**9.44E-06 (+)**	**3.31E-18 (+)**	**3.79E-13 (+)**
		300	**2.67E-60 (+)**	**3.55E-45 (+)**	**7.11E-16 (+)**	**1.13E-09 (+)**	**2.28E-21 (+)**	**1.87E-18 (+)**

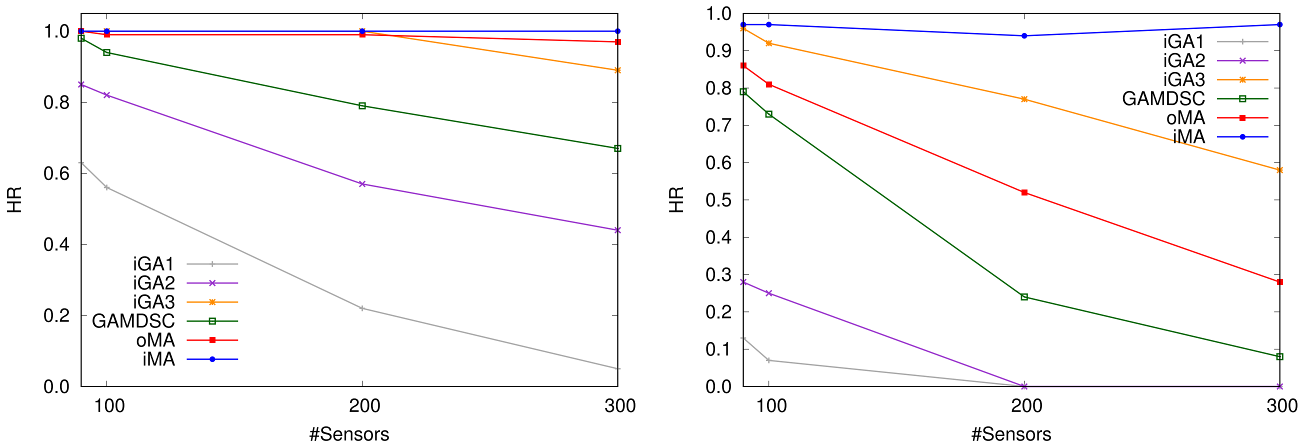


Fig. 10. HRs of iGAs, GAMDSC, oMA, and iMA for different numbers of sensors with (left) $|T| = 10$, $r = 250$ and (right) $|T| = 500$, $r = 400$.

TABLE X
RUNNING TIME OF IGAS, GAMDSC, OMA, AND IMA FOR DIFFERENT NUMBERS OF SENSORS $|S|$ WITH $|T|$ TARGETS AND SENSING RANGE r. DASH DENOTES THAT THE ALGORITHM FAILS TO ACHIEVE ub COVERS

$\|S\|$	$\|T\| = 10$, $r = 250$						$\|T\| = 500$, $r = 400$					
	iGA1	iGA2	iGA3	GAMDSC	oMA	iMA	iGA1	iGA2	iGA3	GAMDSC	oMA	iMA
90	0.339	0.128	0.054	0.124	0.014	0.010	4.391	2.459	1.768	2.475	3.737	0.497
100	0.397	0.169	0.068	0.210	0.021	0.015	6.164	3.431	2.244	2.962	5.333	0.622
200	1.260	0.717	0.330	0.557	0.119	0.045	–	–	12.228	13.524	29.423	3.725
300	2.985	2.091	0.796	1.329	0.252	0.088	–	–	24.074	24.250	50.179	7.078

lists the average number of covers obtained and HRs of the comparing algorithms, and Table IX examines their statistical significance. Moreover, Fig. 10 plots the HRs of the six algorithms for different numbers of sensors. In general, the number of covers obtained increases while the HR decreases with the growing number of sensors. The superior performance of iGA2 to iGA1 verifies the effectiveness of average proportion of sensor contribution in enhancing fitness evaluation. The iGA3

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

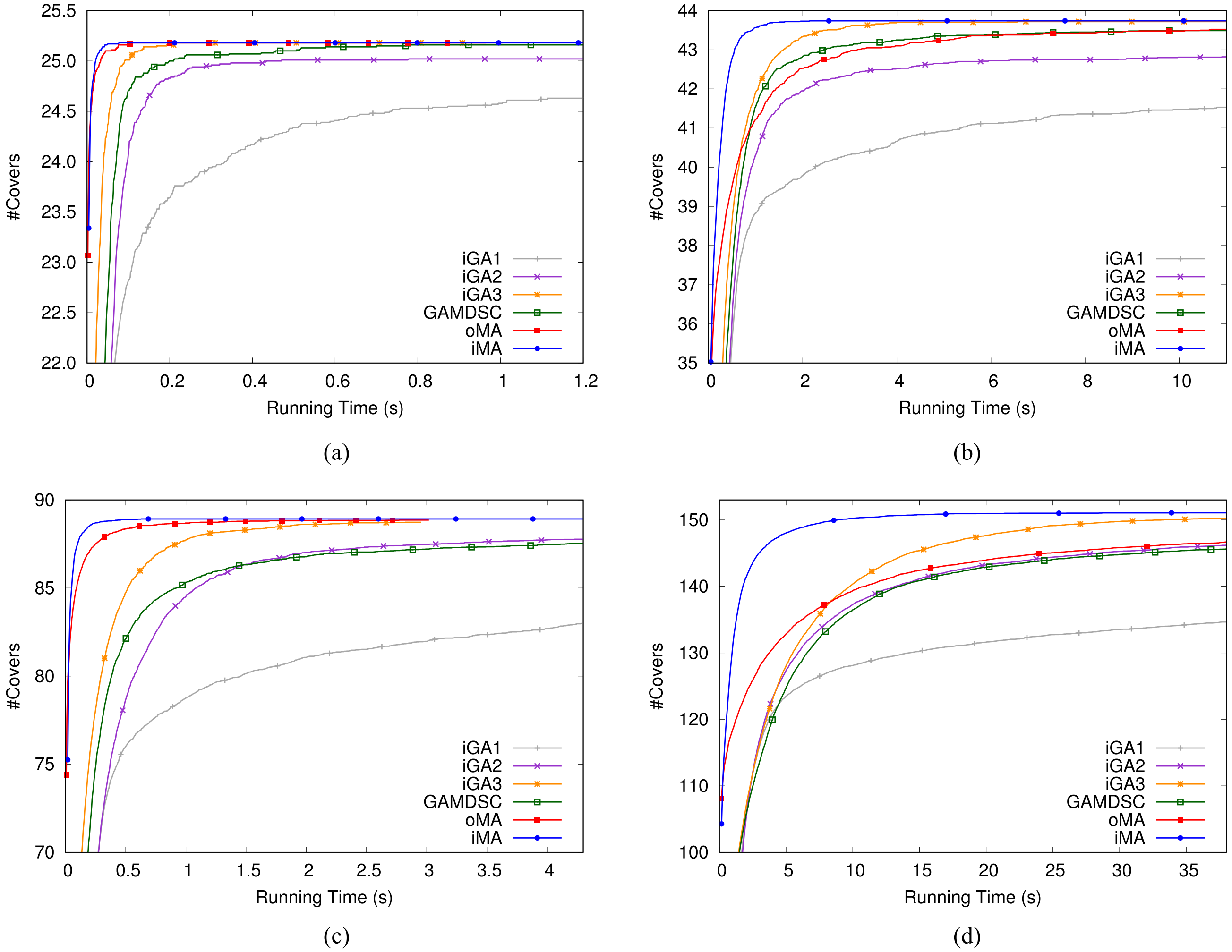


Fig. 11. Anytime behavior of iGAs, GAMDSC, oMA, and iMA for different numbers of sensors $|S| \in \{90, 300\}$ with (left) $|T| = 10$, $r = 250$ and (right) $|T| = 500$, $r = 400$. (a) $|S| = 90$, $|T| = 10$, $r = 250$. (b) $|S| = 90$, $|T| = 500$, $r = 400$. (c) $|S| = 300$, $|T| = 10$, $r = 250$. (d) $|S| = 300$, $|T| = 500$, $r = 400$.

achieves more covers and higher HR than iGA2. These results show the advantages of the proposed mix crossover and mix mutation operators. In addition, iGA3 outperforms GAMDSC on all test cases and does oMA on 6 out of 8 cases. The proposed iMA further betters iGA3, validating the benefit of the recycling operator. The oMA performs comparably to iGA3 and iMA on the small instances; nevertheless, its performance worsens drastically with the increase in the number of sensors. This deterioration reveals the vulnerability of oMA to the number of sensors. The iMA, by contrast, can maintain high HR (11%–69% higher than oMA) and large number of covers against the growing number of sensors on the large instances. These results indicate the advantages of the proposed iMA for the Set *k*-Cover problem in WSN lifetime extension.

Table X presents the running time of the test algorithms for different numbers of sensors. Increasing the number of sensors generally results in longer running time. Fig. 11 shows the anytime behavior of the six algorithms for different numbers of sensors. On the small cases, iMA and oMA obtain faster convergence than the other test algorithms. As the problem scale increases, iMA still performs best, whereas oMA deteriorates and converges slower than iGA3 and even GAMDSC. The results reflect the sensitivity of oMA to the number of sensors: its required running time increases rapidly with the number of sensors on the large instances. According to the comparative results, the iMA holds the highest efficiency among the six test algorithms in both problem scales; notably, iMA saves 28.6%–65.1% running time of oMA on small test cases and 85.9%–88.7% on large test cases. Comparing the three iGA variants, iGA3 requires only 50.6% running time of iGA2, while iGA2 needs merely 53.2% running time of iGA1 on average. Fig. 11 shows the same preference in convergence speed: iGA3, iGA2, and then iGA1. The convergence speed of GAMDSC lies between that of iGA2 and iGA3. These outcomes reconfirm the merits of the proposed fitness function, variation operators, and recycling operator in algorithmic efficiency for the Set *k*-Cover problem.

V. Conclusion

The Set *k*-Cover problem renders an important formulation for extending the WSN lifetime with a full coverage of targets. In this paper, we design an effective integer-coded MA for the Set k-Cover problem. The proposed MA uses integer representation to indicate the group number assigned to a sensor; a chromosome therefore represents the arrangement of all sensors for formation of covers. This paper presents a tighter upper bound for the integer representation to reduce the search space. The mix crossover and mix mutation operators are further adapted to the chromosome representation. Moreover, we design a new fitness evaluation that considers both the number of covers and the contribution of each sensor to covers. A local search method, called the recycling operator, is developed to enhance the performance on the Set k-Cover problem.

This paper conducts a series of experiments with different settings for the sensing range, number of targets, and number of sensors, to evaluate the proposed MA. The experimental results show that the integer-coded MA can outperform state-of-the-art EAs, including GAMDSC and order-based MA, in terms of HR, the number of covers obtained, and running time. The proposed iMA, on average, increases 38.1% HR and reduces 78.7% running time of oMA on the 22 large test instances. These satisfactory results show the advantages of the proposed fitness function and the recycling operator in performance improvement. They also validate the effectiveness and efficiency of the proposed MA on the Set k-Cover problem in WSNs.

Future work may further consider different aspects of EAs and the Set k-Cover problem. The problem formulation can be extended to dynamic network environments. Design of effective EAs for the dynamic Set k-Cover problem provides an important research direction and application to WSN lifetime extension.

Acknowledgment

The authors would like to thank the editor and reviewers for their valuable comments and suggestions.

References

[1] Z. Abrams, A. Goel, and S. Plotkin, “Set K-cover algorithms for energy efficient monitoring in wireless sensor networks,” in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, Berkeley, CA, USA, 2004, pp. 424–432.

[2] X. Ai, V. Srinivasan, and C.-K. Tham, “DRACo: Distributed, robust an asynchronous coverage in wireless sensor networks,” in *Proc. 4th Annu. IEEE Commun. Soc. Conf. Sensor Mesh Ad Hoc Commun. Netw.*, San Diego, CA, USA, 2007, pp. 530–539.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[4] B. Badrinath, J. Scholtz, M. Srivastava, K. Mills, and V. Stanford, *Special Issue on Smart Spaces and Environments* (IEEE Personal Communications). New York, NY, USA: IEEE Commun. Soc., 2000.

[5] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, “Power efficient monitoring management in sensor networks,” in *Proc. Wireless Commun. Netw. Conf.*, vol. 4. Atlanta, GA, USA, 2004, pp. 2329–2334.

[6] A. L. Buczak, Y. Jin, H. Darabi, and M. Jafari, “Genetic algorithm based sensor network optimization for target tracking,” *Intell. Eng. Syst. Through Artif. Neural Netw.*, vol. 9, pp. 349–354, Dec. 1999.

[7] M. Cardei and D.-Z. Du, “Improving wireless sensor network lifetime through power aware organization,” *Wireless Netw.*, vol. 11, no. 3, pp. 333–340, 2005.

[8] M. Cardei, M. T. Thai, Y. Li, and W. Wu, “Energy-efficient target coverage in wireless sensor networks,” in *Proc. IEEE INFOCOM*, Miami, FL, USA, 2005, pp. 1976–1984.

[9] M. Cardei and J. Wu, “Energy-efficient coverage problems in wireless ad-hoc sensor networks,” *Comput. Commun.*, vol. 29, no. 4, pp. 413–420, 2006.

[10] C.-P. Chen *et al.*, “A hybrid memetic framework for coverage optimization in wireless sensor networks,” *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2309–2322, Oct. 2015.

[11] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, “A multi-facet survey on memetic computation,” *IEEE Trans. Evol. Comput.*, vol. 15, no. 5, pp. 591–607, Oct. 2011.

[12] L. Feng, Y.-S. Ong, M.-H. Lim, and I. W. Tsang, “Memetic search with interdomain learning: A realization between CVRP and CARP,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 644–658, Oct. 2015.

[13] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, “Multiobjective multifactorial optimization in evolutionary multitasking,” *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1652–1665, Jul. 2017.

[14] W. E. Hart, N. Krasnogor, and J. E. Smith, Eds., *Recent Advances in Memetic Algorithms*. Heidelberg, Germany: Springer, 2004.

[15] Z. He, B. S. Lee, and X. S. Wang, “Aggregation in sensor networks with a user-provided quality of service goal,” *Inf. Sci.*, vol. 178, no. 9, pp. 2128–2149, 2008.

[16] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, 2000, pp. 1–10.

[17] X.-M. Hu *et al.*, “Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks,” *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 766–781, Oct. 2010.

[18] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, Boston, MA, USA, 2000, pp. 56–67.

[19] L. Krishnamachari, D. Estrin, and S. Wicker, “The impact of data aggregation in wireless sensor networks,” in *Proc. 22nd Int. Conf. Distrib. Comput. Syst. Workshop*, Vienna, Austria, 2002, pp. 575–578.

[20] C.-C. Lai, C.-K. Ting, and R.-S. Ko, “An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications,” in *Proc. Congr. Evol. Comput.*, Singapore, 2007, pp. 3531–3538.

[21] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, “Detection, classification, and tracking of targets,” *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 17–29, Mar. 2002.

[22] C.-C. Liao and C.-K. Ting, “Extending wireless sensor network lifetime through order-based genetic algorithm,” in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Singapore, 2008, pp. 1434–1439.

[23] X. Liu, “A deployment strategy for multiple types of requirements in wireless sensor networks,” *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2364–2376, Oct. 2015.

[24] F. Marcelloni and M. Vecchio, “Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization,” *Inf. Sci.*, vol. 180, no. 10, pp. 1924–1941, 2010.

[25] C. Ok, S. Lee, P. Mitra, and S. Kumara, “Distributed routing in wireless sensor networks using energy welfare metric,” *Inf. Sci.*, vol. 180, no. 9, pp. 1656–1670, 2010.

[26] Y.-S. Ong, M. H. Lim, and X. Chen, “Memetic computation—Past, present & future,” *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 24–31, May 2010.

[27] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, “Energy-aware wireless microsensor networks,” *IEEE Signal Process.*, vol. 19, no. 2, pp. 40–50, Mar. 2002.

[28] P. S. Sausen, M. A. Spohn, and A. Perkusich, “Broadcast routing in wireless sensor networks with dynamic power management and multi-coverage backbones,” *Inf. Sci.*, vol. 180, no. 5, pp. 653–663, 2010.

[29] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Optimizing sensor networks in the energy-latency-density design space,” *IEEE Trans. Mobile Comput.*, vol. 90, no. 1, pp. 70–80, Jan./Mar. 2002.

[30] V. A. Shim, K. C. Tan, and H. Tang, “Adaptive memetic computing for evolutionary multiobjective optimization,” *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 610–621, Apr. 2015.

[31] S. Slijepcevic and M. Potkonjak, “Power efficient organization of wireless sensor networks,” in *Proc. IEEE Int. Conf. Commun.*, vol. 2. Helsinki, Finland, 2001, pp. 472–476.

[32] J. A. Stine and G. D. Veciana, “Improving energy efficiency of centrally controlled wireless data networks,” *Wireless Netw.*, vol. 8, no. 6, pp. 681–700, 2002.

[33] C.-K. Ting and C.-C. Liao, “A memetic algorithm for extending wireless sensor network lifetime,” *Inf. Sci.*, vol. 180, no. 24, pp. 4818–4833, 2010.

[34] P. K. Varshney, *Distributed Detection and Data Fusion*. New York, NY, USA: Springer-Verlag, 1996.

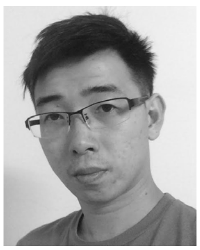
[35] Y. Yoon and Y.-H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1473–1483, Oct. 2013.

[36] G. Zhang and Y. Li, "A memetic algorithm for global optimization of multimodal nonseparable problems," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1375–1387, Jun. 2016.

[37] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Int. J. Ad Hoc Sensor Wireless Netw.*, vol. 1, nos. 1–2, pp. 89–124, 2005.

[38] X. Zhang, Y. Zhou, Q. Zhang, V. C. S. Lee, and M. Li, "Problem specific MOEA/D for barrier coverage with wireless sensors," *IEEE Trans. Cybern.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/document/7515224/

[39] X.-Y. Zhang *et al.*, "Kuhn–Munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 695–710, Oct. 2016.



Chien-Chih Liao received the Ph.D. degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan, in 2016.

His current research interests include evolutionary computation, memetic algorithm, computational intelligence, and their applications in wireless sensor networks.

Chuan-Kang Ting (S'01–M'06–SM'13) received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, in 1994, the M.S. degree from National Tsing Hua University, Hsinchu, Taiwan, in 1996, and the Dr. rer. nat. degree from the University of Paderborn, Paderborn, Germany, in 2005.

He is currently a Professor with the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan. His current research interests include evolutionary computation, computational intelligence, machine learning, and their applications in music, art, networks, and bioinformatics.

Dr. Ting is an Associate Editor of the *IEEE Computational Intelligence Magazine* and the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, and an Editorial Board Member of *Soft Computing* and *Memetic Computing* journals. He serves as the IEEE Computational Intelligence Society (CIS) Newsletter Editor, the Vice Chair of Intelligent Systems Applications Technical Committee, the Chair of Creative Intelligence Task Force, and the Vice Chair of Intelligent Network Systems Task Force, all in IEEE CIS. He is an Executive Board Member of Taiwanese Association for Artificial Intelligence.