Knowledge-Based Systems 239 (2022) 108000

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# Combining max–min ant system with effective local search for solving the maximum set k-covering problem

Yupeng Zhou [a,b], Xiaofan Liu [a], Shuli Hu [a,b,*], Yiyuan Wang [a,b,*], Minghao Yin [a,b,*]

[a] College of Information Science and Technology, Northeast Normal University, Changchun, China
[b] Key Laboratory of Applied Statistics of MOE, Northeast Normal University, Changchun, China

ARTICLE INFO

*Article history:*
Received 10 June 2021
Received in revised form 14 December 2021
Accepted 17 December 2021
Available online 24 December 2021

*Keywords:*
Maximum set k-covering problem
Double layer selection heuristic
Enhanced configuration checking
Obvious row weighting
Max–min ant system

ABSTRACT

The maximum set k-covering problem (MKCP) is a well-known combinatorial NP-hard problem with rich application scenarios, with the objective of covering as many elements as possible with a limited number of candidate sets. In this paper, an effective ant colony optimization (ACO) algorithm, namely MMAS-ML, is proposed to solve this important problem. Specifically, the double-layer selection heuristic is presented to obtain a high-quality solution, which is used in the initialization process to accelerate the convergence of the algorithm. Moreover, a customized local search is carefully designed, empowering ants with exploitation capacity around the food sources. For such local search procedures, some useful techniques, that is, the scoring function, delayed configuration checking, oblivious row weighting, and best from multiple selections, have been developed to enhance the performance. In addition, the max–min ant system with memory ants is further designed to confine the upper and lower bounds of the pheromone to avoid ACO stagnation. A detailed experimental evaluation of various instances reveals that the newly proposed algorithm outperforms other state-of-the-art heuristics for the MKCP, and the developed components contribute to the entire framework.



## 1. Introduction

The minimum set cover problem (MSCP) is considered a well-known combinational optimization problem, which aims to identify the set cover with the smallest number of sets [1]. Given a row set $R = \{r_1, \ldots, r_m\}$ and a column set $C = \{C_1, \ldots, C_n\}$, where each column $C_i$ is a subset of $R$, the set cover refers to finding a subset $S$ of $C$ such that the union set of $S$ is equal to $R$. As a generalized version of the MKCP, the maximum set k-covering problem of MKCP seeks to find a subset $S$ with a given cardinality $k$ such that the number of rows covered by $S$ is maximized.

Recently, the big data era has witnessed an explosion in data, which brings challenges to tackling optimization thoroughly within limited time and with limited computing resources. In this case, MKCP becomes more valuable as the flexible setting of $k$, allows a compromise between complexity and quality. Thus, it is natural that MKCP has practical significance in many real-world applications, such as maximal covering location [2], k-barrier coverage in wireless sensor networks [3], and blog monitoring applications [4]. To highlight the practical value of MKCP, a real scenario in a web search is illustrated [5]. Suppose that a search engine wishes to focus on 1000 queries to satisfy as many users as possible. Each query is considered a column and the users are represented by the rows contained in the column. This optimization problem can be viewed as MKCP.

To the best of our knowledge, there have been only two attempts to use swarm intelligence algorithms to solve MKCP [6,7], and potential improvements of these solutions await further studies. In this study, to further narrow the gap between true and generated solutions of MKCP, an algorithm combining a novel max–min ant system with an effective local search is proposed. Ant colony optimization (ACO) is a classical metaheuristic in combinatorial optimization [8], which is inspired by the foraging behavior of real ant colonies. In ACO, each ant can be regarded as a probabilistic construction heuristic that uses historical experiences, including pheromone trails and problem-specific knowledge. As one of the most popular frameworks in the swarm intelligence community, ACO has been successfully applied to many challenging optimization problems, such as the minimum weight vertex cover problem [9], the multidimensional knapsack problem [10], the maximum clique problem [11], the set covering problem [12,13], the maximum independent set problem [14], and feature selection [15,16]. For a deeper comprehension of ACO, readers are referred to surveys [17,18]. With the development of ACO, many variants of ACO have been proposed to adjust the rich optimization scenarios. Among these variants, the max–min ant system (MMAS) has attracted significant attention, which limits

* Corresponding author.
*E-mail addresses:* zhouyp605@nenu.edu.cn (Y. Zhou), liuxf469@nenu.edu.cn (X. Liu), husl903@nenu.edu.cn (S. Hu), wangyy912@nenu.edu.cn (Y. Wang), ymh@nenu.edu.cn (M. Yin).

https://doi.org/10.1016/j.knosys.2021.108000
0950-7051/

the pheromone values to an interval $[\tau_{min}, \tau_{max}]$ to avoid premature stagnation [19]. Thus, to handle MKCP, MMAS is adopted as the basic framework, and improved upon in three aspects: the initialization process, local search method, and memory ant construction.

In the initialization process, an effective double-layer selection heuristic is designed to refine the selection of the candidate columns, which extends the greedy constructive procedure adopted in [6]. After the heuristic, two types of local search methods are applied to obtain two excellent solutions. Then, with the help of the ant construction, all ants are initialized with high-quality and diversified solutions. This novel initialization heuristic can handle more complicated conditions compared with the greedy heuristic.

As for the local search method, two novel ideas are proposed to improve effectiveness. First, the multiple scoring mechanism (MSM) is introduced to determine the column to select. Second, to overcome the serious cycling problem in local search, a new version of configuration checking, namely, delayed configuration checking (DCC) is presented. In addition, the popular removal technique of the best from multiple selections (BMS) [20], is used to reduce the complexity of the removal process. Furthermore, classical edge weighting [21] is extended to solve MKCP, resulting in an oblivious row weighting (ORW) strategy.

The memory ant construction part specifies the work of each ant in the max–min ant system, where the ants are often regarded as a whole population without distinction to previous works. The ants with memories will reserve parts of their visited routes, regardless of others' pheromones. In this way, the solution construction process of an ant is accelerated, and a different search strength is achieved.

Based on the above three contributions, a max–min ant system is proposed with memory ants and a novel local search, namely MMAS-ML, to solve MKCP. It is worth mentioning that the proposed algorithm is essentially a memetic algorithm (MA), which can be viewed as a hybrid evolutionary algorithm that combines global and local search by using evolutionary or swarm intelligence algorithms to perform exploration while local search to perform exploitation. Because MAs have attracted significant attention in computing methodologies [22] and practical implementations [23], we decided to further explore combinational optimization problems.

Experiments were carried out to compare MMAS-ML with state-of-the-art MKCP algorithms on a broad range of classical benchmarks from OR-Library [24]. The experimental results show that MMAS-ML significantly outperforms previous algorithms in terms of solution quality for most instances. Specifically, MMAS-ML obtains better results than CPLEX, GOPS, RNKC, ABPSO, and HBABC for 136, 150, 95, 25, and 24 instances, respectively. Moreover, MMAS-ML also outperforms two MSCP algorithms, uSLC [25] and RWLS [26], for all instances.

The main contributions are outlined as follows:

1. First, with regard to solution methods, a max–min ant system is proposed with memory ants and a novel local search, namely MMAS-ML, to solve MKCP. Five valid strategies are designed, including the double-layer selection heuristic, the multiple scoring mechanism of the local search, delayed configuration checking, oblivious row weighting, and memory ant constructions.
2. Second, the computational performance of the proposed algorithm is assessed by comparing with seven state-of-the-art algorithms on a set of 150 benchmark instances. The experimental evaluation reveals that the proposed algorithm outperforms other state-of-the-art heuristics for the MKCP.
3. Third, three key components of MMAS-ML have been investigated and discussed in the form of experimental verification. The results show that all of the key components of MMAS-ML play an important role in the proposed algorithm, making MMAS-ML competitive.

The remainder of this paper is structured as follows. In Section 2, related work is reviewed. In Section 3, preliminary knowledge is introduced. In Section 4, the entire framework of the proposed MMAS-ML algorithm is given. In Section 5, the details of the double-layer selection heuristic, the local search and the ant construction process are presented. In Section 6, the computational results are reported and discussed. Finally, conclusions are drawn and future work is suggested.

## 2. Related work

It is well known that MKCP is an NP-hard problem [1], which means that no polynomial-time algorithms exist unless NP=P. To this end, approximation algorithms have been designed to maintain the approximation guarantee. A (1-1/$e$)-approximation greedy algorithm [27] was used to solve MKCP. Subsequently, Chierichetti et al. [5] improved the approximation ratio to approximately 0.63 by a map reduce-based algorithm. Yu and Yuan [28] then proposed a single-pass algorithm with an approximation factor of approximately 0.3. Until recently, two (1-1/$e$-$\epsilon$)-approximation algorithms [29] were designed for a small value of $\epsilon$. For large and hard instances however, approximation algorithms usually exhibit poor performance in practice.

To address this, some researchers focused on heuristic algorithms to obtain a good solution within an acceptable time for MKCP. In 2018, Wang et al. [30] designed a restart local search algorithm to solve MKCP and compared the results with a high-performance mathematical programming solver CPLEX.[1] They generated competitive solutions for the lower and upper bounds of the tested instances. It is worth noting that the proposed local search was efficient during the optimization process, which made a breakthrough in tackling MKCP in a short time. They first used a random restart method to overcome the serious cycling problem in which the algorithm returns to a candidate solution that has already been visited. To further improve the solution quality, a novel neighborhood search was applied to explore a more feasible search space. Noting the significance of MKCP and the gap between the approximate and optimal solutions, another heuristic algorithm, adaptive binary particle swarm optimization (ABPSO) [6], was proposed. They integrated a greedy constructive procedure and a fast iterative local search into the binary particle swarm optimization, and then applied adaptive mutation and gain updating techniques to further improve the effectiveness of ABPSO. Based on these techniques, the proposed ABPSO achieved better performance on a wide range of benchmarks. In a recent study, Lin et al. proposed a hybrid binary artificial bee colony algorithm (HBABC) to solve MKCP [7]. They developed a new food source updating method and tabu-based simulated annealing to balance the intensification and diversification of the algorithm. Experimental results show that HBABC improves the solution, greatly advancing the research on MKCP.

## 3. Preliminaries

Given a finite row set $R = \{r_1, \ldots, r_m\}$ of $m$ elements, the column set is defined as $C = \{C_1, \ldots, C_n\}$ of $n$ columns, where each column $C_i = \{r_{i_1}, \ldots, r_{i_{k_i}}\}$ is a subset of $R$. To present the inclusion relation between the row and column sets, an $m$-row,

[1] https://www.ibm.com/products/ilog-cplex-optimization-studio.

$n$-column, zero–one matrix $M = (m_{i,j})$ is introduced. $m_{i,j} = 1$ indicates that the $i$th row is covered by the $j$th column. Thus, the matrix $M$ is considered to be a look-up table that can provide the covering information of the sets within $O(1)$ time. In addition, two Boolean variables $x_i$ and $y_j$ are imported to denote whether $r_i$ is covered, and $C_j$ is selected. The following definitions are based on this notation.

**Definition 1** (*Minimum Set Cover Problem, MSCP*)**.** Given a row set $R$ and a column set $C$, the set cover problem is to find a nonempty subset $S$ of $C$, such that the union set of $S$ is equal to $R$. In particular, when the aim of the optimization is to seek the set $S'$ of the minimum cardinality among all sets $S$, it is called the minimum set cover problem.

**Definition 2** (*Maximum Set k-Covering Problem, MKCP*)**.** Given a row set $R$ together with a column set $C$, the maximum set k-covering problem aims to find a subset of $C$ with cardinality $k$ to cover as many rows as possible.

The MKCP can be represented through the following integer programming formulations:

$$\text{Maximize } F(X) = \sum_{i=1}^{m} x_i \tag{1}$$

Subject to

$$\sum_{j=1}^{n} m_{i,j} y_j \geq x_i, \forall i \in R \tag{2}$$

$$\sum_{j=1}^{n} y_j = k \tag{3}$$

$$y_j \in \{0, 1\}, \ \forall j \in C \tag{4}$$

$$x_i \in \{0, 1\}, \ \forall i \in R \tag{5}$$

Eq. (1) describes the objective function of MKCP. As for the constraints, Eq. (2) defines the relation between $x_i$ and $y_j$. Eq. (3) requires that only $k$ subsets are selected in the solution. Eqs. (5) and (6) stipulate the domains of the variables. From the definitions of MSCP and MKCP, it is easy to see that MKCP can be transformed into MSCP when the value of $k$ is sufficiently large such that all elements in $R$ are covered. Thus, it can be concluded that MKCP is a general version of MSCP, which plays a significant role in the set covering family. The main distinction between MSCP and MKCP is that MSCP seeks full coverage of the row set $R$, whereas MKCP focuses on covering as many elements as possible with limited subsets of $C$. In other words, the optimization of MSCP should pay special attention to the unique sets even if they contain a small number of elements. However, for MKCP, the primary task is to select the sets with the maximum benefits.

MKCP is illustrated on an example in Fig. 1 to enhance comprehension.

One of the minimum set covers of the column set $C = \{C_1, C_2, \ldots, C_{13}\}$ is $S = \{C_1, C_2, C_3, C_5, C_6, C_7, C_8\}$ with an objective of 7. For the optimization of MKCP when $k = 4$, one of the maximum set k-coverings is $S_a = \{C_2, C_3, C_4, C_5\}$ with an objective of 12. When $k = 5$, the solution is $S_b = \{C_2, C_3, C_5, C_6, C_7\}$ with an objective of 15. Obviously, compared with $S_a$, the structure of $S_b$ changes: column $C_4$ needs to be removed from $S_a$, and the other two columns $C_6$ and $C_7$ should be added. This indicates that the maximum set $k$-covering problem cannot be optimized by simply adding an appropriate column into a maximum set ($k$-1)-covering solution.

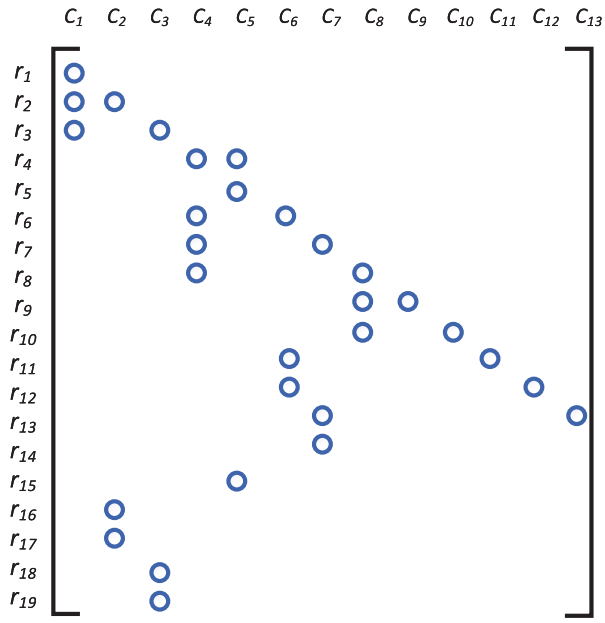


**Fig. 1.** A 19-row, 13-column, 0–1 matrix of MKCP .

## 4. The MAX-MIN ant system with memory ants for MKCP

The max–min ant system with memory ants for MKCP is introduced as the top-level algorithm in this section. After a brief background on the framework, details related to the important components of MMAS-ML are discussed in the next section.

**Algorithm 1:** The framework of MMAS-ML

**Input:** element-set matrix $M$
**Output:** current best solution $S_g$
1 initialize pheromone trails and parameters;
2 $S_0 := DoubleLayerSelection(M)$;
3 $S_g := ObliviousLS(S_0)$, $S'_g := LS(S_0)$;
4 $[SS_1 : SS_2] := AntConstruction([S_g : S'_g], \tau, \eta, \chi)$;
5 **for** $t = 1, 2, \cdots, |Ants|$ **do**
6 **if** $S_t \in SS_1$ **then** $S_t := ObliviousLS(S_t)$ ;
7 **else if** $S_t \in SS_2$ **then** $S_t := LS(S_t)$ ;
8 $S_g := \arg\max_{S \in [SS_1 : SS_2]} F(S)$; // Eq. (1)
9 **while** *termination criterion is not met* **do**
10 **for** $t = 1, 2, \cdots, |Ants|$ **do**
11 $S_t := AntConstruction(S_g, \tau, \eta, \chi)$;
12 **if** $S_t \in SS_1$ **then** $S_t := ObliviousLS(S_t)$;
13 **else if** $S_t \in SS_2$ **then** $S_t := LS(S_t)$ ;
14 **if** $F(S_g) < F(S_t)$ **then** $S_g := S_t$ ;
15 update pheromone trails according to Eq. (7).;
16 **return** $S_g$;

The main framework of MMAS-ML is presented in Algorithm 1. The algorithm loads the matrix $M$ and outputs the current best solution whereby the output matrix captures the characteristics of different instances. Specifically, initializations of the parameters and pheromone trails are carried out at the beginning (line 1). Then, $S_0$ is produced by the double-layer selection heuristic (see Section 5.1) as the reference solution of the initial population (line 2). Based on the reference solution, two global best solutions $S_g$ and $S'_g$ are generated through two types of local search procedures, namely *obliviousLS*() and *LS*() (line 3). Note that the

local search adopted in line 3 applies fewer iterations than the method discussed in Section 5.3, saving computational resources while maintaining a high-quality solution. *ObliviousLS*, which has objective function Eq. (8), is stated in Section 5.3.4, and *LS* is the version of *obliviousLS* without the oblivious row weighting scheme and optimizes Eq. (1). Obviously, *LS* pays more attention to the original objective function itself because the objective function guides the search direction using Eq. (1), and thus is not interfered with by uncovered rows. Because the local search procedures are handled separately, the ant colony is divided into two categories, $S$ and $S'$, in which each ant has a different memory level, that is, $\chi\% = 0\%, 0\% - -50\%, 50 - -100\%$, and 100% (line 4). Thus, in our study, eight ants are used in MMAS-ML. More details on memory ants are described in Section 5.2. These two categories of ants perform different local search processes to fully explore the search space (lines 5–7). In the subsequent ant colony iteration, the current best solution is selected from the joint population of all ants and is used as the input of the overall ant colony during the next iteration (line 8).

In lines 9–15, the iterative process loops until the termination criterion is satisfied. In this study, the stopping criterion was set as the time limit. During one loop, each ant constructs a feasible solution $S_t$ based on three factors: the pheromone factor $\tau$, heuristic factor $\eta$, and memory level $\chi$ (line 11). Then, the first group of ants $SS_1$ adopts the oblivious local search to improve the current solution (line 12), while the second group of ants $SS_2$ perform a different local search on the current solution (line 13). After the application of these two local search methods, the global best solution $S_g$ is updated for the next ant construction procedure (line 14). At the end of each iteration, the pheromone trails are renewed by the global best solution, as shown in Eq. (7) (line 15), which is used in the elitist ant system to increase the efficiency of ACO by intensifying the search near the global best solution. Finally, $S_g$ is returned as the best solution that MMAS-ML has found (line 16).

To determine the efficiency of the proposed algorithm, the time complexity of MMAS-ML is analyzed as follows. As the initialization is performed only once in the entire procedure, the discussion only focuses on the time complexity of the MMAS and local search in each iteration. The notation adopted is as follows: $k$ is the number of columns to select; $n$ is the cardinality of the column set $C$; and *Iter* is the number of iterations for the local search. For MMAS, the ant construction is bounded in $O(k * n)$, and the pheromone is updated in $O(n)$ time. For the local search process, the total time complexity is $O(Iter * n)$. Finally, the time complexity of a single MMAS-ML iteration is $O(k * n + Iter * n)$.

## 5. Main components of MMAS-ML

### 5.1. The double layer selection heuristic

Although the metaheuristics can be adapted to solve different optimization problems without significant changes, convergence is usually slow because of a lack of knowledge on the specific problem. In this case, heuristics are necessary to deliver the information reflecting the comprehension of domain experts into the optimization procedure.

To solve MKCP, a natural idea is to add the worthiest column each time, which covers most of the rows in the current uncovered row set. This greedy heuristic has been used in previous algorithms such as RNKC [30] and ABPSO [6], and has achieved good results. However, the selection of the worthiest column may be uncertain because more than one valuable column can exist at the same time. Therefore, additional criteria are proposed for selecting promising columns to cover more rows. The pseudocode for the double-layer selection is given in Algorithm 2. Before proceeding, some definitions are provided to describe the benefit of the columns to be selected for the candidate solution.

**Definition 3** (*Gain of Column*)**.** The gain of the column $C_i$, denoted as $gain(C_i)$, is defined as the number of uncovered rows that will be covered when the column $C_i$ is added into the solution.

**Definition 4** (*Independent and Pseudo-Independent Rows*)**.** The row $r_i$ is called the independent row iff $r_i$ is covered by only one column of $C$. In particular, the row $r_j$ is called the pseudo-independent row if it is covered by only one column of $C \backslash C^l$, where $C^l \subseteq C$.

Compared with the independent rows, the pseudo-independent rows have more relaxed constraints. That is, if some columns are blocked (denoted as $C^l$) from $C$, there exist independent rows, referred to as the pseudo-independent rows. Given these definitions, the double layer selection heuristic is used to construct a promising solution using the following rules: (1) In the first level (lines 3–5), only one column with the largest $gain(C_j)$ is added to the solution. If more than one column exists, record these sets in $C'$ and go to the second level. (2) In the second level (lines 7–15), the column with the most independent rows in $C'$ is selected for the solution. If more than one column in $C''$ exists, randomly select one column (lines 7–9). Otherwise, if no independent rows exist, select the row with the most pseudo-independent rows in $C'$. Note that the pseudo-independent row is covered by only one column in $C'$. In addition, if more than one column in $C'''$ exist, randomly select one from $C'''$, or else select the column with the largest cardinality from $C'$ (lines 11–15). This heuristic is executed iteratively until $k$ columns are selected. Finally, the solution $S_0$ is returned.

**Algorithm 2:** *DoubleLayerSelection*

```
   Input: element-Set matrix M
   Output: initial solution S_0
1  Initialize S_0 = ∅;
2  while |S_0| < k do
3      C' := {C_i | gain(C_i) ≥ gain(C_j), C_i ∈ C, ∀C_j ∈ C};
4      if |C'| = 1 then
5          select only one column C_r in C';
6      else
7          C'' := columns with most independent rows in C';
8          if |C''| ≥ 1 then
9              select a random column C_r in C'';
10         else
11             C''' := columns with most pseudo-independent
                   rows in C';
12             if |C'''| ≥ 1 then
13                 select a random column C_r in C''';
14             else
15                 select a random column C_r with most
                       elements in C';
16     S_0 := S_0 ∪ C_r;
17 return S_0;
```

From a comprehensive perspective, the proposed method adds columns with the most uncovered rows or columns that can cover most independent and pseudo-independent rows. The method also considers the diversification of $S_0$, as it randomly selects the columns in certain cases. Take Fig. 1 for example, the double layer selection can find the maximum set k-covering $S_a = \{C_2, C_3, C_4, C_5\}$ with the objective value of 12 when $k = 4$, while the previous greedy heuristic only had a probability of 2/3 to obtain the same solution.

The gain values are calculated once the instance is input, and the largest gain values of the columns are determined in $O(n)$ time in this process (Line 3). Considering the time complexity of calculating the number of independent rows in each set, finding the columns with the most independent rows takes $O(m*n)$ time (Line 7). Similarly, the pseudo-independent rows have a time complexity of $O(m*n)$ (Line 11). As the gain values should be dynamically updated when the solution is changed, the operation in Line 16 takes $O(m*n)$ time. Given the above, the double-layer selection has a time complexity of $O(k*m*n)$.

### 5.2. Memory ant construction

Ant colony optimization is a popular swarm-based algorithm that is suitable for combinatorial optimization problems. The main idea of ACO is to model the problem by searching for a minimum cost path in a graph by a colony of artificial ants. The ants find the optimal path by communicating through artificial pheromone trails. For MKCP, the construction of the graph by each ant is a $k$-step process, where one vertex is selected at each step according to the weight of the edges. In the constructed graph, the vertex is the column of MKCP, and the weight of each edge is defined as the probability that measures the pheromone trails and problem-dependent heuristic information. When the ant determines a feasible solution, it releases pheromones on traveled vertices, which helps the ants move towards promising areas. The pheromone manifests the cooperation of the ant colony and accumulates gradually as the iteration increases. With the development of ACO research, MMAS [19] has been proposed to deal with early stagnation by imposing upper and lower bounds of pheromone trails.

Suppose that $\tau_j$ denotes the pheromone of column $C_j$, and $\eta_j$ represents the heuristic information on $C_j$ in MKCP. Given a partial solution $S_t$ coupled with the complementary solution $\bar{S}_t$, the state transition rule for ants can be set as follows:

$$P_j = \begin{cases} \frac{\tau_j \cdot \eta_j^{\beta}}{\sum_{i \in \bar{S}_t} \tau_i \cdot \eta_i^{\beta}}, & \text{if } C_j \in \bar{S}_t \\ 0, & \text{if } C_j \notin \bar{S}_t \end{cases} \tag{6}$$

where the parameter $\beta$ controls the heuristic strength in MMAS. The heuristic information $\eta_j$ is equal to the fitness gain after a certain operation. This transition rule states that column $C_j$ can be selected as the solution with a probability of $P_j$. The pheromone updating rule is accumulated by the global best ant.

$$\tau_j = \begin{cases} \tau_{min}, & if\ \tau_j \le \tau_{min} \\ \tau_{max}, & if\ \tau_j \ge \tau_{max} \\ \rho \cdot \tau_j + \Delta\tau_j, & otherwise \end{cases} \tag{7}$$

where $\rho$ is the pheromone volatilization coefficient, and $\Delta\tau_j$ is the amount of pheromone released on column $C_j$ by the global best ant.

After introducing the preliminaries of MMAS, an improvement to MMAS, called MMAS with memory ants (MMAS-ML), is presented. In MMAS-ML, each ant is noticed by the best ant at the end of each path-finding process and has certain memories of the best path. Meanwhile, each ant has its own work and adopts different searching modes to expand the search area.

Different from the original ant construction process, each ant has a memory of the learned information from the best ant, and the memory effect varies for different ants. Specifically, there are four memory levels in MMAS-ML: $\chi_1\%$, $\chi_2\%$, $\chi_3\%$, and $\chi_4\%$. The first level $\chi_1\% = 0$ is a new reconstruction of the solution based on the pheromone and heuristic information. In contrast, the ant with the fourth level $\chi_4\% = 100\%$ inherits the same solution from the best ant to prevent insufficient local searches. The remaining two levels maintain parts of the good structures of the best ant as well as the diversification capacity of ACO, which results in a faster convergence of MMAS-ML. The ant with $\chi_2\% = rnd\%$, where $rnd\%$ is a random number between 0 and 50% and shares $rnd\%$ of the same columns of the best ant. Likewise, $\chi_3\% = rnd'\%$, where $rnd'\%$ is a random number between 50% and 100%. The pseudocode for the memory-ant construction procedure is given in Algorithm 3.

**Algorithm 3:** *AntConstruction*

**Input:** $S_g$, $\tau$, $\eta$ and $\chi$
**Output:** $S_t$
1 $S_t = \emptyset$;
2 generate a memory level $\chi\%$;
3 select $\chi\%$ of columns from $S_g$ randomly;
4 add the selected columns into $S_t$;
5 **while** $|S_t| < k$ **do**
6  select $C_j \in C \setminus S_t$ by roulette according to Eq. (6).;
7  $S_t := S_t \cup C_j$;
8 **return** $S_t$;

### 5.3. The score-based local search with delayed configuration checking

Although MMAS improves the solution by the positive feedback of pheromone accumulation, the optimal solution may be unpolished and time-consuming to determine. To address this issue, a local search is proposed to remedy the exploitation capacity of MMAS. In other words, each ant should learn to search the surrounding space of the food sources carefully to generate a better pheromone for the whole population. The strategies of the local search for MKCP are first described, and then the entire local search framework is provided.

#### 5.3.1. Delayed configuration checking

Configuration checking (CC) is an efficient strategy for dealing with the cycling problem and has achieved very good results for the minimum vertex cover problem [31]. The CC strategy maintains a vector of Boolean values, which represent the configuration of each variable. The configuration of one variable determines whether this variable is operational and can be dynamically changed according to the neighborhood state of the variable. Prior to stating the CC rules for MKCP, the key concept of the neighborhood of the column is defined.

**Definition 5** (*Neighborhood of Columns*)**.** The neighborhood of the column $C_i$ is defined as the whole columns $C^N$ of $C$, such that each column $C_j \in C^N$ covers at least one row that is also contained by $C_i$. Formally, $Neighbor(C_i) = \{C_j | C_i \cap C_j \neq \emptyset, C_i \in C, C_j \in C, i \neq j\}$.

Given this definition, a Boolean array, denoted as $conf$, is introduced to determine whether the column can be operated on at that iteration. The delayed configuration checking (DCC) was maintained using the following rules:

**Rule 1:** The configuration of all columns, namely $conf$, is initialized as true, indicating that all columns can be added into the solution.

**Rule 2:** When column $C_i$ is removed from the solution, its configuration $conf[C_i]$ is set to false to prevent from being added back to the solution immediately. In addition, all the neighborhoods of that column, $Neighbor(C_i)$, set their configuration as true.

**Rule 3:** When column $C_i$ is added to the solution, its configuration $conf[C_i]$ is set to false temporarily to restraint the removal of $C_i$ in the next round. Furthermore, the previous column with a temporary false value of the configuration is reset to true. Meanwhile, all the neighborhoods $Neighbor(C_i)$ set the configuration as true.

The difference between DCC and the classical CC is the enhanced limitation of Rule 3, which prevents an added column $C_i$ from being deleted again in the next iteration. Of course, this configuration $conf[C_i]$ is maintained as false until the next column $C_j$ is added, to avoid the condition that no candidate columns exist to be removed.

#### 5.3.2. Oblivious row weighting

In Definition 3 above, the gain of column $C_i$ reflects the real benefits $C_i$ can bring to the objective function. However, this gain may lead to a local optimum, as some fixed columns are often selected. Therefore, the row weighting strategy is imported to increase the importance of "inactive" rows, as adopted in [32]. Note that the importance of "inactive" rows should not be increased without limit, so in this study, oblivious row weighting is proposed to set an upper bound on the inactive level for each row.

**Definition 6** (*Inactive Level*)**.** For each row $r_i \in R$, the inactive level is denoted as $Inal[r_i]$, which is initialized to 1 at the beginning. At the end of each iteration, for each uncovered row $r'_j$, $Inal[r'_j]$ is increased by 1. When the inactive level $Inal[r'_j]$ reaches the threshold *ts*, it begins to reduce parts of the level, that is, $Inal[r'_j] = Inal[r'_j] \times \delta$, where $\delta \in (0, 1)$ is the oblivious ratio.

In brief, uncovered rows will continually increase the likelihood of being selected into the solution through an oblivious row weighting scheme. To balance the unfair increment speed between covered and uncovered rows, the value of $Inal[r'_j]$ is decreased once it reaches the threshold. Based on the definition of the inactive level, the final fitness function of each solution $S$ is determined as follows:

$$fitness(S) = \sum_{j=1}^{m} Inal[r_j] * x_j \tag{8}$$

Although the oblivious row weighting strategy concentrates on the balance of the intensification and diversification of the selection process, it is difficult to seize the opportunity when the inactive rows dominate the frequently covered rows. In this case, a variant of local search is shown in Algorithm 4, namely *ObliviousLS*, is applied to guarantee a detailed search of the promising areas by eliminating the strategy of the oblivious row weighting. Thus, the ants are divided into two groups, one for *ObliviousLS* and the other for *LS*.

#### 5.3.3. Multiple scoring mechanism

To actuate the local search, the scoring function is often applied to decide which neighborhoods to move forward to. Intuitively, the fitness value is assumed to be an adequate metric to guide the search direction. To be specific, the scoring function of the column $C_i$ can be stated as:

$$score(C_i) = fitness(S') - fitness(S) \tag{9}$$

where $S'$ is the solution after the operation (add or remove) of $C_i$, and $S$ is a candidate solution. Each *score* value of the column is initialized as the cardinality of the column and is updated dynamically to save computation time. Also, as Eq. (9) shows that the *score* value of the column to be added is positive, while for the column to be removed it is negative. Therefore, the column with the largest value of *score* is operated on during the approximation of the optimal solution.

After using the above scoring method, it was determined that a number of columns have the same *score* values. To this end, a multiple-scoring mechanism is required to break the ties. In general, subscores are often employed when the score functions have the same value [33,34]. The evaluation of the subscore differs as

**Algorithm 4:** *ObliviousLS*

**Input:** current solution $S_t$
**Output:** improved solution $S_t$
1 **while** *termination criteria not met* **do**
2 $C_j$=*BMS*($S_t$) based on the selection rule;
3 $S_t := S_t \setminus C_j$;
4 update *Inal*, *score* and *subscore*;
5 update the DCC array *conf*;
6 select one uncovered row $r_i$ randomly;
7 $C' := C \setminus S_t$ with $r_i$ being covered;
8 select $C'_j \in C'$ based on the selection rule;
9 $S_t := S_t \cup C_j$;
10 update *Inal*, *score* and *subscore*;
11 update the DCC array *conf*;
12 **if** $F(S_g) < F(S_t)$ **then** $S_g := S_t$ ;
13 **return** $S_t$.

the optimization problems change. In MKCP, the independent row of the solution is another important factor that should be considered in the local search procedure. The independent rows are identified as important because the number of covered elements decreases when the column with independent rows is removed from the candidate solution. Suppose that the number of times row $r_j$ is covered is $CT(r_j)$, then two functions called $oneTotwo(C_i)$ and $twoToone(C_i)$ can be defined. $oneTotwo(C_i)$ counts the number of rows from $CT = 1$ to $CT = 2$ after adding $C_i$, whereas $twoToone(C_i)$ calculates the number of rows from $CT = 2$ to $CT = 1$ after removing $C_i$. Then, the subscore is defined as:

$$subscore(C_i) = \begin{cases} oneTotwo(C_i) \text{ if } C_i \text{ is added} \\ -twoToone(C_i) \text{ if } C_i \text{ is removed} \end{cases} \tag{10}$$

A larger *subscore* value is preferred to make the solution robust when the score values are equal. The combination of *score* and *subscore* provides a comprehensive estimation of the columns, which makes the selection process effective and robust. Finally, the selection rule to decide which column to choose, is presented as follows:

**Selection Rule:** Select one column $C_i \in C$ with the largest *score* value and $conf[C_i]$ true, and in case of ties, pick the column with the largest *subscore* value. If more than one selected column has the same maximum *subscore* and the same maximum *score*, the oldest column, that is, the one that has not been changed for the longest time is selected.

#### 5.3.4. Framework of the local search

The framework of the oblivious local search is briefly demonstrated from an overall perspective (Algorithm 4).

During each iteration, a destroy-and-rebuild process is executed to dislodge the current solution to its neighborhood solutions, which can improve the number of covered rows (lines 1–11). To handle large-scale instances, the *BMS* strategy [20] is applied such that the search space is shrunk to a reasonable size (line 2). As the name indicates, *BMS* chooses the best solution from multiple $\mu$ selections. The method of selecting the best solution obeys the rules defined in the multiple scoring mechanism (Section 5.3.3). It should be noted that during the selection process in *BMS*, only the column labeled $conf[C_j] = true$ can be selected. Once column $C_j$ is picked, it is removed from the current solution $S_t$ (line 3). After removal, the *score*, *subscore*, and *Inal* are updated (line 4). In addition, the delayed configuration arrays of the operated column along with its neighborhoods are also updated, as shown in Section 5.3.1 (line 5). In contrast to the removal operator, the add operator does not perform *BMS*

**Table 1**
Parameter tuning results of MMAS-LS.

| Parameter | Description | Range | Final setting |
|---|---|---|---|
| $\delta$ | Forgetting coefficient | {0.1, 0.3, 0.5} | 0.3 |
| $ts$ | Oblivion upper bound | {500, 1000, 2000} | 1000 |
| $\mu$ | BMS value | {1/3, 1/2, 2/3, 3/4} | 2/3 |
| $Iter$ | Local search termination condition | $\{10^4, 5 \times 10^4, 10^5\}$ | $5 \times 10^4$ |
| $\rho$ | Volatilization coefficient | {0.90,0.95,0.99} | 0.95 |
| $\beta$ | Relative importance of heuristic factor | {0, 1, 2, 3} | 1 |

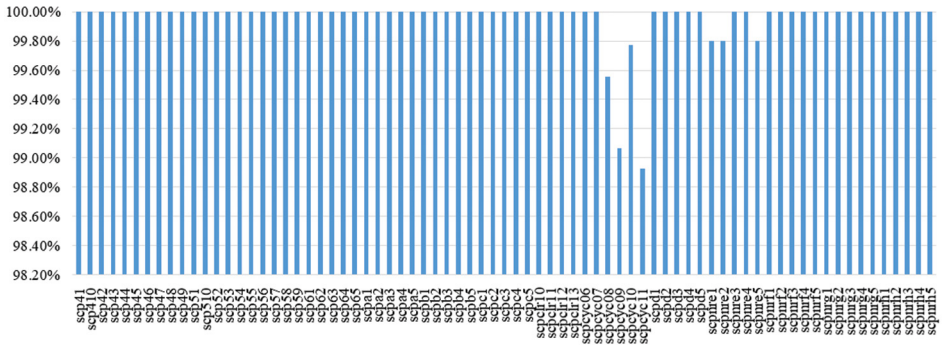


**Fig. 2.** The coverage rate of MMAS-ML for MSCP.

because it uses substitution to reduce the solution space. Specifically, an uncovered row $r_i$ is randomly selected, which is to be covered next (line 6). Then, all the columns $C'$ that cover $r_i$ are collected, except for those selected in $S_t$ (line 7). The multiple scoring mechanism is applied to the collected columns once again to select the best column to add (line 8). Of course, the selected column should satisfy the constraint $conf[C'_j] = true$. When the selected column $C_j$ is added to the current solution (line 9), a similar updating process is performed, as mentioned before (lines 10–11). The newly generated solution $S_t$ is checked to determine whether the global best solution can be updated. If so, $S_g$ is replaced with $S_t$ (line 12). The entire local search process terminates when the solution has not been improved for *Iter* times. After the stop criterion is met, $S_t$ is returned as the output (line 13). The destroy-and-rebuild method ensures that the generated solution is feasible.

## 6. Experimental studies

MMAS-ML was evaluated on a large number of instances from the OR-Library [24] by comparing it with state-of-the-art algorithms. In addition, additional experiments were conducted to analyze the impact of the main components of MMAS-ML, that is, the double-layer selection heuristic, the max–min ant system framework, and the score-based local search.

### 6.1. Experiment preliminaries

To verify the effectiveness of MMAS-ML, it was compared with four incomplete algorithms, namely, GOPS [28], RNKC [30], ABPSO [6], and HBABC [7]. The results from an exact solver called CPLEX (a public high-performance mathematical programming solver) were also compared. MMAS-ML and all competitors were implemented in C++ and compiled by g++ with '-O3' option. All the experiments were performed on an Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20 GHz with 128 GB RAM under CentOS 6.9.

All algorithms were evaluated on a large number of instances from the OR-Library [24]. For each instance, two values of $k$ were considered, ($k_{90}$ and $k_{95}$), indicating that only 90% and 95% of the columns in the best-known solutions of MSCP [26] are selected. Detailed information on a total of 150 instances, including the number of elements, columns, and density, can be found in [6].

All the algorithms were performed 10 times for each instance within a cutoff time of 100 s. The following indicators are reported: the best solution ("*BEST*") among 10 runs, the corresponding average solution ("*AVG*"), and the average run time ("*TIME*") for when the algorithm obtains the maximal solution value. In addition, "*#Better*" indicates the number of instances where MMAS-ML finds better maximal(average) solutions, while "*#Worse*" indicates the number of instances where MMAS-ML finds worse maximal(average) solutions. "*#Draw*" calculates the number of instances where MMAS-ML does not significantly differ from its variants.

### 6.2. Parameter tuning

In this subsection, the automatic configuration tool irace [35] was used to find well-working values for the parameters of MMAS-ML. Table 1 summarizes the parameters of the MMAS-ML algorithm to be adjusted, i.e., $\delta$, $ts$, $\mu$, $Iter$, $\rho$, $\beta$. The training set was restricted to 14 representative training instances selected from all 14 categories of the tested instances. The tuning process was given a budget of 5000 runs of MMAS-ML, with a cutoff time of 100 s for each run. Table 1 lists the final choices of the parameter values. From the table, it is concluded that an appropriate combination of the parameter values is $p = 0.3$, $ts = 1000$, $\mu = 2/3$, $Iter = 5 \times 10^4$, $\rho = 0.95$, $\beta = 1$. The final version of MMAS-ML used in subsequent experiments was based on these values.

### 6.3. Computational results and comparison

The experimental results show that MMAS-ML obtains better solutions than CPLEX, GOPS, and RNKC for 136, 150, and 95 instances, respectively. Thus, the focus here is on comparing MMAS-ML against ABPSO and HBABC.

As Table 2 illustrates, MMAS-ML obtains 11 better solutions than ABPSO on $k_{90}$ instances in terms of the "BEST" indicator, except for scpnrf1 and scpcyc10. For other instances where MMAS-ML and ABPSO can both find the same fitness values, MMAS-ML achieves a better average solution quality, indicating that MMAS-ML is more robust than ABPSO. In particular, the

**Table 2**
Results of MMAS-ML, GOPS, RNKC, ABPSO, HBABC and CPLEX on instances with $k_{90}$.

| Eg | CPLEX | | GOPS | | RNKC | | | ABPSO | | | HBABC | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LB | UB | BEST | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scp41 | 192 | 195.42 | 174 | 0.009 | 192 | 191.9 | 3.09 | **193** | **193** | 7.16 | **193** | **193** | 3.22 | **193** | **193** | 2.58 |
| scp42 | **193** | **193** | 179 | 0.01 | **193** | **193** | 3.23 | **193** | **193** | 0.03 | **193** | **193** | 0.02 | **193** | **193** | 0.93 |
| scp43 | 193 | 195.21 | 178 | 0.008 | 193 | 193 | 0.98 | **194** | **194** | 0.40 | **194** | **194** | 0.05 | **194** | **194** | 1.09 |
| scp44 | **191** | 194.92 | 180 | 0.01 | **191** | 190.4 | 37.40 | **191** | **191** | 0.12 | **191** | **191** | 0.06 | **191** | **191** | 0.96 |
| scp45 | 192 | 195.72 | 179 | 0.011 | 192 | 192 | 3.86 | **193** | **193** | 0.91 | **193** | **193** | 0.37 | **193** | **193** | 1.38 |
| scp46 | **191** | 193.95 | 177 | 0.01 | **191** | **191** | 1.95 | **191** | **191** | 0.04 | **191** | **191** | 0.02 | **191** | **191** | 0.95 |
| scp47 | **191** | 193.5 | 176 | 0.011 | **191** | **191** | 13.69 | **191** | **191** | 0.13 | **191** | **191** | 0.02 | **191** | **191** | 1.01 |
| scp48 | 190 | 195.46 | 178 | 0.009 | **192** | 190.8 | 3.12 | **192** | **192** | 0.21 | **192** | **192** | 0.20 | **192** | **192** | 1.12 |
| scp49 | **192** | 194.91 | 183 | 0.012 | **192** | 191.4 | 48.57 | **192** | **192** | 0.07 | **192** | **192** | 0.02 | **192** | **192** | 0.98 |
| scp410 | **192** | 193.22 | 180 | 0.009 | **192** | 191.1 | 17.09 | **192** | **192** | 1.04 | **192** | **192** | 1.13 | **192** | **192** | 2.70 |
| scp51 | 192 | 200 | 182 | 0.039 | **193** | 191.9 | 3.89 | **193** | **193** | 0.19 | **193** | **193** | 0.16 | **193** | **193** | 1.58 |
| scp52 | 191 | 200 | 182 | 0.033 | 193 | 192.6 | 33.36 | **194** | **194** | 2.26 | **194** | **194** | 2.39 | **194** | **194** | 1.99 |
| scp53 | 192 | 199.72 | 181 | 0.024 | **194** | 193.8 | 34.91 | **194** | **194** | 0.28 | **194** | **194** | 0.14 | **194** | **194** | 1.54 |
| scp54 | 190 | 199.47 | 180 | 0.029 | 193 | 193 | 35.18 | **194** | **194** | 0.29 | **194** | **194** | 0.19 | **194** | **194** | 1.77 |
| scp55 | 192 | 200 | 180 | 0.025 | **194** | 193.9 | 30.28 | **194** | **194** | 0.27 | **194** | **194** | 0.15 | **194** | **194** | 1.63 |
| scp56 | 193 | 199.83 | 179 | 0.024 | 193 | 193 | 23.98 | **194** | **194** | 0.34 | **194** | **194** | 0.07 | **194** | **194** | 1.74 |
| scp57 | 193 | 200 | 183 | 0.022 | **195** | 194.1 | 32.23 | **195** | **195** | 0.48 | **195** | **195** | 0.30 | **195** | **195** | 1.92 |
| scp58 | 193 | 199.19 | 178 | 0.022 | 193 | 192.4 | 34.82 | **194** | **194** | 0.24 | **194** | **194** | 0.41 | **194** | **194** | 1.73 |
| scp59 | **194** | 200 | 184 | 0.023 | **194** | **194** | 5.77 | **194** | **194** | 0.13 | **194** | **194** | 0.04 | **194** | **194** | 1.19 |
| scp510 | 190 | 199.65 | 177 | 0.024 | 193 | 192.1 | 54.74 | **194** | 193.9 | 3.05 | **194** | **194** | 4.08 | **194** | **194** | 2.94 |
| scp61 | 192 | 200 | 182 | 0.016 | **196** | **196** | 1.63 | **196** | **196** | 0.08 | **196** | **196** | 0.15 | **196** | **196** | 1.56 |
| scp62 | 191 | 200 | 176 | 0.018 | **195** | **195** | 13.28 | **195** | **195** | 0.26 | **195** | **195** | 0.91 | **195** | **195** | 1.87 |
| scp63 | 194 | 200 | 188 | 0.02 | **196** | **196** | 3.35 | **196** | **196** | 0.07 | **196** | **196** | 0.14 | **196** | **196** | 1.35 |
| scp64 | 187 | 200 | 175 | 0.025 | **192** | **192** | 5.66 | **192** | **192** | 0.09 | **192** | **192** | 0.09 | **192** | **192** | 1.30 |
| scp65 | 193 | 200 | 179 | 0.026 | **196** | 195.8 | 23.83 | **196** | **196** | 0.34 | **196** | **196** | 0.60 | **196** | **196** | 1.77 |
| scpa1 | 283 | 300 | 263 | 0.073 | 288 | 286.8 | 49.81 | **289** | 288.9 | 7.34 | **289** | **289** | 5.91 | **289** | **289** | 7.47 |
| scpa2 | 283 | 300 | 264 | 0.085 | 288 | 286.6 | 19.89 | **289** | **289** | 15.02 | **289** | **289** | 3.28 | **289** | **289** | 7.19 |
| scpa3 | 283 | 300 | 272 | 0.08 | 288 | 287.1 | 47.80 | **289** | **289** | 0.74 | **289** | **289** | 0.56 | **289** | **289** | 2.90 |
| scpa4 | 283 | 300 | 264 | 0.084 | 287 | 286.8 | 26.50 | **288** | **288** | 0.46 | **288** | **288** | 0.12 | **288** | **288** | 2.92 |
| scpa5 | 286 | 300 | 266 | 0.097 | 289 | 288.4 | 11.02 | **290** | **290** | 0.87 | **290** | **290** | 0.68 | **290** | **290** | 2.96 |
| scpb1 | 288 | 300 | 279 | 0.0167 | **296** | 295 | 35.59 | **296** | **296** | 1.63 | **296** | **296** | 2.02 | **296** | **296** | 5.70 |
| scpb2 | 286 | 300 | 276 | 0.208 | **295** | 294.3 | 55.29 | **295** | **295** | 0.66 | **295** | **295** | 0.47 | **295** | **295** | 5.39 |
| scpb3 | 287 | 300 | 274 | 0.207 | 295 | 293.5 | 85.89 | **296** | **296** | 5.97 | **296** | **296** | 27.12 | **296** | **296** | 6.66 |
| scpb4 | 286 | 300 | 281 | 0.127 | 294 | 293 | 16.97 | **295** | 294.6 | 27.62 | **295** | 294.3 | 1.48 | **295** | **295** | 14.67 |
| scpb5 | 285 | 300 | 278 | 0.112 | **295** | 294.2 | 43.28 | **295** | **295** | 1.61 | **295** | **295** | 3.07 | **295** | **295** | 5.84 |
| scpc1 | 384 | 400 | 364 | 0.124 | 388 | 387.3 | 60.23 | **391** | **391** | 4.25 | **391** | **391** | 5.31 | **391** | **391** | 4.21 |
| scpc2 | 380 | 400 | 365 | 0.181 | 388 | 387 | 46.00 | **391** | **391** | 7.24 | **391** | **391** | 12.56 | **391** | **391** | 7.54 |
| scpc3 | 384 | 400 | 365 | 0.148 | 388 | 386.9 | 86.52 | **391** | 390.7 | 19.48 | **391** | **390.8** | 17.73 | **391** | **390.8** | 14.84 |
| scpc4 | 384 | 400 | 363 | 0.126 | 388 | 387.4 | 49.40 | **392** | **392** | 2.03 | **392** | **392** | 5.92 | **392** | **392** | 6.11 |
| scpc5 | 383 | 400 | 364 | 0.157 | 387 | 386.5 | 49.30 | **391** | 390.7 | 15.71 | **391** | **391** | 12.21 | **391** | 390.9 | 17.40 |
| scpd1 | 376 | 400 | 377 | 0.332 | 391 | 390.9 | 43.10 | **395** | **394.3** | 26.93 | **395** | 394 | 25.32 | **395** | 394.2 | 26.51 |
| scpd2 | 377 | 400 | 365 | 0.367 | 392 | 390.8 | 57.69 | **393** | 392.6 | 27.24 | **393** | 392.7 | 46.20 | **393** | **393** | 23.95 |
| scpd3 | 378 | 400 | 373 | 0.364 | 391 | 389.6 | 50.64 | **393** | 392.8 | 17.79 | **393** | 392.6 | 19.88 | **393** | **393** | 17.13 |
| scpd4 | 379 | 400 | 368 | 0.391 | 390 | 389 | 41.95 | **393** | 392.1 | 43.38 | **393** | 392.1 | 35.59 | **393** | **392.8** | 40.32 |
| scpd5 | 365 | 400 | 363 | 0.323 | 390 | 389.3 | 45.75 | **393** | 392.3 | 32.55 | **393** | 392 | 32.79 | **393** | **392.7** | 28.34 |
| scpnre1 | 438 | 500 | 464 | 1.463 | 485 | 484.7 | 29.58 | **488** | 487.2 | 27.11 | **488** | 487 | 39.72 | **488** | **487.9** | 59.88 |
| scpnre2 | 447 | 500 | 471 | 1.312 | 487 | 485.1 | 59.02 | **488** | 487.2 | 73.58 | **488** | 487 | 37.52 | **488** | **487.3** | 47.97 |
| scpnre3 | 455 | 500 | 469 | 1.457 | 487 | 484.7 | 31.58 | **488** | 487.1 | 56.37 | 487 | 487 | 33.14 | **488** | **487.3** | 51.10 |
| scpnre4 | 456 | 500 | 473 | 1.545 | 487 | 485.6 | 46.50 | **488** | 487 | 45.47 | **488** | 486.8 | 38.35 | **488** | **487.4** | 55.57 |
| scpnre5 | 451 | 500 | 470 | 1.48 | 487 | 485.1 | 46.92 | **488** | 487.1 | 73.79 | 487 | 486.9 | 32.04 | **488** | **487.3** | 59.87 |
| scpnrf1 | 475 | 500 | 483 | 2.699 | 496 | 495 | 44.58 | 497 | 493.6 | 68.28 | **498** | 495.2 | 36.96 | 496 | **496** | 62.04 |
| scpnrf2 | 475 | 500 | 489 | 2.544 | 495 | 494.5 | 61.19 | **497** | 494.7 | 20.00 | **497** | 494.7 | 38.17 | **497** | **496.2** | 47.21 |
| scpnrf3 | 474 | 500 | 476 | 2.501 | 495 | 494.4 | 42.94 | 497 | 494 | 57.03 | **498** | 494.9 | 29.79 | 497 | **496.1** | 31.78 |
| scpnrf4 | 482 | 500 | 484 | 2.351 | 495 | 494.5 | 69.25 | **497** | 494.3 | 23.68 | 496 | 494.6 | 52.12 | **497** | **496.4** | 28.00 |
| scpnrf5 | 487 | 500 | 481 | 2.705 | 496 | 494.7 | 85.03 | **497** | 494.6 | 82.66 | 496 | 494.5 | 40.12 | **497** | **496** | 24.53 |
| scpnrg1 | 952 | 1000 | 928 | 2.441 | 975 | 970.2 | 80.48 | **984** | 981.9 | 64.60 | **984** | **982.2** | 35.76 | **984** | 981.8 | 38.04 |
| scpnrg2 | 768 | 1000 | 935 | 2.666 | 971 | 969.1 | 63.58 | **984** | 982.1 | 56.88 | **984** | **982.6** | 56.02 | **984** | 982.5 | 49.96 |
| scpnrg3 | 948 | 1000 | 927 | 3.146 | 971 | 968.9 | 9.44 | 981 | 979.2 | 56.11 | 981 | 979.7 | 39.59 | **983** | **981.1** | 37.29 |
| scpnrg4 | 947 | 1000 | 925 | 2.514 | 971 | 968.6 | 57.61 | 981 | 978.7 | 39.72 | **982** | 979.9 | 32.03 | **982** | **980.9** | 45.87 |
| scpnrg5 | 945 | 1000 | 923 | 2.569 | 970 | 968.8 | 66.31 | 981 | 979.4 | 76.23 | 981 | 979.2 | 33.45 | **983** | **980.9** | 46.82 |
| scpnrh1 | 836 | 1000 | 937 | 4.984 | 982 | 981 | 50.89 | 989 | 986.9 | 43.95 | **991** | 988.2 | 46.50 | **991** | **989.8** | 77.69 |
| scpnrh2 | 844 | 1000 | 940 | 4.693 | 983 | 981.1 | 67.75 | 988 | 986.3 | 20.31 | 989 | 988.3 | 69.30 | **992** | **989.8** | 68.81 |
| scpnrh3 | 849 | 1000 | 934 | 4.785 | 983 | 981 | 14.91 | 991 | 987.9 | 51.38 | 990 | 988.4 | 47.81 | **993** | **990.3** | 82.14 |
| scpnrh4 | 931 | 1000 | 936 | 4.612 | 982 | 981 | 56.15 | 989 | 987.4 | 29.34 | 990 | 988.8 | 74.88 | **992** | **989.9** | 71.99 |
| scpnrh5 | 846 | 1000 | 937 | 4.899 | 983 | 981.5 | 77.72 | 990 | 987.6 | 91.53 | 990 | 988.2 | 19.68 | **992** | **989.9** | 73.04 |
| scpclr10 | 504 | 506.94 | 479 | 0.095 | **504** | **504** | 0.80 | **504** | **504** | 0.02 | **504** | **504** | 0.00 | **504** | **504** | 2.97 |
| scpclr11 | 1011 | 1023 | 940 | 0.274 | **1011** | **1011** | 26.03 | **1011** | **1011** | 0.05 | **1011** | **1011** | 0.49 | **1011** | **1011** | 5.55 |
| scpclr12 | 1967 | 2047 | 1632 | 22.804 | 2021 | 2020.7 | 30.51 | **2023** | 2022.6 | 34.99 | **2023** | **2023** | 24.67 | **2023** | 2022.8 | 22.41 |
| scpclr13 | 3839 | 4095 | 3388 | 17.764 | **4045** | 4043.8 | 29.41 | **4045** | **4045** | 2.11 | **4045** | **4045** | 13.00 | **4045** | **4045** | 45.99 |

(*continued on next page*)

proposed algorithm performs well on the "scpnrg" and "scpnrh" test suits. The results on $k_{95}$ instances are given in Table 3. The contrasting experiments show that MMAS-ML outperforms ABPSO on 14 instances and achieves the same performance for the remaining instances. As the coverage ratio of the problem increases, the superiority of MMAS-ML becomes more distinct.

**Table 2** (*continued*).

| Eg | CPLEX | | GOPS | | RNKC | | | ABPSO | | | HBABC | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LB | UB | BEST | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scpcyc06 | 226 | 239.4 | 214 | 0.003 | **227** | **227** | 4.96 | **227** | **227** | 0.10 | **227** | **227** | 0.00 | **227** | **227** | 1.71 |
| scpcyc07 | 639 | 672 | 593 | 0.019 | 634 | 632.8 | 66.42 | **642** | **642** | 12.24 | **642** | **642** | 0.01 | **642** | **642** | 0.49 |
| scpcyc08 | 1661 | 1792 | 1567 | 0.276 | 1677 | 1674.3 | 86.19 | 1700 | 1692 | 95.82 | **1711** | **1704.3** | 31.42 | **1711** | 1697.7 | 48.65 |
| scpcyc09 | 4016 | 4608 | 3947 | 2.241 | 4239 | 4235.3 | 80.81 | 4317 | 4316.4 | 24.44 | **4349** | **4348.2** | 60.52 | 4318 | 4316.8 | 1.54 |
| scpcyc10 | 9185 | 11 520 | 9918 | 9.308 | 10 702 | 10 687.2 | 13.48 | **11 004** | **10 980.8** | 82.93 | 10 978 | 10 967.3 | 53.38 | 10 980 | 10 958.8 | 50.23 |
| scpcyc11 | 13 287 | 28 160 | 23 986 | 38.85 | 25 913 | 25 900.8 | 70.09 | 26 542 | 26 497.5 | 96.60 | 26 583 | **26 578.1** | 68.94 | **26 649** | 26 569 | 63.37 |
| Avg | 817 | 1088.09 | 954.93 | 2.04 | 1031 | 1029.31 | 38.17 | 1046 | 1044.81 | 22.46 | 1047 | 1046.42 | 18.11 | 1048 | 1045.99 | 21.2 |

**Table 3**
Results of MMAS-ML, GOPS, RNKC, ABPSO, HBABC and CPLEX on instances with $k_{95}$.

| Eg | CPLEX | | GOPS | | RNKC | | | ABPSO | | | HBABC | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LB | UB | BEST | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scp41 | 196 | 200 | 177 | 0.01 | **197** | 196.1 | 19.85 | **197** | **197** | 0.75 | **197** | **197** | 0.40 | **197** | **197** | 1.00 |
| scp42 | **198** | 198 | 183 | 0.009 | 197 | 197 | 6.17 | **198** | **198** | 0.32 | **198** | **198** | 0.14 | **198** | **198** | 0.99 |
| scp43 | 197 | 200 | 182 | 0.01 | 197 | 197 | 2.30 | **198** | **198** | 1.93 | **198** | **198** | 0.15 | **198** | **198** | 2.25 |
| scp44 | 194 | 199.89 | 183 | 0.01 | 195 | 194.4 | 23.33 | **196** | **196** | 0.21 | **196** | **196** | 0.12 | **196** | **196** | 1.04 |
| scp45 | 196 | 200 | 183 | 0.009 | **197** | 196.2 | 5.89 | **197** | **197** | 0.38 | **197** | **197** | 0.24 | **197** | **197** | 1.06 |
| scp46 | **196** | 199.11 | 182 | 0.011 | **196** | 195.6 | 30.49 | **196** | **196** | 0.10 | **196** | **196** | 0.02 | **196** | **196** | 0.96 |
| scp47 | **196** | 198.38 | 180 | 0.009 | **196** | 195.9 | 29.79 | **196** | **196** | 0.33 | **196** | **196** | 0.02 | **196** | **196** | 0.97 |
| scp48 | **196** | 199.89 | 182 | 0.01 | **196** | 195.1 | 21.60 | **196** | **196** | 0.23 | **196** | **196** | 0.13 | **196** | **196** | 1.02 |
| scp49 | **196** | 199.54 | 187 | 0.009 | **196** | 195.7 | 38.70 | **196** | **196** | 0.15 | **196** | **196** | 0.00 | **196** | **196** | 0.93 |
| scp410 | **196** | 198.62 | 184 | 0.01 | **196** | 195.2 | 48.06 | **196** | **196** | 0.17 | **196** | **196** | 0.06 | **196** | **196** | 0.99 |
| scp51 | 194 | 200 | 184 | 0.019 | **196** | 194.3 | 4.43 | **196** | **196** | 5.36 | **196** | **196** | 4.63 | **196** | **196** | 2.78 |
| scp52 | 195 | 200 | 184 | 0.025 | **196** | 195.1 | 19.71 | **196** | **196** | 0.36 | **196** | **196** | 0.65 | **196** | **196** | 1.85 |
| scp53 | 194 | 200 | 183 | 0.024 | **196** | **196** | 33.66 | **196** | **196** | 0.15 | **196** | **196** | 0.00 | **196** | **196** | 1.47 |
| scp54 | 195 | 200 | 182 | 0.021 | **196** | 195.1 | 2.55 | **196** | **196** | 0.29 | **196** | **196** | 0.05 | **196** | **196** | 1.54 |
| scp55 | 196 | 200 | 183 | 0.025 | 196 | 196 | 12.76 | **197** | **197** | 15.50 | **197** | **197** | 37.04 | **197** | 196.7 | 16.97 |
| scp56 | 195 | 200 | 181 | 0.027 | **196** | 195.3 | 82.48 | **196** | **196** | 0.36 | **196** | **196** | 0.31 | **196** | **196** | 1.59 |
| scp57 | 196 | 200 | 185 | 0.022 | **197** | 196.1 | 32.36 | **197** | **197** | 0.44 | **197** | **197** | 0.28 | **197** | **197** | 1.83 |
| scp58 | 193 | 200 | 180 | 0.023 | **196** | 194.6 | 36.97 | **196** | **196** | 0.51 | **196** | **196** | 0.61 | **196** | **196** | 1.65 |
| scp59 | 193 | 200 | 184 | 0.029 | **197** | 196.8 | 34.38 | **197** | **197** | 0.52 | **197** | **197** | 0.23 | **197** | **197** | 1.81 |
| scp510 | 194 | 200 | 180 | 0.027 | 195 | 194.6 | 47.99 | **196** | **196** | 0.25 | **196** | **196** | 0.39 | **196** | **196** | 1.74 |
| scp61 | 197 | 200 | 185 | 0.029 | **199** | **199** | 3.97 | **199** | **199** | 0.28 | **199** | **199** | 0.41 | **199** | **199** | 1.63 |
| scp62 | 194 | 200 | 180 | 0.031 | **198** | 197.1 | 39.46 | **198** | **198** | 0.91 | **198** | **198** | 2.02 | **198** | **198** | 1.92 |
| scp63 | 196 | 200 | 191 | 0.033 | **199** | 198.9 | 47.03 | **199** | **199** | 0.48 | **199** | **199** | 0.73 | **199** | **199** | 1.65 |
| scp64 | 192 | 200 | 178 | 0.037 | **196** | **196** | 16.62 | **196** | **196** | 0.36 | **196** | **196** | 0.99 | **196** | **196** | 1.76 |
| scp65 | 196 | 200 | 182 | 0.035 | **198** | **198** | 10.44 | **198** | **198** | 0.11 | **198** | **198** | 0.14 | **198** | **198** | 1.50 |
| scpa1 | 288 | 300 | 269 | 0.103 | 293 | 292.2 | 37.11 | **295** | 294.8 | 23.22 | **295** | 294.9 | 18.53 | **295** | **295** | 3.78 |
| scpa2 | 288 | 300 | 271 | 0.097 | 293 | 292.2 | 35.71 | **295** | **295** | 7.04 | **295** | **295** | 7.43 | **295** | **295** | 3.14 |
| scpa3 | 290 | 300 | 278 | 0.089 | 294 | 292.8 | 75.27 | **295** | **295** | 0.63 | **295** | **295** | 0.66 | **295** | **295** | 3.01 |
| scpa4 | 292 | 300 | 269 | 0.081 | 293 | 292.3 | 30.48 | **294** | **294** | 0.62 | **294** | **294** | 0.20 | **294** | **294** | 3.03 |
| scpa5 | 290 | 300 | 272 | 0.08 | 294 | 293.6 | 28.24 | **296** | **296** | 5.68 | **296** | **296** | 6.57 | **296** | **296** | 6.98 |
| scpb1 | 287 | 300 | 283 | 0.232 | **299** | 297.6 | 51.17 | **299** | **299** | 2.28 | **299** | **299** | 3.72 | **299** | **299** | 6.11 |
| scpb2 | 293 | 300 | 279 | 0.178 | 298 | 297.5 | 37.86 | **299** | **299** | 14.76 | **299** | 298.7 | 30.15 | **299** | **299** | 8.35 |
| scpb3 | 290 | 300 | 278 | 0.265 | 297 | 296.5 | 44.63 | **299** | **299** | 9.66 | **299** | 298.7 | 30.29 | **299** | **299** | 9.51 |
| scpb4 | 287 | 300 | 285 | 0.236 | 298 | 296.6 | 40.09 | **299** | 298.2 | 25.12 | **299** | 298.1 | 17.86 | **299** | **298.9** | 32.78 |
| scpb5 | 285 | 300 | 282 | 0.212 | **298** | 297.1 | 63.72 | **298** | **298** | 1.55 | **298** | **298** | 6.65 | **298** | **298** | 5.58 |
| scpc1 | 387 | 400 | 371 | 0.235 | 394 | 392.6 | 89.07 | **397** | **396.9** | 36.58 | **397** | 396.8 | 34.11 | **397** | 396.8 | 14.00 |
| scpc2 | 388 | 400 | 371 | 0.208 | 394 | 392.8 | 28.02 | **397** | 396.9 | 40.79 | **397** | 396.7 | 30.18 | **397** | **397** | 10.32 |
| scpc3 | 389 | 400 | 370 | 0.271 | 394 | 392.6 | 39.91 | **397** | 396.3 | 27.31 | **397** | 396.6 | 21.21 | **397** | **396.7** | 22.82 |
| scpc4 | 387 | 400 | 371 | 0.237 | 394 | 393 | 67.58 | **397** | **397** | 2.36 | **397** | **397** | 5.59 | **397** | **397** | 4.31 |
| scpc5 | 391 | 400 | 370 | 0.234 | 394 | 392.4 | 96.28 | **397** | 396.3 | 30.44 | **397** | **396.4** | 19.61 | **397** | 396.3 | 15.44 |
| scpd1 | 382 | 400 | 381 | 0.428 | 396 | 395.1 | 81.44 | **398** | **398** | 11.93 | **398** | 397.7 | 49.28 | **398** | **398** | 12.04 |
| scpd2 | 385 | 400 | 371 | 0.47 | 395 | 394.4 | 47.45 | **397** | 396.5 | 55.72 | **397** | 396.3 | 15.71 | **397** | **397** | 18.93 |
| scpd3 | 385 | 400 | 378 | 0.489 | 394 | 393.6 | 36.10 | **397** | 396.3 | 21.93 | **397** | 396.2 | 36.56 | **397** | **396.7** | 30.18 |
| scpd4 | 381 | 400 | 374 | 0.347 | 394 | 393.1 | 57.45 | **397** | 396.2 | 43.60 | **397** | 395.9 | 53.31 | **397** | **396.6** | 33.35 |
| scpd5 | 386 | 400 | 369 | 0.494 | 394 | 393.3 | 49.70 | **397** | 396.3 | 52.01 | **397** | 396 | 58.60 | **397** | **396.5** | 45.79 |
| scpnre1 | 470 | 500 | 475 | 1.494 | 493 | 491.5 | 41.44 | **495** | 494 | 22.09 | **495** | 493.3 | 37.14 | **495** | **494.4** | 52.94 |
| scpnre2 | 470 | 500 | 477 | 1.554 | 492 | 491.7 | 58.88 | **495** | 494 | 16.83 | **495** | 493.3 | 44.36 | **495** | **494.3** | 54.62 |
| scpnre3 | 449 | 500 | 477 | 1.422 | 492 | 491.5 | 48.41 | **495** | 493.7 | 22.65 | 494 | 493.6 | 28.07 | **495** | **494.6** | 63.06 |
| scpnre4 | 469 | 500 | 479 | 1.892 | 492 | 491.2 | 39.47 | **495** | 494.1 | 52.95 | 494 | 493.1 | 34.44 | **495** | **494.6** | 67.85 |
| scpnre5 | 469 | 500 | 478 | 1.228 | 492 | 491.3 | 60.58 | **495** | 493.7 | 70.08 | **495** | 493.4 | 60.69 | **495** | **494.3** | 49.46 |
| scpnrf1 | 488 | 500 | 492 | 3.777 | **500** | 499.3 | 26.71 | **500** | 499.3 | 23.56 | **500** | 499.6 | 56.39 | **500** | **500** | 25.53 |
| scpnrf2 | 491 | 500 | 495 | 3.81 | **500** | 499.6 | 39.27 | **500** | 499.7 | 26.14 | **500** | 499.2 | 31.13 | **500** | **500** | 38.72 |
| scpnrf3 | 488 | 500 | 484 | 3.788 | **500** | 499.4 | 40.80 | **500** | 499.8 | 24.76 | **500** | 498.9 | 65.39 | **500** | **500** | 57.46 |
| scpnrf4 | 483 | 500 | 492 | 2.764 | **500** | 499.4 | 26.81 | **500** | 499.9 | 21.71 | **500** | 499.2 | 46.56 | **500** | **500** | 41.56 |
| scpnrf5 | 496 | 500 | 489 | 3.241 | **500** | 499.8 | 37.14 | **500** | 499.5 | 17.13 | **500** | 499 | 39.96 | **500** | **500** | 51.79 |
| scpnrg1 | 964 | 1000 | 942 | 3.998 | 984 | 981.9 | 68.28 | 994 | 992.5 | 39.83 | **995** | **993** | 62.52 | **995** | 992.8 | 27.29 |
| scpnrg2 | 960 | 1000 | 946 | 3.471 | 982 | 981.2 | 57.50 | 993 | 992.4 | 62.45 | **995** | **993.4** | 52.23 | **995** | 992.9 | 36.86 |

(*continued on next page*)

Moreover, compared with ABPSO on the AVG metric, MMAS-ML still confirms the robustness of the performance.

As for the comparison with HBABC, MMAS-ML outperforms on 12 $k_{90}$ instances and 12 $k_{95}$ instances, respectively (see Tables 2

**Table 3** (*continued*).

| Eg | CPLEX | | GOPS | | RNKC | | | ABPSO | | | HBABC | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LB | UB | BEST | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scpnrg3 | 949 | 1000 | 942 | 3.067 | 983 | 981.6 | 58.95 | 992 | 990.3 | 10.05 | **993** | 991.6 | 61.89 | **993** | **991.7** | 39.88 |
| scpnrg4 | 959 | 1000 | 934 | 2.583 | 983 | 981.4 | 43.25 | 991 | 990.1 | 30.78 | **995** | **991.6** | 50.17 | 994 | 991.4 | 35.13 |
| scpnrg5 | 836 | 1000 | 937 | 3.048 | 982 | 981.3 | 23.58 | 992 | 990.1 | 21.98 | 992 | 991.3 | 42.44 | **993** | **991.4** | 45.29 |
| scpnrh1 | 843 | 1000 | 944 | 7.425 | 987 | 986.3 | 30.34 | 993 | 991.2 | 4.60 | 995 | 993.3 | 79.80 | **996** | **994.4** | 59.51 |
| scpnrh2 | 851 | 1000 | 946 | 7.941 | 988 | 986.5 | 31.98 | 993 | 991.2 | 2.65 | 994 | 992.3 | 42.72 | **995** | **994.1** | 52.87 |
| scpnrh3 | 856 | 1000 | 941 | 7.881 | 988 | 986.6 | 18.55 | 994 | 991.8 | 16.16 | 995 | 993.1 | 49.91 | **996** | **994** | 59.77 |
| scpnrh4 | 863 | 1000 | 942 | 6.752 | 989 | 987.3 | 75.73 | 994 | 991.3 | 8.21 | 994 | 993 | 78.73 | **996** | **994.5** | 44.86 |
| scpnrh5 | 859 | 1000 | 943 | 7.385 | 988 | 986.4 | 12.13 | 993 | 991.4 | 1.92 | 994 | 992.8 | 29.57 | **995** | **994.3** | 43.74 |
| scpclr10 | **509** | 510 | 484 | 0.134 | **509** | **509** | 2.41 | **509** | **509** | 0.06 | **509** | **509** | 0.00 | **509** | **509** | 2.84 |
| scpclr11 | **1015** | 1223 | 948 | 0.614 | **1015** | 1014.8 | 45.88 | **1015** | **1015** | 0.03 | **1015** | **1015** | 0.00 | **1015** | **1015** | 6.78 |
| scpclr12 | 1978 | 2047 | 1638 | 19.315 | 2027 | 2026.3 | 57.56 | **2031** | **2031** | 0.87 | **2031** | **2031** | 4.59 | **2031** | **2031** | 18.07 |
| scpclr13 | 3886 | 4095 | 3443 | 17.526 | 4055 | 4053.7 | 47.23 | **4063** | 4058.2 | 32.62 | 4057 | 4057 | 50.54 | **4063** | **4059.3** | 66.80 |
| scpcyc06 | 234 | 240 | 220 | 0.004 | **235** | **235** | 16.99 | **235** | **235** | 0.10 | **235** | **235** | 0.00 | **235** | **235** | 0.49 |
| scpcyc07 | 649 | 672 | 605 | 0.016 | 649 | 647 | 38.14 | **658** | **658** | 5.39 | **658** | **658** | 1.34 | **658** | **658** | 2.50 |
| scpcyc08 | 1720 | 1792 | 1601 | 0.265 | 1715 | 1712.8 | 16.73 | 1744 | 1740.2 | 42.04 | **1752** | **1747.5** | 23.47 | **1752** | 1744 | 15.31 |
| scpcyc09 | 4022 | 4608 | 4042 | 1.511 | 4353 | 4346.2 | 27.26 | 4444 | 4421.5 | 9.14 | 4461 | **4455** | 18.58 | **4468** | 4454.1 | 45.56 |
| scpcyc10 | 9626 | 11 520 | 10 151 | 8.791 | 10 956 | 10 945.3 | 49.21 | 11 256 | 11 170.9 | 70.15 | 11 265 | 11 249.5 | 48.28 | **11 292** | **11 265** | 69.73 |
| scpcyc11 | 13 893 | 28 160 | 24 563 | 41.837 | 26 546 | 26 524.7 | 78.10 | 27 267 | 27 245.9 | 85.53 | 27 314 | **27 302** | 52.54 | **27 315** | 27 243.9 | 61.05 |
| Avg | 838.36 | 1091.47 | 973.43 | 2.32 | 1049.19 | 1047.9 | 38.77 | 1066.47 | 1064.25 | 15.81 | 1067.61 | 1066.66 | 22.53 | 1068.27 | 1066.33 | 21.06 |

**Table 4**
Comparative results of MMAS-ML, RWLS and uSLC on $k_{90}$ instances.

| Eg | K90 | RWLS | | | uSLC | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scp41 | 34 | 186 | 186 | 0.021 | 186 | 186 | 0.02 | **193** | **193** | 2.58 |
| scp42 | 33 | 186 | 186 | 0.02 | 186 | 186 | 0.02 | **193** | **193** | 0.93 |
| scp43 | 34 | 186 | 186 | 0.021 | 186 | 186 | 0.02 | **194** | **194** | 1.09 |
| scp44 | 34 | 184 | 184 | 0.02 | 184 | 184 | 0.02 | **191** | **191** | 0.96 |
| scp45 | 34 | 185 | 185 | 0.02 | 185 | 185 | 0.02 | **193** | **193** | 1.38 |
| scp46 | 33 | 181 | 181 | 0.02 | 181 | 181 | 0.02 | **191** | **191** | 0.95 |
| scp47 | 34 | 186 | 186 | 0.02 | 186 | 186 | 0.02 | **191** | **191** | 1.01 |
| scp48 | 33 | 186 | 186 | 0.021 | 186 | 186 | 0.02 | **192** | **192** | 1.12 |
| scp49 | 34 | 186 | 186 | 0.02 | 186 | 186 | 0.021 | **192** | **192** | 0.98 |
| scp410 | 34 | 186 | 186 | 0.02 | 186 | 186 | 0.02 | **192** | **192** | 2.70 |
| scp51 | 31 | 187 | 187 | 0.02 | 187 | 187 | 0.02 | **193** | **193** | 1.58 |
| scp52 | 31 | 188 | 188 | 0.021 | 188 | 188 | 0.02 | **194** | **194** | 1.99 |
| scp53 | 31 | 189 | 189 | 0.02 | 189 | 189 | 0.02 | **194** | **194** | 1.54 |
| scp54 | 31 | 188 | 188 | 0.021 | 188 | 188 | 0.02 | **194** | **194** | 1.77 |
| scp55 | 31 | 189 | 189 | 0.02 | 189 | 189 | 0.02 | **194** | **194** | 1.63 |
| scp56 | 31 | 185 | 185 | 0.02 | 185 | 185 | 0.02 | **194** | **194** | 1.74 |
| scp57 | 31 | 185 | 185 | 0.02 | 185 | 185 | 0.02 | **195** | **195** | 1.92 |
| scp58 | 31 | 183 | 183 | 0.02 | 183 | 183 | 0.02 | **194** | **194** | 1.73 |
| scp59 | 32 | 190 | 190 | 0.021 | 190 | 190 | 0.021 | **194** | **194** | 1.19 |
| scp510 | 31 | 183 | 183 | 0.02 | 183 | 183 | 0.02 | **194** | **194** | 2.94 |
| scp61 | 19 | 190 | 190 | 0.02 | 190 | 190 | 0.02 | **196** | **196** | 1.56 |
| scp62 | 18 | 191 | 191 | 0.021 | 191 | 191 | 0.02 | **195** | **195** | 1.87 |
| scp63 | 19 | 190 | 190 | 0.02 | 190 | 190 | 0.02 | **196** | **196** | 1.35 |
| scp64 | 18 | 191 | 191 | 0.02 | 191 | 191 | 0.02 | **192** | **192** | 1.30 |
| scp65 | 19 | 192 | 192 | 0.02 | 192 | 192 | 0.02 | **196** | **196** | 1.77 |
| scpa1 | 34 | 280 | 280 | 0.02 | 280 | 280 | 0.022 | **289** | **289** | 7.47 |
| scpa2 | 34 | 281 | 281 | 0.021 | 281 | 281 | 0.02 | **289** | **289** | 7.19 |
| scpa3 | 34 | 280 | 280 | 0.02 | 280 | 280 | 0.022 | **289** | **289** | 2.90 |
| scpa4 | 33 | 282 | 282 | 0.021 | 282 | 282 | 0.02 | **288** | **288** | 2.92 |
| scpa5 | 34 | 281 | 281 | 0.022 | 281 | 281 | 0.021 | **290** | **290** | 2.96 |
| scpb1 | 20 | 287 | 287 | 0.028 | 287 | 287 | 0.027 | **296** | **296** | 5.70 |
| scpb2 | 20 | 294 | 294 | 0.027 | 294 | 294 | 0.029 | **295** | **295** | 5.39 |
| scpb3 | 20 | 292 | 292 | 0.027 | 292 | 292 | 0.026 | **296** | **296** | 6.66 |
| scpb4 | 20 | 289 | 289 | 0.029 | 289 | 289 | 0.028 | **295** | **295** | 14.67 |
| scpb5 | 20 | 287 | 287 | 0.028 | 287 | 287 | 0.027 | **295** | **295** | 5.84 |
| scpc1 | 39 | 381 | 381 | 0.024 | 381 | 381 | 0.026 | **391** | **391** | 4.21 |
| scpc2 | 39 | 383 | 383 | 0.026 | 383 | 383 | 0.026 | **391** | **391** | 7.54 |
| scpc3 | 39 | 382 | 382 | 0.028 | 382 | 382 | 0.023 | **391** | **390.8** | 14.84 |
| scpc4 | 39 | 380 | 380 | 0.028 | 380 | 380 | 0.024 | **392** | **392** | 6.11 |
| scpc5 | 39 | 381 | 381 | 0.024 | 381 | 381 | 0.026 | **391** | **390.9** | 17.40 |
| scpd1 | 22 | 385 | 385 | 0.039 | 385 | 385 | 0.039 | **395** | **394.2** | 26.51 |
| scpd2 | 22 | 387 | 387 | 0.039 | 387 | 387 | 0.036 | **393** | **393** | 23.95 |
| scpd3 | 22 | 386 | 386 | 0.038 | 386 | 386 | 0.04 | **393** | **393** | 17.13 |
| scpd4 | 22 | 386 | 386 | 0.039 | 386 | 386 | 0.038 | **393** | **392.8** | 40.32 |
| scpd5 | 22 | 384 | 384 | 0.039 | 384 | 384 | 0.039 | **393** | **392.7** | 28.34 |
| scpnre1 | 14 | 480 | 480 | 0.078 | 480 | 480 | 0.078 | **488** | **487.9** | 59.88 |

(*continued on next page*)

**Table 4** (*continued*).

| Eg | K90 | RWLS | | | uSLC | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scpnre2 | 14 | 483 | 483 | 0.076 | 483 | 483 | 0.08 | **488** | **487.3** | 47.97 |
| scpnre3 | 14 | 483 | 483 | 0.077 | 483 | 483 | 0.076 | **488** | **487.3** | 51.10 |
| scpnre4 | 14 | 483 | 483 | 0.076 | 483 | 483 | 0.077 | **488** | **487.4** | 55.57 |
| scpnre5 | 14 | 483 | 483 | 0.078 | 483 | 483 | 0.075 | **488** | **487.3** | 59.87 |
| scpnrf1 | 9 | 491 | 491 | 0.137 | 491 | 491 | 0.139 | **496** | **496** | 62.04 |
| scpnrf2 | 9 | 491 | 491 | 0.134 | 491 | 491 | 0.137 | **497** | **496.2** | 47.21 |
| scpnrf3 | 9 | 491 | 491 | 0.136 | 491 | 491 | 0.137 | **497** | **496.1** | 31.78 |
| scpnrf4 | 9 | 491 | 491 | 0.137 | 491 | 491 | 0.137 | **497** | **496.4** | 28.00 |
| scpnrf5 | 9 | 491 | 491 | 0.137 | 491 | 491 | 0.139 | **497** | **496** | 24.53 |
| scpnrg1 | 55 | 967 | 967 | 0.076 | 967 | 967 | 0.075 | **984** | **981.8** | 38.04 |
| scpnrg2 | 55 | 970 | 970 | 0.076 | 970 | 970 | 0.077 | **984** | **982.5** | 49.96 |
| scpnrg3 | 55 | 969 | 969 | 0.074 | 969 | 969 | 0.074 | **983** | **981.1** | 37.29 |
| scpnrg4 | 55 | 964 | 964 | 0.076 | 964 | 964 | 0.079 | **982** | **980.9** | 45.87 |
| scpnrg5 | 55 | 964 | 964 | 0.077 | 964 | 964 | 0.076 | **983** | **980.9** | 46.82 |
| scpnrh1 | 31 | 982 | 982 | 0.151 | 982 | 982 | 0.149 | **991** | **989.8** | 77.69 |
| scpnrh2 | 31 | 982 | 982 | 0.147 | 982 | 982 | 0.149 | **992** | **989.8** | 68.81 |
| scpnrh3 | 31 | 982 | 982 | 0.15 | 982 | 982 | 0.148 | **993** | **990.3** | 82.14 |
| scpnrh4 | 31 | 982 | 982 | 0.146 | 982 | 982 | 0.15 | **992** | **989.9** | 71.99 |
| scpnrh5 | 31 | 982 | 982 | 0.149 | 982 | 982 | 0.15 | **992** | **989.9** | 73.04 |
| scpclr10 | 23 | 494 | 494 | 0.02 | 494 | 494 | 0.02 | **504** | **504** | 2.97 |
| scpclr11 | 21 | 985 | 985 | 0.021 | 985 | 985 | 0.021 | **1011** | **1011** | 5.55 |
| scpclr12 | 21 | 1987 | 1987 | 0.047 | 1987 | 1987 | 0.045 | **2023** | **2022.8** | 22.41 |
| scpclr13 | 21 | 4001 | 4001 | 0.099 | 4001 | 4001 | 0.1 | **4045** | **4045** | 45.99 |
| scpcyc06 | 54 | 224 | 224 | 0.058 | 222 | 222 | 0.02 | **227** | **227** | 1.71 |
| scpcyc07 | 130 | 639 | 639 | 3.618 | 618 | 618 | 0.02 | **642** | **642** | 0.49 |
| scpcyc08 | 308 | 1689 | 1688.2 | 15.406 | 1624 | 1624 | 0.02 | **1711** | **1697.7** | 48.65 |
| scpcyc09 | 695 | 4299 | 4297.1 | 49.726 | 4245 | 4245 | 0.032 | **4318** | **4316.8** | 1.54 |
| scpcyc10 | 1618 | 10 900 | 10 895.1 | 76.027 | 10 590 | 10 590 | 0.062 | **10 980** | **10 959** | 50.23 |
| scpcyc11 | 3571 | 26 394 | 26 381.8 | 90.206 | 25 948 | 25 948 | 0.173 | **26 649** | **26 569** | 63.37 |
| Avg | | 1034.45 | 1034.19 | 3.18 | 1022.48 | 1022.48 | 0.05 | 1047.9 | 1045.99 | 21.20 |

**Table 5**
Comparative results of MMAS-ML, RWLS and uSLC on $k_{95}$ instances.

| Eg | K95 | RWLS | | | uSLC | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scp41 | 36 | 190 | 190 | 0.02 | 190 | 190 | 0.02 | **197** | **197** | 1.00 |
| scp42 | 35 | 190 | 190 | 0.02 | 190 | 190 | 0.021 | **198** | **198** | 0.99 |
| scp43 | 36 | 190 | 190 | 0.02 | 190 | 190 | 0.02 | **198** | **198** | 2.25 |
| scp44 | 36 | 188 | 188 | 0.02 | 188 | 188 | 0.02 | **196** | **196** | 1.04 |
| scp45 | 36 | 189 | 189 | 0.02 | 189 | 189 | 0.02 | **197** | **197** | 1.06 |
| scp46 | 35 | 186 | 186 | 0.02 | 186 | 186 | 0.02 | **196** | **196** | 0.96 |
| scp47 | 36 | 190 | 190 | 0.02 | 190 | 190 | 0.02 | **196** | **196** | 0.97 |
| scp48 | 35 | 190 | 190 | 0.02 | 190 | 190 | 0.02 | **196** | **196** | 1.02 |
| scp49 | 36 | 190 | 190 | 0.02 | 190 | 190 | 0.02 | **196** | **196** | 0.93 |
| scp410 | 36 | 190 | 190 | 0.02 | 190 | 190 | 0.021 | **196** | **196** | 0.99 |
| scp51 | 32 | 190 | 190 | 0.021 | 190 | 190 | 0.02 | **196** | **196** | 2.78 |
| scp52 | 32 | 190 | 190 | 0.021 | 190 | 190 | 0.02 | **196** | **196** | 1.85 |
| scp53 | 32 | 191 | 191 | 0.02 | 191 | 191 | 0.02 | **196** | **196** | 1.47 |
| scp54 | 32 | 190 | 190 | 0.02 | 190 | 190 | 0.021 | **196** | **196** | 1.54 |
| scp55 | 32 | 191 | 191 | 0.02 | 191 | 191 | 0.02 | **197** | **196.7** | 16.97 |
| scp56 | 32 | 187 | 187 | 0.02 | 187 | 187 | 0.02 | **196** | **196** | 1.59 |
| scp57 | 32 | 188 | 188 | 0.02 | 188 | 188 | 0.02 | **197** | **197** | 1.83 |
| scp58 | 32 | 186 | 186 | 0.02 | 186 | 186 | 0.021 | **196** | **196** | 1.65 |
| scp59 | 33 | 192 | 192 | 0.02 | 192 | 192 | 0.02 | **197** | **197** | 1.81 |
| scp510 | 32 | 186 | 186 | 0.02 | 186 | 186 | 0.02 | **196** | **196** | 1.74 |
| scp61 | 20 | 193 | 193 | 0.02 | 193 | 193 | 0.02 | **199** | **199** | 1.63 |
| scp62 | 19 | 194 | 194 | 0.02 | 194 | 194 | 0.02 | **198** | **198** | 1.92 |
| scp63 | 20 | 193 | 193 | 0.02 | 193 | 193 | 0.02 | **199** | **199** | 1.65 |
| scp64 | 19 | 195 | 195 | 0.02 | 195 | 195 | 0.02 | **196** | **196** | 1.76 |
| scp65 | 20 | 195 | 195 | 0.02 | 195 | 195 | 0.02 | **198** | **198** | 1.50 |
| scpa1 | 36 | 286 | 286 | 0.02 | 286 | 286 | 0.021 | **295** | **295** | 3.78 |
| scpa2 | 36 | 287 | 287 | 0.02 | 287 | 287 | 0.022 | **295** | **295** | 3.14 |
| scpa3 | 36 | 286 | 286 | 0.02 | 286 | 286 | 0.021 | **295** | **295** | 3.01 |
| scpa4 | 35 | 288 | 288 | 0.02 | 288 | 288 | 0.021 | **294** | **294** | 3.03 |
| scpa5 | 36 | 287 | 287 | 0.02 | 287 | 287 | 0.02 | **296** | **296** | 6.98 |

(*continued on next page*)

and 3). Note that MMAS-ML is inferior to HBABC in four instances, but it achieves better average fitness values in half of these instances (scpnrf1 and scpnrf3). It is also observed that MMAS-ML outperforms HBABC on 24 $k_{90}$ instances and 30 $k_{95}$ instances on the AVG metric.

To make the experimental results convincing, the Superior-Draw-Inferior (S-D-I) values of the results are calculated compared to RNKC, ABPSO and HBABC with the Mann–Whitney U test [36]. Superior, Draw, and Inferior indicate the number of instances the compared algorithms perform better, equivalent, or

**Table 5** (*continued*).

| Eg | K95 | RWLS | | | uSLC | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scpb1 | 21 | 292 | 292 | 0.029 | 292 | 292 | 0.027 | **299** | **299** | 6.11 |
| scpb2 | 21 | 297 | 297 | 0.028 | 297 | 297 | 0.028 | **299** | **299** | 8.35 |
| scpb3 | 21 | 296 | 296 | 0.027 | 296 | 296 | 0.029 | **299** | **299** | 9.51 |
| scpb4 | 21 | 293 | 293 | 0.027 | 293 | 293 | 0.026 | **299** | **298.9** | 32.78 |
| scpb5 | 21 | 291 | 291 | 0.029 | 291 | 291 | 0.028 | **298** | **298** | 5.58 |
| scpc1 | 41 | 389 | 389 | 0.024 | 389 | 389 | 0.026 | **397** | **396.8** | 14.00 |
| scpc2 | 41 | 389 | 389 | 0.024 | 389 | 389 | 0.024 | **397** | **397** | 10.32 |
| scpc3 | 41 | 388 | 388 | 0.027 | 388 | 388 | 0.025 | **397** | **396.7** | 22.82 |
| scpc4 | 41 | 388 | 388 | 0.026 | 388 | 388 | 0.025 | **397** | **397** | 4.31 |
| scpc5 | 41 | 387 | 387 | 0.024 | 387 | 387 | 0.025 | **397** | **396.3** | 15.44 |
| scpd1 | 23 | 390 | 390 | 0.039 | 390 | 390 | 0.04 | **398** | **398** | 12.04 |
| scpd2 | 23 | 391 | 391 | 0.039 | 391 | 391 | 0.038 | **397** | **397** | 18.93 |
| scpd3 | 23 | 390 | 390 | 0.038 | 390 | 390 | 0.038 | **397** | **396.7** | 30.18 |
| scpd4 | 23 | 391 | 391 | 0.039 | 391 | 391 | 0.039 | **397** | **396.6** | 33.35 |
| scpd5 | 23 | 389 | 389 | 0.038 | 389 | 389 | 0.04 | **397** | **396.5** | 45.79 |
| scpnre1 | 15 | 488 | 488 | 0.077 | 488 | 488 | 0.08 | **495** | **494.4** | 52.94 |
| scpnre2 | 15 | 489 | 489 | 0.079 | 489 | 489 | 0.08 | **495** | **494.3** | 54.62 |
| scpnre3 | 15 | 489 | 489 | 0.078 | 489 | 489 | 0.078 | **495** | **494.6** | 63.06 |
| scpnre4 | 15 | 489 | 489 | 0.078 | 489 | 489 | 0.077 | **495** | **494.6** | 67.85 |
| scpnre5 | 15 | 489 | 489 | 0.079 | 489 | 489 | 0.08 | **495** | **494.3** | 49.46 |
| scpnrf1 | 10 | 498 | 498 | 0.136 | 498 | 498 | 0.137 | **500** | **500** | 25.53 |
| scpnrf2 | 10 | 498 | 498 | 0.138 | 498 | 498 | 0.136 | **500** | **500** | 38.72 |
| scpnrf3 | 10 | 498 | 498 | 0.133 | 498 | 498 | 0.136 | **500** | **500** | 57.46 |
| scpnrf4 | 10 | 498 | 498 | 0.136 | 498 | 498 | 0.137 | **500** | **500** | 41.56 |
| scpnrf5 | 10 | 498 | 498 | 0.135 | 498 | 498 | 0.137 | **500** | **500** | 51.79 |
| scpnrg1 | 58 | 980 | 980 | 0.079 | 980 | 980 | 0.077 | **995** | **992.8** | 27.29 |
| scpnrg2 | 58 | 982 | 982 | 0.076 | 982 | 982 | 0.075 | **995** | **992.9** | 36.86 |
| scpnrg3 | 58 | 982 | 982 | 0.075 | 982 | 982 | 0.076 | **993** | **991.7** | 39.88 |
| scpnrg4 | 58 | 977 | 977 | 0.076 | 977 | 977 | 0.075 | **994** | **991.4** | 35.13 |
| scpnrg5 | 58 | 977 | 977 | 0.078 | 977 | 977 | 0.072 | **993** | **991.4** | 45.29 |
| scpnrh1 | 32 | 987 | 987 | 0.149 | 987 | 987 | 0.15 | **996** | **994.4** | 59.51 |
| scpnrh2 | 32 | 987 | 987 | 0.147 | 987 | 987 | 0.152 | **995** | **994.1** | 52.87 |
| scpnrh3 | 32 | 987 | 987 | 0.151 | 987 | 987 | 0.151 | **996** | **994** | 59.77 |
| scpnrh4 | 32 | 987 | 987 | 0.15 | 987 | 987 | 0.15 | **996** | **994.5** | 44.86 |
| scpnrh5 | 32 | 987 | 987 | 0.148 | 987 | 987 | 0.148 | **995** | **994.3** | 43.74 |
| scpclr10 | 24 | 497 | 497 | 0.02 | 497 | 497 | 0.02 | **509** | **509** | 2.84 |
| scpclr11 | 22 | 991 | 991 | 0.022 | 991 | 991 | 0.021 | **1015** | **1015** | 6.78 |
| scpclr12 | 22 | 2000 | 2000 | 0.044 | 2000 | 2000 | 0.046 | **2031** | **2031** | 18.07 |
| scpclr13 | 22 | 4018 | 4018 | 0.099 | 4018 | 4018 | 0.101 | **4063** | **4059.3** | 66.80 |
| scpcyc06 | 57 | 232 | 232 | 0.117 | 231 | 231 | 0.02 | **235** | **235** | 0.49 |
| scpcyc07 | 137 | 654 | 654 | 2.941 | 639 | 639 | 0.02 | **658** | **658** | 2.50 |
| scpcyc08 | 325 | 1740 | 1738.8 | 38.63 | 1675 | 1675 | 0.02 | **1752** | **1744** | 15.31 |
| scpcyc09 | 733 | 4458 | 4438.4 | 64.398 | 4359 | 4359 | 0.03 | **4468** | **4454.1** | 45.56 |
| scpcyc10 | 1708 | 11 193 | 11 175.3 | 80.528 | 10 860 | 10 860 | 0.064 | **11 292** | **11 265.4** | 69.73 |
| scpcyc11 | 3770 | 27 064 | 27 043 | 83.472 | 26 558 | 26 558 | 0.181 | **27 315** | **27 243.9** | 61.05 |
| Avg | | 1055.51 | 1054.71 | 3.65 | 1041.92 | 1041.92 | 0.05 | 1068 | 1066.33 | 21.06 |

**Table 6**
Summary of comparative results of MMAS-ML, RWLS, uSLC, RNKC, ABPSO and HBABC.

| Instances | MMAS-ML vs. | RWLS | uSLC | RNKC | ABPSO | HBABC |
|---|---|---|---|---|---|---|
| K90 | #Better | 75 | 75 | 50 | 11 | 12 |
| | #Draw | 0 | 0 | 25 | 62 | 60 |
| | #Worse | 0 | 0 | 0 | 2 | 3 |
| K95 | #Better | 75 | 75 | 45 | 14 | 12 |
| | #Draw | 0 | 0 | 30 | 61 | 62 |
| | #Worse | 0 | 0 | 0 | 0 | 1 |

worse than MMAS-ML. Finally, the S-D-I of RNKC was 0-55-95, while that of ABPSO was 2-123-25, and the S-D-I of HBABC was 4-122-24, which verifies the effectiveness of MMAS-ML.

### 6.4. Further comparison with MSCP's algorithms

In this subsection, MMAS-ML is further compared with two state-of-the-art algorithms for the minimum set cover problem, including uSLC [25] and RWLS [26]. By changing the constraint of the feasible solution, the MSCP algorithms above can be modified to solve MKCP. Specifically, the feasible constraint for MSCP is to cover all rows. Thus, the solution of the MSCP algorithm is acceptable when all rows are covered. For MKCP, only $k$ columns are allowed to cover as many rows as possible. Therefore, the solution of MKCP algorithms is judged to be feasible provided $k$ columns are selected. The detailed comparative results, together with the statistical summary of the comparisons, are displayed in Tables 4–6.

Once again, MMAS-ML outperforms uSLC and RWLS in terms of solution quality in all instances. The experimental results show that the integration model of swarm intelligence and heuristic rules has a significant effect on MKCP. At the same time, the contrast experiment also indicates that the algorithms designed for MSCP may not fit MKCP because of the different characteristics of the problems.

MKCP is a generalization of MSCP. Thus, whether the proposed MMAS-ML can find a valid solution to MSCP if the given $k$ is equal to the best-known solution for MSCP is further discussed. The experimental settings are the same as those of Section 6.1, except for the predefined values of $k$, which are derived from the results of RWLS [26]. In addition, it takes RWLS more than 100 s to find the best solutions of some difficult instances (e.g. 6437.21 s for NRE2). However, MMAS-ML is still executed with a time limit of 100 s to evaluate efficiency in solving MSCP. Fig. 2 shows the

**Table 7**
Comparison between MMAS-ML and three variants on $k_{90}$ instances.

| Eg | K90 | MMAS-G | | | ML | | | MMAS-M | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scp41 | 34 | **193** | **193** | 4.88 | 188 | 188 | 0.00 | 188 | 188 | 0.00 | **193** | **193** | 2.58 |
| scp42 | 33 | **193** | **193** | 1.09 | 190 | 190 | 0.00 | 192 | 191 | 0.00 | **193** | **193** | 0.93 |
| scp43 | 34 | **194** | **194** | 1.80 | 190 | 190 | 0.00 | 193 | 190.4 | 0.00 | **194** | **194** | 1.09 |
| scp44 | 34 | **191** | **191** | 1.56 | **191** | 186.2 | 0.08 | 186 | 185.3 | 0.00 | **191** | **191** | 0.96 |
| scp45 | 34 | **193** | **193** | 2.72 | 191 | 191 | 0.00 | 192 | 191.4 | 0.00 | **193** | **193** | 1.38 |
| scp46 | 33 | **191** | **191** | 1.37 | 186 | 186 | 0.00 | 189 | 187.3 | 0.00 | **191** | **191** | 0.95 |
| scp47 | 34 | **191** | **191** | 1.57 | 186 | 186 | 0.00 | 187 | 186.6 | 0.00 | **191** | **191** | 1.01 |
| scp48 | 33 | **192** | **192** | 1.92 | 187 | 187 | 0.00 | 189 | 187.8 | 0.00 | **192** | **192** | 1.12 |
| scp49 | 34 | **192** | **192** | 1.44 | 187 | 187 | 0.00 | 189 | 187.6 | 0.00 | **192** | **192** | 0.98 |
| scp410 | 34 | **192** | **192** | 2.67 | 189 | 189 | 0.00 | 189 | 189 | 0.00 | **192** | **192** | 2.70 |
| scp51 | 31 | **193** | **193** | 2.57 | 191 | 191 | 0.00 | 192 | 191.4 | 0.00 | **193** | **193** | 1.58 |
| scp52 | 31 | **194** | **194** | 2.98 | 191 | 191 | 0.00 | 192 | 191.6 | 0.00 | **194** | **194** | 1.99 |
| scp53 | 31 | **194** | **194** | 2.59 | **194** | 190.5 | 0.20 | 192 | 189.9 | 0.00 | **194** | **194** | 1.54 |
| scp54 | 31 | **194** | **194** | 2.62 | **194** | 191.9 | 0.17 | 193 | 192.8 | 0.00 | **194** | **194** | 1.77 |
| scp55 | 31 | **194** | **194** | 2.58 | 191 | 191 | 0.00 | 191 | 191 | 0.00 | **194** | **194** | 1.63 |
| scp56 | 31 | **194** | **194** | 2.76 | 191 | 191 | 0.00 | 191 | 191 | 0.00 | **194** | **194** | 1.74 |
| scp57 | 31 | **195** | **195** | 2.98 | 191 | 191 | 0.00 | 191 | 191 | 0.00 | **195** | **195** | 1.92 |
| scp58 | 31 | **194** | **194** | 2.72 | **194** | 186.8 | 0.02 | 189 | 187.6 | 0.00 | **194** | **194** | 1.73 |
| scp59 | 32 | **194** | **194** | 2.21 | 191 | 191 | 0.00 | **194** | 192 | 0.00 | **194** | **194** | 1.19 |
| scp510 | 31 | **194** | **194** | 4.05 | 188 | 188 | 0.00 | 190 | 188.3 | 0.00 | **194** | **194** | 2.94 |
| scp61 | 19 | **196** | **196** | 2.27 | 193 | 193 | 0.00 | 195 | 194 | 0.00 | **196** | **196** | 1.56 |
| scp62 | 18 | **195** | **195** | 2.79 | **195** | 192.2 | 0.20 | 194 | 192.7 | 0.00 | **195** | **195** | 1.87 |
| scp63 | 19 | **196** | **196** | 2.31 | 194 | 194 | 0.00 | **196** | 194.2 | 0.00 | **196** | **196** | 1.35 |
| scp64 | 18 | **192** | **192** | 2.78 | 188 | 188 | 0.00 | **192** | 189.8 | 0.00 | **192** | **192** | 1.30 |
| scp65 | 19 | **196** | **196** | 2.50 | 191 | 191 | 0.00 | 195 | 192.6 | 0.00 | **196** | **196** | 1.77 |
| scpa1 | 34 | **289** | **289** | 17.41 | 282 | 282 | 0.00 | 285 | 282.7 | 0.00 | **289** | **289** | 7.47 |
| scpa2 | 34 | **289** | **289** | 12.63 | 282 | 282 | 0.00 | 286 | 285.5 | 0.00 | **289** | **289** | 7.19 |
| scpa3 | 34 | **289** | **289** | 5.44 | 280 | 280 | 0.00 | 286 | 283.1 | 0.01 | **289** | **289** | 2.90 |
| scpa4 | 33 | **288** | **288** | 4.55 | 281 | 281 | 0.00 | 286 | 284.3 | 0.01 | **288** | **288** | 2.92 |
| scpa5 | 34 | **290** | **290** | 4.89 | 281 | 281 | 0.00 | 286 | 283.6 | 0.00 | **290** | **290** | 2.96 |
| scpb1 | 20 | **296** | **296** | 10.79 | 290 | 290 | 0.00 | 291 | 290.5 | 0.01 | **296** | **296** | 5.70 |
| scpb2 | 20 | **295** | **295** | 9.53 | 294 | 294 | 0.00 | 294 | 294 | 0.01 | **295** | **295** | 5.39 |
| scpb3 | 20 | **296** | **296** | 13.09 | 294 | 294 | 0.00 | 295 | 294.1 | 0.01 | **296** | **296** | 6.66 |
| scpb4 | 20 | **295** | 294.9 | 17.72 | 290 | 290 | 0.00 | 290 | 290 | 0.01 | **295** | **295** | 14.67 |
| scpb5 | 20 | **295** | **295** | 9.71 | 290 | 290 | 0.00 | 291 | 290.4 | 0.01 | **295** | **295** | 5.84 |
| scpc1 | 39 | **391** | **391** | 13.11 | 383 | 383 | 0.00 | 384 | 383.1 | 0.00 | **391** | **391** | 4.21 |
| scpc2 | 39 | **391** | **391** | 20.15 | 382 | 382 | 0.00 | 387 | 384.5 | 0.01 | **391** | **391** | 7.54 |
| scpc3 | 39 | **391** | 390.4 | 28.83 | 383 | 383 | 0.00 | 388 | 386.4 | 0.01 | **391** | **390.8** | 14.84 |
| scpc4 | 39 | **392** | **392** | 11.47 | 381 | 381 | 0.00 | 387 | 384.6 | 0.01 | **392** | **392** | 6.11 |
| scpc5 | 39 | **391** | **391** | 20.87 | 383 | 383 | 0.00 | 387 | 383.7 | 0.01 | **391** | 390.9 | 17.40 |
| scpd1 | 22 | **395** | 394.1 | 16.23 | 386 | 386 | 0.00 | 389 | 386.6 | 0.01 | **395** | **394.2** | 26.51 |
| scpd2 | 22 | **393** | 392.7 | 33.67 | 387 | 387 | 0.00 | 388 | 387.3 | 0.01 | **393** | **393** | 23.95 |
| scpd3 | 22 | **393** | **393** | 29.42 | 388 | 388 | 0.00 | 390 | 388.3 | 0.01 | **393** | **393** | 17.13 |
| scpd4 | 22 | **393** | 392.6 | 33.11 | 386 | 386 | 0.00 | 388 | 386.2 | 0.01 | **393** | **392.8** | 40.32 |
| scpd5 | 22 | **393** | 392.3 | 25.35 | 386 | 386 | 0.00 | 388 | 386.6 | 0.01 | **393** | **392.7** | 28.34 |
| scpnre1 | 14 | **488** | **487.9** | 84.07 | 481 | 481 | 0.01 | 484 | 481.9 | 0.05 | **488** | **487.9** | 59.88 |
| scpnre2 | 14 | **488** | **487.3** | 70.47 | 481 | 481 | 0.01 | 484 | 482.1 | 0.04 | **488** | **487.3** | 47.97 |
| scpnre3 | 14 | **488** | **487.3** | 78.52 | 481 | 481 | 0.01 | 486 | 482.7 | 0.04 | **488** | **487.3** | 51.10 |
| scpnre4 | 14 | **488** | 487.3 | 78.31 | 481 | 481 | 0.01 | 486 | 483.7 | 0.05 | **488** | **487.4** | 55.57 |
| scpnre5 | 14 | **488** | **487.3** | 76.79 | 487 | 481.6 | 1.02 | 485 | 483 | 0.05 | **488** | **487.3** | 59.87 |
| scpnrf1 | 9 | **497** | 495.8 | 58.41 | 493 | 493 | 0.01 | 494 | 493.1 | 0.10 | 496 | **496** | 62.04 |
| scpnrf2 | 9 | 496 | 495.7 | 50.54 | 496 | 492.8 | 11.75 | 495 | 492.5 | 0.10 | **497** | **496.2** | 47.21 |
| scpnrf3 | 9 | **497** | 495.7 | 58.88 | 496 | 492.9 | 21.43 | 495 | 492.7 | 0.11 | **497** | **496.1** | 31.78 |
| scpnrf4 | 9 | **497** | 495.5 | 44.96 | 496 | 492.7 | 9.73 | 494 | 492.4 | 0.12 | **497** | **496.4** | 28.00 |
| scpnrf5 | 9 | 496 | 495.7 | 34.85 | 496 | 492.7 | 7.34 | 495 | 492.9 | 0.12 | **497** | **496** | 24.53 |
| scpnrg1 | 55 | **984** | **982.4** | 57.86 | 972 | 972 | 0.01 | 975 | 972.9 | 0.03 | **984** | 981.8 | 38.04 |
| scpnrg2 | 55 | 983 | 982.1 | 58.96 | **984** | 975 | 19.12 | 977 | 975.2 | 0.04 | **984** | **982.5** | 49.96 |
| scpnrg3 | 55 | **983** | 979.8 | 48.69 | 980 | 968.3 | 3.71 | 972 | 969.7 | 0.04 | **983** | **981.1** | 37.29 |
| scpnrg4 | 55 | **982** | 980.4 | 45.02 | 981 | 968.4 | 3.61 | 972 | 970 | 0.03 | **982** | **980.9** | 45.87 |
| scpnrg5 | 55 | 981 | 980.2 | 53.85 | 978 | 968.1 | 5.97 | 972 | 970 | 0.04 | **983** | **980.9** | 46.82 |
| scpnrh1 | 31 | **991** | 989.7 | 63.98 | 984 | 984 | 0.02 | 985 | 984.1 | 0.07 | **991** | **989.8** | 77.69 |
| scpnrh2 | 31 | 991 | 989.6 | 77.62 | 984 | 984 | 0.02 | 984 | 984 | 0.06 | **992** | **989.8** | 68.81 |
| scpnrh3 | 31 | 992 | 989.8 | 72.35 | 984 | 984 | 0.02 | 984 | 984 | 0.07 | **993** | **990.3** | 82.14 |
| scpnrh4 | 31 | **992** | **990.4** | 68.22 | 984 | 984 | 0.02 | 985 | 984.1 | 0.08 | **992** | 989.9 | 71.99 |
| scpnrh5 | 31 | 991 | **989.9** | 67.43 | 984 | 984 | 0.02 | 985 | 984.1 | 0.08 | **992** | **989.9** | 73.04 |
| scpclr10 | 23 | **504** | **504** | 2.82 | **504** | 500.5 | 0.00 | **504** | 500.8 | 0.00 | **504** | **504** | 2.97 |
| scpclr11 | 21 | **1011** | **1011** | 9.11 | **1011** | **1011** | 0.08 | **1011** | 1005.5 | 0.02 | **1011** | **1011** | 5.55 |
| scpclr12 | 21 | **2023** | **2023** | 32.03 | **2023** | 2011.8 | 7.95 | 2018 | 2011.9 | 0.03 | **2023** | 2022.8 | 22.41 |
| scpclr13 | 21 | **4045** | **4045** | 68.03 | **4045** | **4045** | 2.34 | 4029 | 4024.5 | 0.14 | **4045** | **4045** | 45.99 |

(*continued on next page*)

coverage rate of the elements with a given $k$ by MMAS-ML for MSCP. It is observed that MMAS-ML can find the full coverage of the set cover problem in 68 out of 75 instances. For the remaining seven instances, the proposed algorithm can cover at least 98.93% of the elements. Recall that the optimization of MSCP should pay special attention to the unique sets even if they contain a small

**Table 7** (*continued*).

| Eg | K90 | MMAS-G | | | ML | | | MMAS-M | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scpcyc06 | 54 | **227** | **227** | 3.03 | **227** | 224.2 | 1.07 | 226 | 225.3 | 0.00 | **227** | **227** | 1.71 |
| scpcyc07 | 130 | **642** | **642** | 1.47 | 630 | 630 | 0.00 | 637 | 634.7 | 0.00 | **642** | **642** | 0.49 |
| scpcyc08 | 308 | 1700 | 1696.5 | 49.64 | 1696 | 1693.5 | 33.54 | 1686 | 1676.8 | 0.02 | **1711** | **1697.7** | 48.65 |
| scpcyc09 | 695 | **4326** | **4319.3** | 8.10 | 4318 | 4316.7 | 46.32 | 4258 | 4245.2 | 0.10 | 4318 | 4316.8 | 1.54 |
| scpcyc10 | 1618 | **10 980** | **10 968.2** | 29.96 | 10 742 | 10 742 | 0.26 | 10 742 | 10 742 | 0.51 | **10 980** | 10 958.8 | 50.23 |
| scpcyc11 | 3571 | 26 587 | **26 573.1** | 59.79 | 26 534 | 26 325.4 | 42.95 | 26 133 | 26 130.9 | 3.00 | **26 649** | 26 569 | 63.37 |
| Avg | | 1046.93 | 1046.11 | 25.51 | 1038.72 | 1034.56 | 2.92 | 1033.07 | 1031.23 | 0.07 | 1047.89 | 1045.99 | 21.2 |

**Table 8**
Comparison between MMAS-ML and three variants on $k_{95}$ instances.

| Eg | K95 | MMAS-G | | | ML | | | MMAS-M | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scp41 | 36 | **197** | **197** | 1.68 | **197** | 193.5 | 0.14 | 196 | 192.9 | 0.00 | **197** | **197** | 1.00 |
| scp42 | 35 | **198** | **198** | 1.50 | **198** | 197.2 | 0.07 | 197 | 195.4 | 0.00 | **198** | **198** | 0.99 |
| scp43 | 36 | **198** | **198** | 2.67 | **198** | 195.3 | 0.02 | 196 | 195.1 | 0.00 | **198** | **198** | 2.25 |
| scp44 | 36 | **196** | **196** | 1.78 | **196** | **196** | 0.28 | 192 | 191 | 0.00 | **196** | **196** | 1.04 |
| scp45 | 36 | **197** | **197** | 2.00 | **197** | 196.2 | 0.22 | 196 | 195.6 | 0.00 | **197** | **197** | 1.06 |
| scp46 | 35 | **196** | **196** | 1.29 | **196** | **196** | 0.06 | 194 | 192.9 | 0.00 | **196** | **196** | 0.96 |
| scp47 | 36 | **196** | **196** | 1.49 | **196** | 195.4 | 0.04 | 193 | 191.1 | 0.00 | **196** | **196** | 0.97 |
| scp48 | 35 | **196** | **196** | 1.77 | **196** | **196** | 0.20 | 194 | 192.7 | 0.00 | **196** | **196** | 1.02 |
| scp49 | 36 | **196** | **196** | 1.53 | 192 | 192 | 0.00 | 192 | 192 | 0.00 | **196** | **196** | 0.93 |
| scp410 | 36 | **196** | **196** | 1.47 | **196** | 193.6 | 0.01 | 194 | 193.2 | 0.00 | **196** | **196** | 0.99 |
| scp51 | 32 | **196** | **196** | 3.71 | 194 | 194 | 0.00 | 194 | 194 | 0.00 | **196** | **196** | 2.78 |
| scp52 | 32 | **196** | **196** | 2.55 | **196** | 195.4 | 0.28 | 195 | 194.1 | 0.00 | **196** | **196** | 1.85 |
| scp53 | 32 | **196** | **196** | 2.50 | **196** | **196** | 0.09 | 194 | 192.8 | 0.00 | **196** | **196** | 1.47 |
| scp54 | 32 | **196** | **196** | 2.38 | **196** | 195.7 | 0.20 | 195 | 194.7 | 0.00 | **196** | **196** | 1.54 |
| scp55 | 32 | **197** | **196.8** | 17.75 | 194 | 194 | 0.00 | 194 | 194 | 0.00 | **197** | 196.7 | 16.97 |
| scp56 | 32 | **196** | **196** | 2.39 | **196** | 193.3 | 0.01 | 194 | 193.1 | 0.00 | **196** | **196** | 1.59 |
| scp57 | 32 | **197** | **197** | 2.75 | 197 | 194.6 | 0.11 | 194 | 193.4 | 0.00 | **197** | **197** | 1.83 |
| scp58 | 32 | **196** | **196** | 2.61 | **196** | 193.2 | 0.32 | 190 | 189.6 | 0.00 | **196** | **196** | 1.65 |
| scp59 | 33 | **197** | **197** | 2.94 | **197** | 194.2 | 0.11 | 196 | 193.5 | 0.00 | **197** | **197** | 1.81 |
| scp510 | 32 | **196** | **196** | 3.10 | **196** | 191.5 | 0.07 | 192 | 191.1 | 0.00 | **196** | **196** | 1.74 |
| scp61 | 20 | **199** | **199** | 2.95 | **199** | 197.5 | 0.12 | 198 | 196.6 | 0.00 | **199** | **199** | 1.63 |
| scp62 | 19 | **198** | **198** | 2.99 | **198** | 197.1 | 0.48 | 197 | 196.2 | 0.00 | **198** | **198** | 1.92 |
| scp63 | 20 | **199** | **199** | 2.69 | **199** | 197.2 | 0.01 | 198 | 197.1 | 0.00 | **199** | **199** | 1.65 |
| scp64 | 19 | **196** | **196** | 2.61 | **196** | **196** | 0.11 | 195 | 193.4 | 0.00 | **196** | **196** | 1.76 |
| scp65 | 20 | **198** | **198** | 2.10 | **198** | 196.2 | 0.05 | 197 | 195.6 | 0.00 | **198** | **198** | 1.50 |
| scpa1 | 36 | **295** | **295** | 8.20 | **295** | 294.3 | 14.59 | 292 | 290.2 | 0.00 | **295** | **295** | 3.78 |
| scpa2 | 36 | **295** | **295** | 5.81 | **295** | **295** | 3.80 | 292 | 291.3 | 0.00 | **295** | **295** | 3.14 |
| scpa3 | 36 | **295** | **295** | 5.03 | **295** | **295** | 1.82 | 291 | 289 | 0.00 | **295** | **295** | 3.01 |
| scpa4 | 35 | **294** | **294** | 4.75 | **294** | **294** | 1.81 | 292 | 290.1 | 0.00 | **294** | **294** | 3.03 |
| scpa5 | 36 | **296** | **296** | 9.26 | **296** | 295.1 | 10.34 | 293 | 291 | 0.01 | **296** | **296** | 6.98 |
| scpb1 | 21 | **299** | **299** | 9.68 | 295 | 295 | 0.00 | 295 | 295 | 0.01 | **299** | **299** | 6.11 |
| scpb2 | 21 | **299** | **299** | 14.05 | **299** | 297.2 | 0.22 | 298 | 297.1 | 0.01 | **299** | **299** | 8.35 |
| scpb3 | 21 | **299** | **299** | 15.43 | 297 | 297 | 0.00 | 297 | 297 | 0.01 | **299** | **299** | 9.51 |
| scpb4 | 21 | **299** | 298.5 | 26.49 | **299** | 295.7 | 10.26 | 295 | 294.4 | 0.01 | **299** | **298.9** | 32.78 |
| scpb5 | 21 | **298** | **298** | 8.80 | **298** | 295.6 | 0.33 | 296 | 294.5 | 0.01 | **298** | **298** | 5.58 |
| scpc1 | 41 | **397** | 396.7 | 34.72 | **397** | 389.8 | 0.66 | 391 | 389.2 | 0.01 | **397** | **396.8** | 14.00 |
| scpc2 | 41 | **397** | **397** | 24.06 | **397** | 396 | 31.16 | 394 | 391.2 | 0.01 | **397** | **397** | 10.32 |
| scpc3 | 41 | **397** | 396.4 | 12.89 | **397** | 396.1 | 29.49 | 394 | 392.3 | 0.01 | **397** | **396.7** | 22.82 |
| scpc4 | 41 | **397** | **397** | 10.35 | **397** | **397** | 13.60 | 393 | 390.2 | 0.01 | **397** | **397** | 4.31 |
| scpc5 | 41 | **397** | **396.4** | 25.68 | **397** | 392 | 12.86 | 391 | 389.5 | 0.00 | **397** | 396.3 | 15.44 |
| scpd1 | 23 | **398** | **398** | 23.78 | **398** | 391.7 | 1.15 | 393 | 391.2 | 0.01 | **398** | **398** | 12.04 |
| scpd2 | 23 | **397** | **397** | 39.06 | **397** | 393.5 | 3.54 | 393 | 392.3 | 0.01 | **397** | **397** | 18.93 |
| scpd3 | 23 | **397** | 396.5 | 22.91 | **397** | 395.4 | 14.62 | 394 | 393.1 | 0.02 | **397** | **396.7** | 30.18 |
| scpd4 | 23 | **397** | 396.5 | 30.33 | 396 | 391.8 | 6.39 | 392 | 390.4 | 0.01 | **397** | **396.6** | 33.35 |
| scpd5 | 23 | **397** | **396.5** | 29.80 | **397** | 393.8 | 17.04 | 393 | 390.9 | 0.01 | **397** | **396.5** | 45.79 |
| scpnre1 | 15 | **495** | 494.3 | 72.86 | **495** | 491.3 | 24.59 | 491 | 489.7 | 0.05 | **495** | **494.4** | 52.94 |
| scpnre2 | 15 | **495** | **494.6** | 76.20 | **495** | 491.8 | 19.31 | 492 | 490.1 | 0.04 | **495** | 494.3 | 54.62 |
| scpnre3 | 15 | **495** | 494.3 | 77.50 | 494 | 491.6 | 14.39 | 492 | 490.3 | 0.05 | **495** | **494.6** | 63.06 |
| scpnre4 | 15 | **495** | 494.5 | 69.42 | **495** | 492.8 | 34.21 | 492 | 490.1 | 0.05 | **495** | **494.6** | 67.85 |
| scpnre5 | 15 | **495** | 494.2 | 77.91 | 494 | 492.6 | 33.75 | 492 | 490.3 | 0.05 | **495** | **494.3** | 49.46 |
| scpnrf1 | 10 | **500** | **500** | 56.16 | **500** | 498.9 | 19.87 | 499 | 498.6 | 0.09 | **500** | **500** | 25.53 |
| scpnrf2 | 10 | **500** | 499.9 | 50.94 | 498 | 498 | 0.01 | 498 | 498 | 0.08 | **500** | **500** | 38.72 |
| scpnrf3 | 10 | **500** | **500** | 62.09 | 498 | 498 | 0.01 | 498 | 498 | 0.09 | **500** | **500** | 57.46 |
| scpnrf4 | 10 | **500** | **500** | 44.49 | **500** | 498.2 | 0.54 | 499 | 498.1 | 0.09 | **500** | **500** | 41.56 |
| scpnrf5 | 10 | **500** | 499.9 | 50.02 | **500** | 498.2 | 7.91 | 499 | 498.1 | 0.11 | **500** | **500** | 51.79 |
| scpnrg1 | 58 | 994 | **993.2** | 54.53 | 994 | 990 | 37.06 | 987 | 985.4 | 0.03 | **995** | 992.8 | 27.29 |
| scpnrg2 | 58 | 994 | **992.9** | 50.17 | 993 | 990.9 | 39.37 | 988 | 986.4 | 0.03 | **995** | **992.9** | 36.86 |
| scpnrg3 | 58 | **993** | 991.5 | 43.92 | 992 | 988.5 | 33.34 | 985 | 982.6 | 0.03 | **993** | **991.7** | 39.88 |

(*continued on next page*)

**Table 8** (*continued*).

| Eg | K95 | MMAS-G | | | ML | | | MMAS-M | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scpnrg4 | 58 | 993 | **991.4** | 42.97 | 991 | 988.3 | 43.28 | 985 | 983.1 | 0.04 | **994** | **991.4** | 35.13 |
| scpnrg5 | 58 | **993** | **991.6** | 46.24 | **993** | 989.8 | 32.76 | 985 | 982.9 | 0.04 | **993** | 991.4 | 45.29 |
| scpnrh1 | 32 | **996** | **994.5** | 67.44 | 994 | 991.4 | 27.14 | 991 | 990.1 | 0.08 | **996** | 994.4 | 59.51 |
| scpnrh2 | 32 | **995** | **994.6** | 61.90 | 994 | 991.5 | 31.44 | 991 | 990.2 | 0.09 | **995** | 994.1 | 52.87 |
| scpnrh3 | 32 | **996** | **995** | 67.80 | 995 | 991 | 25.17 | 991 | 990 | 0.08 | **996** | 994 | 59.77 |
| scpnrh4 | 32 | **996** | **994.5** | 72.19 | 994 | 991 | 30.18 | 991 | 990.2 | 0.09 | **996** | **994.5** | 44.86 |
| scpnrh5 | 32 | **995** | **994.6** | 44.91 | 992 | 990.4 | 30.33 | 991 | 989.8 | 0.09 | **995** | 994.3 | 43.74 |
| scpclr10 | 24 | **509** | **509** | 2.40 | **509** | 505.8 | 0.01 | **509** | 505.2 | 0.00 | **509** | **509** | 2.84 |
| scpclr11 | 22 | **1015** | **1015** | 9.87 | **1015** | **1015** | 0.06 | 1010 | 1005.5 | 0.02 | **1015** | **1015** | 6.78 |
| scpclr12 | 22 | **2031** | **2031** | 28.17 | **2031** | **2031** | 1.79 | **2031** | 2023.8 | 0.03 | **2031** | **2031** | 18.07 |
| scpclr13 | 22 | **4063** | 4058.8 | 80.29 | 4059 | 4056.7 | 31.54 | 4048 | 4041.7 | 0.12 | **4063** | **4059.3** | 66.80 |
| scpcyc06 | 57 | **235** | **235** | 0.85 | **235** | **235** | 0.02 | **235** | 232.7 | 0.00 | **235** | **235** | 0.49 |
| scpcyc07 | 137 | **658** | 657.9 | 4.96 | 651 | 651 | 0.00 | 657 | 655.5 | 0.00 | **658** | **658** | 2.50 |
| scpcyc08 | 325 | 1751 | **1746.9** | 38.10 | 1743 | 1743 | 0.97 | 1736 | 1718.9 | 0.03 | **1752** | 1744 | 15.31 |
| scpcyc09 | 733 | 4465 | **4460.3** | 46.36 | 4456 | 4451.3 | 70.80 | 4394 | 4382.9 | 0.13 | **4468** | 4454.1 | 45.56 |
| scpcyc10 | 1708 | 11 258 | 11 251.2 | 64.23 | 11 234 | 11 226 | 84.45 | 11 131 | 11 083.5 | 0.66 | **11 292** | **11 265.4** | 69.73 |
| scpcyc11 | 3770 | 27 300 | **27 270.9** | 60.59 | 27 254 | 27 134 | 74.08 | 26 731 | 26 728.3 | 3.07 | **27 315** | 27 243.9 | 61.05 |
| Avg | | 1067.52 | 1066.62 | 26.13 | 1065.75 | 1062.28 | 12.33 | 1054.07 | 1051.61 | 0.07 | 1068.27 | 1066.33 | 21.06 |

**Table 9**
Comparison between MMAS-ML and two variants on $k_{90}$ instances.

| Eg | K90 | MMAS-L | | | MMAS-W | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scp41 | 34 | **193** | **193** | 12.80 | **193** | **193** | 7.04 | **193** | **193** | 2.58 |
| scp42 | 33 | **193** | **193** | 1.36 | **193** | **193** | 1.76 | **193** | **193** | 0.93 |
| scp43 | 34 | **194** | **194** | 8.29 | **194** | **194** | 2.34 | **194** | **194** | 1.09 |
| scp44 | 34 | **191** | **191** | 1.49 | **191** | **191** | 1.87 | **191** | **191** | 0.96 |
| scp45 | 34 | **193** | **193** | 4.38 | **193** | **193** | 3.15 | **193** | **193** | 1.38 |
| scp46 | 33 | **191** | **191** | 1.47 | **191** | **191** | 1.87 | **191** | **191** | 0.95 |
| scp47 | 34 | **191** | **191** | 1.36 | **191** | **191** | 2.07 | **191** | **191** | 1.01 |
| scp48 | 33 | **192** | **192** | 1.57 | **192** | **192** | 2.33 | **192** | **192** | 1.12 |
| scp49 | 34 | **192** | **192** | 1.38 | **192** | **192** | 1.91 | **192** | **192** | 0.98 |
| scp410 | 34 | **192** | **192** | 9.34 | **192** | **192** | 2.58 | **192** | **192** | 2.70 |
| scp51 | 31 | **193** | **193** | 2.21 | **193** | **193** | 3.46 | **193** | **193** | 1.58 |
| scp52 | 31 | **194** | **194** | 5.41 | **194** | **194** | 3.22 | **194** | **194** | 1.99 |
| scp53 | 31 | **194** | **194** | 2.31 | **194** | **194** | 2.72 | **194** | **194** | 1.54 |
| scp54 | 31 | **194** | **194** | 2.37 | **194** | **194** | 2.69 | **194** | **194** | 1.77 |
| scp55 | 31 | **194** | **194** | 2.08 | **194** | **194** | 3.55 | **194** | **194** | 1.63 |
| scp56 | 31 | **194** | **194** | 2.47 | **194** | **194** | 3.74 | **194** | **194** | 1.74 |
| scp57 | 31 | **195** | **195** | 2.66 | **195** | **195** | 3.62 | **195** | **195** | 1.92 |
| scp58 | 31 | **194** | **194** | 3.02 | **194** | **194** | 2.96 | **194** | **194** | 1.73 |
| scp59 | 32 | **194** | **194** | 1.85 | **194** | **194** | 2.62 | **194** | **194** | 1.19 |
| scp510 | 31 | **194** | **194** | 2.96 | **194** | **194** | 9.37 | **194** | **194** | 2.94 |
| scp61 | 19 | **196** | **196** | 2.09 | **196** | **196** | 2.98 | **196** | **196** | 1.56 |
| scp62 | 18 | **195** | **195** | 2.67 | **195** | **195** | 2.93 | **195** | **195** | 1.87 |
| scp63 | 19 | **196** | **196** | 1.87 | **196** | **196** | 3.40 | **196** | **196** | 1.35 |
| scp64 | 18 | **192** | **192** | 1.92 | **192** | **192** | 2.38 | **192** | **192** | 1.30 |
| scp65 | 19 | **196** | **196** | 2.47 | **196** | **196** | 3.06 | **196** | **196** | 1.77 |
| scpa1 | 34 | **289** | **289** | 10.07 | **289** | **289** | 14.07 | **289** | **289** | 7.47 |
| scpa2 | 34 | **289** | **289** | 19.19 | **289** | **289** | 17.07 | **289** | **289** | 7.19 |
| scpa3 | 34 | **289** | **289** | 4.41 | **289** | **289** | 6.04 | **289** | **289** | 2.90 |
| scpa4 | 33 | **288** | **288** | 4.65 | **288** | **288** | 6.43 | **288** | **288** | 2.92 |
| scpa5 | 34 | **290** | **290** | 3.59 | **290** | **290** | 6.69 | **290** | **290** | 2.96 |
| scpb1 | 20 | **296** | **296** | 8.60 | **296** | **296** | 13.20 | **296** | **296** | 5.70 |
| scpb2 | 20 | **295** | **295** | 8.32 | **295** | **295** | 13.21 | **295** | **295** | 5.39 |
| scpb3 | 20 | **296** | **296** | 10.21 | **296** | **296** | 15.78 | **296** | **296** | 6.66 |
| scpb4 | 20 | **295** | **295** | 15.12 | **295** | **295** | 24.11 | **295** | **295** | 14.67 |
| scpb5 | 20 | **295** | **295** | 8.48 | **295** | **295** | 12.97 | **295** | **295** | 5.84 |
| scpc1 | 39 | **391** | **391** | 9.40 | **391** | **391** | 14.64 | **391** | **391** | 4.21 |
| scpc2 | 39 | **391** | **391** | 13.78 | **391** | 390.9 | 25.24 | **391** | **391** | 7.54 |
| scpc3 | 39 | **391** | **390.8** | 15.69 | **391** | 390.6 | 22.63 | **391** | **390.8** | 14.84 |
| scpc4 | 39 | **392** | **392** | 6.94 | **392** | **392** | 28.56 | **392** | **392** | 6.11 |
| scpc5 | 39 | **391** | **391** | 7.13 | **391** | 390.6 | 33.83 | **391** | 390.9 | 17.40 |
| scpd1 | 22 | **395** | 394.1 | 21.80 | **395** | 394.1 | 35.32 | **395** | **394.2** | 26.51 |
| scpd2 | 22 | **393** | **393** | 18.31 | **393** | 392.7 | 39.39 | **393** | **393** | 23.95 |
| scpd3 | 22 | **393** | **393** | 25.84 | **393** | 392.8 | 35.54 | **393** | **393** | 17.13 |
| scpd4 | 22 | **393** | **392.8** | 29.87 | **393** | 392.3 | 39.90 | **393** | **392.8** | 40.32 |
| scpd5 | 22 | **393** | 392.4 | 22.91 | **393** | 392 | 25.78 | **393** | **392.7** | 28.34 |
| scpnre1 | 14 | **488** | **488** | 71.26 | **488** | 487.3 | 79.79 | **488** | 487.9 | 59.88 |
| scpnre2 | 14 | **488** | **487.4** | 71.87 | **488** | 486.8 | 72.28 | **488** | 487.3 | 47.97 |

(*continued on next page*)

**Table 9** (*continued*).

| Eg | K90 | MMAS-L | | | MMAS-W | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scpnre3 | 14 | **488** | 487.2 | 62.62 | **488** | 487.1 | 68.84 | **488** | **487.3** | 51.10 |
| scpnre4 | 14 | **488** | 487.3 | 64.76 | 487 | 486.9 | 60.94 | **488** | **487.4** | 55.57 |
| scpnre5 | 14 | **488** | **487.4** | 68.45 | 487 | 486.8 | 68.29 | **488** | 487.3 | 59.87 |
| scpnrf1 | 9 | **496** | 495.9 | 47.23 | 495 | 494.9 | 68.47 | **496** | **496** | 62.04 |
| scpnrf2 | 9 | **497** | 496 | 38.33 | 496 | 495.3 | 37.66 | **497** | **496.2** | 47.21 |
| scpnrf3 | 9 | **497** | **496.2** | 41.05 | **497** | 495.5 | 33.47 | **497** | 496.1 | 31.78 |
| scpnrf4 | 9 | **497** | 496.3 | 32.08 | **497** | 495.3 | 30.99 | **497** | **496.4** | 28.00 |
| scpnrf5 | 9 | **497** | **496** | 30.04 | 496 | 495.5 | 27.76 | **497** | **496** | 24.53 |
| scpnrg1 | 55 | 983 | **982.1** | 38.95 | 983 | 979.4 | 58.08 | **984** | 981.8 | 38.04 |
| scpnrg2 | 55 | 983 | 982.3 | 39.88 | 981 | 979 | 49.05 | **984** | **982.5** | 49.96 |
| scpnrg3 | 55 | 982 | 981 | 48.51 | 981 | 979.1 | 67.40 | **983** | **981.1** | 37.29 |
| scpnrg4 | 55 | **982** | **981** | 48.77 | 980 | 978.8 | 62.34 | **982** | 980.9 | 45.87 |
| scpnrg5 | 55 | **983** | **981.1** | 41.55 | 981 | 978.6 | 62.93 | **983** | 980.9 | 46.82 |
| scpnrh1 | 31 | **991** | **990.1** | 82.51 | 990 | 986.6 | 59.64 | **991** | 989.8 | 77.69 |
| scpnrh2 | 31 | 991 | **990.2** | 85.54 | 988 | 985.3 | 60.03 | **992** | 989.8 | 68.81 |
| scpnrh3 | 31 | 992 | 989.9 | 77.23 | 990 | 986.6 | 67.69 | **993** | **990.3** | 82.14 |
| scpnrh4 | 31 | 991 | **990.2** | 80.63 | 990 | 986.8 | 57.61 | **992** | 989.9 | 71.99 |
| scpnrh5 | 31 | **992** | **990.2** | 78.80 | 988 | 986.4 | 64.50 | **992** | 989.9 | 73.04 |
| scpclr10 | 23 | **504** | **504** | 4.31 | **504** | **504** | 4.46 | **504** | **504** | 2.97 |
| scpclr11 | 21 | **1011** | **1011** | 8.87 | **1011** | **1011** | 8.70 | **1011** | **1011** | 5.55 |
| scpclr12 | 21 | **2023** | 2022 | 45.65 | **2023** | **2023** | 32.30 | **2023** | 2022.8 | 22.41 |
| scpclr13 | 21 | 4045 | 4045 | 68.57 | **4047** | **4045.2** | 72.15 | 4045 | 4045 | 45.99 |
| scpcyc06 | 54 | **227** | 226.4 | 22.71 | **227** | **227** | 1.30 | **227** | **227** | 1.71 |
| scpcyc07 | 130 | **642** | 641.8 | 5.41 | **642** | **642** | 1.06 | **642** | **642** | 0.49 |
| scpcyc08 | 308 | 1693 | 1692.2 | 4.31 | 1708 | **1699.1** | 36.27 | **1711** | 1697.7 | 48.65 |
| scpcyc09 | 695 | 4318 | 4317 | 8.30 | **4352** | **4322.9** | 9.97 | 4318 | 4316.8 | 1.54 |
| scpcyc10 | 1618 | **10 981** | 10 956.9 | 51.07 | 10 954 | 10 938 | 27.14 | 10 980 | **10 958.8** | 50.23 |
| scpcyc11 | 3571 | 26 603 | 26 549 | 76.42 | 26 631 | 26 542.1 | 86.71 | **26 649** | **26 569** | 63.37 |
| Avg | | 1046.97 | 1045.62 | 23.89 | 1047.36 | 1044.94 | 25.97 | 1047.89 | 1045.99 | 21.2 |

**Table 10**
Comparison between MMAS-ML and two variants on $k_{95}$ instances.

| Eg | K95 | MMAS-L | | | MMAS-W | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scp41 | 36 | **197** | **197** | 1.85 | **197** | **197** | 2.17 | **197** | **197** | 1.00 |
| scp42 | 35 | **198** | **198** | 1.61 | **198** | **198** | 1.93 | **198** | **198** | 0.99 |
| scp43 | 36 | **198** | 197.3 | 4.83 | **198** | **198** | 3.96 | **198** | **198** | 2.25 |
| scp44 | 36 | **196** | **196** | 1.39 | **196** | **196** | 1.83 | **196** | **196** | 1.04 |
| scp45 | 36 | **197** | **197** | 1.87 | **197** | **197** | 2.14 | **197** | **197** | 1.06 |
| scp46 | 35 | **196** | **196** | 1.41 | **196** | **196** | 1.69 | **196** | **196** | 0.96 |
| scp47 | 36 | **196** | **196** | 1.38 | **196** | **196** | 1.84 | **196** | **196** | 0.97 |
| scp48 | 35 | **196** | **196** | 1.48 | **196** | **196** | 2.07 | **196** | **196** | 1.02 |
| scp49 | 36 | **196** | **196** | 1.43 | **196** | **196** | 1.98 | **196** | **196** | 0.93 |
| scp410 | 36 | **196** | **196** | 2.51 | **196** | **196** | 2.05 | **196** | **196** | 0.99 |
| scp51 | 32 | **196** | **196** | 4.46 | **196** | **196** | 11.34 | **196** | **196** | 2.78 |
| scp52 | 32 | **196** | **196** | 2.79 | **196** | **196** | 3.16 | **196** | **196** | 1.85 |
| scp53 | 32 | **196** | **196** | 2.43 | **196** | **196** | 2.63 | **196** | **196** | 1.47 |
| scp54 | 32 | **196** | **196** | 2.39 | **196** | **196** | 2.67 | **196** | **196** | 1.54 |
| scp55 | 32 | **197** | 196.1 | 4.20 | **197** | 196.5 | 11.85 | **197** | **196.7** | 16.97 |
| scp56 | 32 | **196** | **196** | 2.21 | **196** | **196** | 3.14 | **196** | **196** | 1.59 |
| scp57 | 32 | **197** | **197** | 2.81 | **197** | **197** | 3.17 | **197** | **197** | 1.83 |
| scp58 | 32 | **196** | **196** | 2.84 | **196** | **196** | 3.26 | **196** | **196** | 1.65 |
| scp59 | 33 | **197** | **197** | 2.25 | **197** | **197** | 3.44 | **197** | **197** | 1.81 |
| scp510 | 32 | **196** | **196** | 2.45 | **196** | **196** | 4.04 | **196** | **196** | 1.74 |
| scp61 | 20 | **199** | **199** | 2.40 | **199** | **199** | 3.51 | **199** | **199** | 1.63 |
| scp62 | 19 | **198** | **198** | 2.84 | **198** | **198** | 2.61 | **198** | **198** | 1.92 |
| scp63 | 20 | **199** | **199** | 2.31 | **199** | **199** | 3.59 | **199** | **199** | 1.65 |
| scp64 | 19 | **196** | **196** | 2.61 | **196** | **196** | 2.79 | **196** | **196** | 1.76 |
| scp65 | 20 | **198** | **198** | 1.93 | **198** | **198** | 3.30 | **198** | **198** | 1.50 |
| scpa1 | 36 | **295** | **295** | 7.34 | **295** | 294.9 | 21.44 | **295** | **295** | 3.78 |
| scpa2 | 36 | **295** | **295** | 4.62 | **295** | **295** | 6.55 | **295** | **295** | 3.14 |
| scpa3 | 36 | **295** | **295** | 5.24 | **295** | **295** | 5.79 | **295** | **295** | 3.01 |
| scpa4 | 35 | **294** | **294** | 4.15 | **294** | **294** | 5.81 | **294** | **294** | 3.03 |
| scpa5 | 36 | **296** | **296** | 15.54 | **296** | **296** | 14.60 | **296** | **296** | 6.98 |
| scpb1 | 21 | **299** | **299** | 8.73 | **299** | **299** | 13.86 | **299** | **299** | 6.11 |
| scpb2 | 21 | **299** | **299** | 14.24 | **299** | **299** | 28.03 | **299** | **299** | 8.35 |
| scpb3 | 21 | **299** | **299** | 18.35 | **299** | **299** | 24.16 | **299** | **299** | 9.51 |
| scpb4 | 21 | **299** | 298.7 | 28.95 | **299** | 298.4 | 26.73 | **299** | **298.9** | 32.78 |
| scpb5 | 21 | **298** | **298** | 7.78 | **298** | **298** | 12.42 | **298** | **298** | 5.58 |
| scpc1 | 41 | **397** | **397** | 41.01 | **397** | 396.2 | 23.85 | **397** | 396.8 | 14.00 |
| scpc2 | 41 | **397** | **397** | 19.02 | **397** | 396.8 | 26.35 | **397** | **397** | 10.32 |

(*continued on next page*)

**Table 10** (*continued*).

| Eg | K95 | MMAS-L | | | MMAS-W | | | MMAS-ML | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BEST | AVG | TIME | BEST | AVG | TIME | BEST | AVG | TIME |
| scpc3 | 41 | **397** | 396.5 | 25.59 | **397** | 396.4 | 23.37 | **397** | **396.7** | 22.82 |
| scpc4 | 41 | **397** | **397** | 6.72 | **397** | 396.9 | 11.88 | **397** | **397** | 4.31 |
| scpc5 | 41 | **397** | **396.4** | 23.40 | **397** | 396.2 | 27.97 | **397** | 396.3 | 15.44 |
| scpd1 | 23 | **398** | **398** | 16.54 | **398** | **398** | 33.61 | **398** | **398** | 12.04 |
| scpd2 | 23 | **397** | **397** | 19.77 | **397** | 396.6 | 39.60 | **397** | **397** | 18.93 |
| scpd3 | 23 | **397** | **396.9** | 21.26 | **397** | 396.5 | 31.44 | **397** | 396.7 | 30.18 |
| scpd4 | 23 | **397** | **396.8** | 23.53 | **397** | 396.1 | 32.20 | **397** | 396.6 | 33.35 |
| scpd5 | 23 | **397** | 396.4 | 19.46 | **397** | 396.2 | 30.06 | **397** | **396.5** | 45.79 |
| scpnre1 | 15 | **495** | **494.6** | 69.16 | 494 | 493.6 | 86.26 | **495** | 494.4 | 52.94 |
| scpnre2 | 15 | **495** | **494.6** | 72.21 | **495** | 493.8 | 82.56 | **495** | 494.3 | 54.62 |
| scpnre3 | 15 | **495** | **494.6** | 73.14 | 494 | 493.7 | 84.47 | **495** | **494.6** | 63.06 |
| scpnre4 | 15 | **495** | 494.5 | 67.05 | 494 | 493.4 | 74.68 | **495** | **494.6** | 67.85 |
| scpnre5 | 15 | **495** | **494.5** | 68.92 | 494 | 493.3 | 75.66 | **495** | 494.3 | 49.46 |
| scpnrf1 | 10 | **500** | **500** | 26.97 | **500** | 499.8 | 54.10 | **500** | **500** | 25.53 |
| scpnrf2 | 10 | **500** | **500** | 40.99 | **500** | 499.7 | 58.71 | **500** | **500** | 38.72 |
| scpnrf3 | 10 | **500** | **500** | 35.60 | **500** | 499.6 | 63.21 | **500** | **500** | 57.46 |
| scpnrf4 | 10 | **500** | **500** | 32.22 | **500** | 499.2 | 58.55 | **500** | **500** | 41.56 |
| scpnrf5 | 10 | **500** | **500** | 38.40 | **500** | 499.8 | 64.31 | **500** | **500** | 51.79 |
| scpnrg1 | 58 | 994 | **993.2** | 55.54 | 993 | 990.4 | 49.65 | **995** | 992.8 | 27.29 |
| scpnrg2 | 58 | **995** | **993.3** | 44.54 | 993 | 991.1 | 56.43 | **995** | 992.9 | 36.86 |
| scpnrg3 | 58 | **993** | **992.1** | 41.07 | **993** | 990 | 59.31 | **993** | 991.7 | 39.88 |
| scpnrg4 | 58 | 993 | 991.3 | 52.09 | 991 | 989.4 | 48.89 | **994** | **991.4** | 35.13 |
| scpnrg5 | 58 | **993** | **991.8** | 39.96 | 992 | 990 | 58.19 | **993** | 991.4 | 45.29 |
| scpnrh1 | 32 | **996** | **994.8** | 78.81 | 993 | 991.8 | 55.28 | **996** | 994.4 | 59.51 |
| scpnrh2 | 32 | **996** | **994.7** | 81.85 | 994 | 991.5 | 39.87 | 995 | 994.1 | 52.87 |
| scpnrh3 | 32 | 995 | **994.2** | 68.55 | 994 | 991.5 | 41.68 | **996** | 994 | 59.77 |
| scpnrh4 | 32 | **996** | **994.6** | 79.24 | 993 | 991.4 | 47.41 | **996** | 994.5 | 44.86 |
| scpnrh5 | 32 | **995** | **994.3** | 73.11 | 994 | 991.6 | 63.54 | **995** | **994.3** | 43.74 |
| scpclr10 | 24 | **509** | **509** | 3.65 | **509** | **509** | 4.60 | **509** | **509** | 2.84 |
| scpclr11 | 22 | **1015** | **1015** | 10.02 | **1015** | **1015** | 9.63 | **1015** | **1015** | 6.78 |
| scpclr12 | 22 | **2031** | **2031** | 31.88 | **2031** | **2031** | 24.43 | **2031** | **2031** | 18.07 |
| scpclr13 | 22 | 4059 | 4056.5 | 71.72 | **4063** | 4059.2 | 81.46 | **4063** | **4059.3** | 66.80 |
| scpcyc06 | 57 | **235** | **235** | 0.68 | **235** | **235** | 0.93 | **235** | **235** | 0.49 |
| scpcyc07 | 137 | 657 | 657 | 0.02 | **658** | **658** | 2.44 | **658** | **658** | 2.50 |
| scpcyc08 | 325 | 1743 | 1743 | 0.12 | 1746 | 1743.8 | 13.44 | **1752** | **1744** | 15.31 |
| scpcyc09 | 733 | 4463 | 4451 | 23.61 | **4468** | **4457.2** | 36.45 | **4468** | 4454.1 | 45.56 |
| scpcyc10 | 1708 | 11 257 | 11 248.6 | 42.86 | 11 288 | 11 243 | 83.32 | **11 292** | **11 265.4** | 69.73 |
| scpcyc11 | 3770 | 27 272 | 27 242.7 | 88.34 | 27 222 | 27 182.6 | 74.83 | **27 315** | **27 243.9** | 61.05 |
| Avg | | 1066.95 | 1066.01 | 24.06 | 1066.6 | 1064.82 | 27.74 | 1068.27 | 1066.33 | 21.06 |

**Table 11**
Summary of comparison between MMAS-ML and five variants.

| Instances | (MMAS-ML vs.) | MMAS-G | ML | MMAS-M | MMAS-L | MMAS-W |
|---|---|---|---|---|---|---|
| K90 | #Better | 9(20) | 63(73) | 70(75) | 8(16) | 18(30) |
| | #Draw | 64(48) | 12(2) | 5(0) | 66(46) | 56(41) |
| | #Worse | 2(7) | 0(0) | 0(0) | 1(13) | 1(4) |
| K95 | #Better | 7(14) | 26(63) | 72(75) | 9(13) | 16(36) |
| | #Draw | 68(49) | 49(12) | 3(0) | 65(46) | 59(38) |
| | #Worse | 0(12) | 0(0) | 0(0) | 1(16) | 0(1) |

number of elements, while MKCP focuses on covering as many elements as possible. Thus, MMAS-ML may fail to cover elements that are contained in a few sets. This phenomenon emphasizes that the design ideas of algorithms for MKCP and MSCP should be distinctive.

### 6.5. Effectiveness of the key components of MMAS-ML

To demonstrate the effectiveness of the key components in MMAS-ML, five variants of MMAS-ML were investigated:

1. **MMAS-G**: MMAS-ML without the double layer selection heuristic, that is, greedy initialization was applied.
2. **ML**: MMAS-ML without the memory-ants-based max–min ant system, that is, only the local search and the double-layer selection heuristic were integrated into the solver.
3. **MMAS-M**: MMAS-ML without any score-based local search, that is, only basic MMAS was applied.
4. **MMAS-L**: MMAS-ML without the oblivious local search.
5. **MMAS-W**: MMAS-ML with the oblivious local search.

Five algorithm variants on 150 instances under the same experimental protocol as MMAS-ML were tested. Tables 7–10 provide the detailed results on the strategy verification experiments. In these tables, #BEST is the best fitness value, #AVG is the average fitness value, and #TIME records the convergence time. Table 11 shows the statistical results of the comparison between MMAS-ML and its variants. The following observations are made based on the tables:

1. MMAS-ML outperforms MMAS-G in terms of both best (16 vs. 2) and average (34 vs. 19) results. This proves that the double-layer selection heuristic provides better initial solutions, enhancing the overall performance of MMAS-ML.
2. MMAS-ML outperforms ML in terms of both best (89 vs. 0) and average (136 vs. 0) results. This indicates that although the initial procedure can provide high-quality solutions, the local search is easily trapped in a local optimum. The

MMAS framework ensures that the diversity of the entire population is well maintained.

3. In the comparison with MMAS-M, it is observed that MMAS-ML outperforms MMAS-M. In only eight cases, MMAS-M results are identical to those of MMAS-ML. MMAS-M fails on the remaining instances. This confirms the strength of the local search.
4. As MMAS-ML applies two local search processes conjunctively. Here, only one search process is tested each time. MMAS-ML performs better than MMAS-L on #BEST metric (17 vs. 2) and achieves the same results on the #AVG metric (29 vs. 29). Moreover, MMAS-ML dominates MMAS-W on 34 instances in terms of the #BEST metric and 66 instances in terms of the #AVG metric. The results reveal that both local search and oblivious local search are necessary to balance the exploration and exploitation of MMAS-ML. The combination of these two methods yields the maximum profit overall.

From the above, it can be concluded that all of the key components of MMAS-ML play an important role in the proposed algorithm, making MMAS-ML competitive.

## 7. Conclusions and future work

In this paper, a max–min ant system with memory ants and a novel local search, namely MMAS-ML are proposed for solving MKCP. First, a double-layer selection heuristic is designed based on the scoring of columns, which enhances the selection process when ties occur. To increase the intensification capacity of each ant, a local search based on multiple scoring mechanisms and delayed configuration checking is applied accordingly. The multiple scoring mechanism refines the search direction of the local search, while delayed configuration checking helps avoid repeated searches of previous solutions. Finally, memory ants can maintain parts of the routines from the last iteration to accelerate the convergence of the construction process.

Experimental results demonstrate that MMAS-ML outperforms previous algorithms with an obvious performance enhancement. Specifically, MMAS-ML can discover a total of 16 improved best-known solutions of MKCP. Compared with the state-of-the-art algorithm HBABC, MMAS-ML is superior in 24 instances and inferior in 4 instances. On the results obtained by both algorithms, MMAS-ML outperforms HBABC on 54 instances in terms of the average metric, demonstrating its robustness. Furthermore, three key components of MMAS-ML were investigated and discussed in the form of experimental verification. Comparative results show that the double-layer selection heuristic, the memory-ant framework, and score-based local search are effective. Among them, the score-based local search is the most indispensable component; without it, the algorithm deteriorates on 142 out of 150 instances.

This work advances the solution of the maximum set k-covering problem, which plays a significant role in combinatorial optimization. The idea of distinguishing the different duties of each ant sheds light on the swarm-based algorithms from a more detailed perspective, and thus can also be considered for other optimization problems. Another interesting possibility for future work is to pursue a more efficient local search together with some reduction rules for large-scale MKCP. Although the fixed parameters of MMAS-ML were carefully tuned, the adaptive adjustment of these parameters is still an open question to be answered.

## CRediT authorship contribution statement

**Yupeng Zhou:** Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing. **Xiaofan Liu:** Methodology, Software, Investigation. **Shuli Hu:** Data curation, Resources, Formal analysis. **Yiyuan Wang:** Conceptualization, Writing – original draft, Writing – review & editing, Funding acquisition. **Minghao Yin:** Writing – original draft, Supervision, Project administration, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This work is supported by the Fundamental Research Funds for the Central Universities, China (2412020FZ030, 2412018QD022, 2412020QD008), NSFC, China under Grant No. (61806050, 61972063, 61976050, 61972384, 62106040), Jilin Science and Technology Association, China QT202005, and Jilin Provincial Science and Technology Department, China under Grant No. 20190302109GX.

## References

[1] D.S. Johnson, M.R. Garey, Computers and Intractability: A Guide to the Theory of NP-Completeness, WH Freeman, 1979.
[2] J.-F. Cordeau, F. Furini, I. Ljubić, Benders decomposition for very large scale partial set covering and maximal covering location problems, European J. Oper. Res. 275 (3) (2019) 882–896.
[3] Z. Wang, J. Liao, Q. Cao, H. Qi, Z. Wang, Achieving k-barrier coverage in hybrid directional sensor networks, IEEE Trans. Mob. Comput. 13 (7) (2013) 1443–1455.
[4] B. Saha, L. Getoor, On maximum coverage in the streaming model & application to multi-topic blog-watch, in: Proceedings of the 2009 Siam International Conference on Data Mining, SIAM, 2009, pp. 697–708.
[5] F. Chierichetti, R. Kumar, A. Tomkins, Max-cover in map-reduce, in: Proceedings of the 19th International Conference on World Wide Web, ACM, 2010, pp. 231–240.
[6] G. Lin, J. Guan, Solving maximum set k-covering problem by an adaptive binary particle swarm optimization method, Knowl.-Based Syst. 142 (2018) 95–107.
[7] G. Lin, H. Xu, X. Chen, J. Guan, An effective binary artificial bee colony algorithm for maximum set k-covering problem, Expert Syst. Appl. 161 (2020) 113717.
[8] M. Dorigo, L.M. Gambardella, Ant colonies for the travelling salesman problem, Biosystems 43 (2) (1997) 73–81.
[9] R. Jovanovic, M. Tuba, An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem, Appl. Soft Comput. 11 (8) (2011) 5360–5366.
[10] M. Kong, P. Tian, Y. Kao, A new ant colony optimization algorithm for the multidimensional knapsack problem, Comput. Oper. Res. 35 (8) (2008) 2672–2683.
[11] C. Solnon, S. Fenet, A study of ACO capabilities for solving the maximum clique problem, J. Heuristics 12 (3) (2006) 155–180.
[12] L. Lessing, I. Dumitrescu, T. Stützle, A comparison between ACO algorithms for the set covering problem, in: International Workshop on Ant Colony Optimization and Swarm Intelligence, Springer, 2004, pp. 1–12.
[13] Z.-G. Ren, Z.-R. Feng, L.-J. Ke, Z.-J. Zhang, New ideas for applying ant colony optimization to the set covering problem, Comput. Ind. Eng. 58 (4) (2010) 774–784.
[14] X. Xia, X. Peng, W. Liao, On the analysis of ant colony optimization for the maximum independent set problem, Front. Comput. Sci. 15 (4) (2021) 1–3.
[15] M. Paniri, M.B. Dowlatshahi, H. Nezamabadi-pour, MLACO: A multi-label feature selection algorithm based on ant colony optimization, Knowl.-Based Syst. 192 (2020) 105285.
[16] H. Ghimatgar, K. Kazemi, M.S. Helfroush, A. Aarabi, An improved feature selection algorithm based on graph clustering and ant colony optimization, Knowl.-Based Syst. 159 (2018) 270–285.

[17] B.C. Mohan, R. Baskaran, A survey: Ant Colony Optimization based recent research and implementation on several engineering domain, Expert Syst. Appl. 39 (4) (2012) 4618–4627.
[18] M. Dorigo, T. Stützle, Ant colony optimization: overview and recent advances, in: Handbook of Metaheuristics, Springer, 2019, pp. 311–351.
[19] T. Stützle, H.H. Hoos, MAX–MIN ant system, Future Gener. Comput. Syst. 16 (8) (2000) 889–914.
[20] S. Cai, Balance between complexity and quality: Local search for minimum vertex cover in massive graphs, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI, 2015, pp. 747–753.
[21] S. Cai, K. Su, Q. Chen, EWLS: A new local search for minimum vertex cover, in: Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI, 2010, pp. 45–50.
[22] X. Chen, Y.-S. Ong, M.-H. Lim, K.C. Tan, A multi-facet survey on memetic computation, IEEE Trans. Evol. Comput. 15 (5) (2011) 591–607.
[23] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: A literature review, Swarm Evol. Comput. 2 (2012) 1–14.
[24] J.E. Beasley, OR-Library: distributing test problems by electronic mail, J. Oper. Res. Soc. 41 (11) (1990) 1069–1072.
[25] Y. Wang, D. Ouyang, L. Zhang, M. Yin, A novel local search for unicost set covering problem using hyperedge configuration checking and weight diversity, Sci. China Inf. Sci. 60 (6) (2017) 62103.
[26] C. Gao, X. Yao, T. Weise, J. Li, An efficient local search heuristic with row weighting for the unicost set covering problem, European J. Oper. Res. 246 (3) (2015) 750–761.
[27] D.S. Hochba, Approximation algorithms for np-hard problems, ACM Sigact News 28 (2) (1997) 40–52.
[28] H. Yu, D. Yuan, Set coverage problems in a one-pass data stream, in: Proceedings of the 2013 SIAM International Conference on Data Mining, SIAM, 2013, pp. 758–766.
[29] A. McGregor, H.T. Vu, Better streaming algorithms for the maximum coverage problem, Theory Comput. Syst. 63 (7) (2019) 1595–1619.
[30] Y. Wang, D. Ouyang, M. Yin, L. Zhang, Y. Zhang, A restart local search algorithm for solving maximum set k-covering problem, Neural Comput. Appl. 29 (10) (2018) 755–765.
[31] S. Cai, K. Su, A. Sattar, Local search with edge weighting and configuration checking heuristics for minimum vertex cover, Artificial Intelligence 175 (9–10) (2011) 1672–1696.
[32] Y. Wang, M. Yin, D. Ouyang, L. Zhang, A novel local search algorithm with configuration checking and scoring mechanism for the set k-covering problem, Int. Trans. Oper. Res. 24 (6) (2017) 1463–1485.
[33] S. Cai, K. Su, Comprehensive score: Towards efficient local search for SAT with long clauses, in: Twenty-Third International Joint Conference on Artificial Intelligence, 2013.
[34] Y. Wang, C. Li, H. Sun, J. Chen, M. Yin, MLQCC: an improved local search algorithm for the set k-covering problem, ITOR 26 (3) (2019) 856–887.
[35] M. López-Ibánez, J. Dubois-Lacoste, T. Stützle, M. Birattari, The Irace Package, Iterated Race for Automatic Algorithm Configuration, Tech. rep., Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles …, 2011.
[36] H.B. Mann, D.R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Ann. Math. Stat. (1947) 50–60.