SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2019, Vol. 62 192104:1–192104:18
https://doi.org/10.1007/s11432-018-9720-6

Solving multi-scenario cardinality constrained optimization problems via multi-objective evolutionary algorithms

Xing ZHOU[1], Huaimin WANG[1], Wei PENG[1], Bo DING[1] & Rui WANG[2*]

[1]*College of Computer, National University of Defense Technology, Changsha* 410073*, China;*
[2]*College of Systems Engineering, National University of Defense Technology, Changsha* 410073*, China*

Received 29 July 2018/Revised 3 September 2018/Accepted 24 November 2018/Published online 30 July 2019

Abstract Cardinality constrained optimization problems (CCOPs) are fixed-size subset selection problems with applications in several fields. CCOPs comprising multiple scenarios, such as cardinality values that form an interval, can be defined as multi-scenario CCOPs (MSCCOPs). An MSCCOP is expected to optimize the objective value of each cardinality to support decision-making processes. When the computation is conducted using traditional optimization algorithms, an MSCCOP often requires several passes (algorithmic runs) to obtain all the (near-)optima, where each pass handles a specific cardinality. Such separate passes abandon most of the knowledge (including the potential superior solution structures) learned in one pass that can also be used to improve the results of other passes. By considering this situation, we propose a generic transformation strategy that can be referred to as the Mucard strategy, which converts an MSCCOP into a low-dimensional multi-objective optimization problem (MOP) to simultaneously obtain all the (near-)optima of the MSCCOP in a single algorithmic run. In essence, the Mucard strategy combines separate passes that deal with distinct variable spaces into a single pass, enabling knowledge reuse and knowledge interchange of each cardinality among genetic individuals. The performance of the Mucard strategy was demonstrated using two typical MSCCOPs. For a given number of evolved individuals, the Mucard strategy improved the accuracy of the obtained solutions because of the in-process knowledge than that obtained by untransformed evolutionary algorithms, while reducing the average runtime. Furthermore, the equivalence between the optimal solutions of the transformed MOP and the untransformed MSCCOP can be theoretically proved.

Keywords evolutionary computation, multi-objective optimization, cardinality-constrained optimization problem, multiple scenarios, transformation, p-median problem, portfolio optimization problem

Citation Zhou X, Wang H M, Peng W, et al. Solving multi-scenario cardinality constrained optimization problems via multi-objective evolutionary algorithms. Sci China Inf Sci, 2019, 62(9): 192104, https://doi.org/10.1007/s11432-018-9720-6

1 Introduction

In mathematics, the cardinality of a set is a measure of the "number of elements in the set". For instance, the cardinality of set $\{1, 2, 4\}$ is three, which can be denoted as $\text{card}(\{1, 2, 4\}) = 3$. The minimized single-scenario version[1)] of a cardinality constrained optimization problem (CCOP), which searches for a cardinality-specified subset that optimally minimizes a function, can be expressed as follows [1]:

$$\min \ F(x) \quad \text{s.t. } x \in \Omega(c), \tag{1}$$

* Corresponding author (email: ruiwangnudt@gmail.com)
1) The maximized version would be similar.

where c is a known and fixed cardinality value. Ω is the feasible search space that is related to the constant c. Analogously, the feasible space can be expressed as a subspace of $\text{card}(x) = c$[2]. CCOPs are often NP-hard to optimize[3].

An example of a CCOP is the p-median problem [2], which selects p sites from a site-set that minimizes the sum of the distances between any site and its nearest selected site. An application of the p-median problem is considered to be the selection of school location, which intends to minimize the overall commuting distances of students. CCOPs are commonly applied in several other areas, including ensemble learning [3], sparse principal component analysis [4], compressed sensing [5], and task allocations [6, 7].

In certain cases, the cardinality constraint of a CCOP is not single-valued but a series of values, and the CCOP must optimize the objective values of each cardinality. Such problems can be referred to as multi-scenario CCOP (MSCCOP). Lots of real-life applications embody MSCCOPs. For example, when a municipal government launches a school building project, it must consider various factors (such as the available budget and the distances of students) before making decisions. Suppose that the government hopes to calculate the optimal distances of building three, four, or five schools. After obtaining the distance information in the three cases and by referencing the budget, the government can decide the most appropriate number of schools that should be built (and their locations). The problem of obtaining and optimizing the three cases is an MSCCOP. As highlighted in this example, MSCCOPs usually play a supporting role in the decision-making process.

This study proposes MSCCOP as follows:

$$\begin{aligned}
&\text{Scenario 1: } \min\ F(x) \quad \text{s.t. } x \in \Omega(c_1),\\
&\text{Scenario 2: } \min\ F(x) \quad \text{s.t. } x \in \Omega(c_2),\\
&\qquad\qquad \vdots\\
&\text{Scenario } k\text{: } \min\ F(x) \quad \text{s.t. } x \in \Omega(c_k),
\end{aligned} \tag{2}$$

where c_i $(i = 1, \ldots, k)$ is given beforehand. Note that each scenario is already NP-hard and that different cardinality values have created completely different and disjoint variable spaces, resulting in several difficulties that are associated with handling MSCCOPs.

Traditionally, MSCCOPs are solved by optimizing the concerned scenarios one by one, subjecting each scenario to a single algorithmic run. However, this approach will be impractical when k is large. The total time cost largely depends on k and on the runtime of a single scenario. More importantly, the isolated algorithmic runs disregard the knowledge learned in one scenario which could otherwise improve the results of other scenarios. In the previous municipal-government example, the knowledge includes high (low) potential site combinations: some site combinations that produce superior (or inferior) results in one scenario are likely to breed superior (or inferior) combinations in other scenarios. Thus, these superior (inferior) site combinations should be preferred (avoided) in other scenarios during the optimization process. Instead, traditional approaches waste these in-process knowledge learned in one scenario (e.g., the selection of four sites) that could be exploited in another scenario (e.g., the selection of five sites). By considering this waste, we propose that all the scenarios can be simultaneously handled in a single run, enabling knowledge about each cardinality to be reuse and interchanged among genetic individuals of a genetic algorithm. It is promising that both the solution quality and the efficiency of the algorithm can be improved.

Multi-objective evolutionary algorithms (MOEAs), e.g., [8, 9], effectively solve the multi-objective optimization (MOO) problems by obtaining a set of Pareto optimal solutions in a single run [10]. However, if the F's in (2) are trivially combined into a k-tuple objective, the problem is not a multi-objective optimization problem (MOP) because no x can simultaneously possess multiple cardinalities and F might not be inherently conflicting with another F. The backbone of our MOEA is the well-known MOEA that can be referred to as NSGA-II even though other MOEAs can also be accepted.

2) Note that the card() function counts the number of 1-bits in the string when it is used on a binary string.

3) For additional details on CCOPs, readers can refer to http://www.zib.de/projects/cardinality-constrained-combinatorial-optimization.

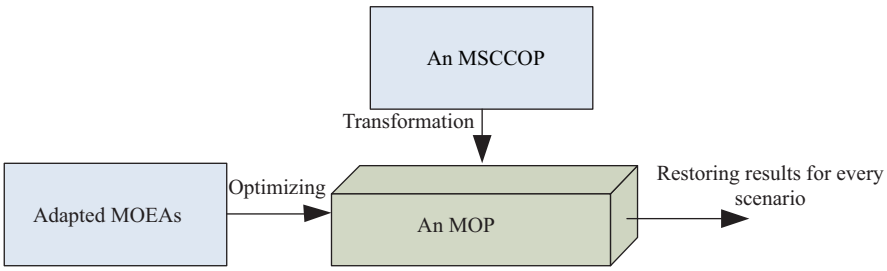

Figure 1 (Color online) Stages of solving an MSCCOP by MOEAs using the Mucard transformation.

In this paper, we propose a novel transformation strategy, namely the Mucard (Multi-scenario cardinality), to convert an MSCCOP into a low-dimensional MOP and to solve the MOP via MOEAs. The Mucard strategy enables individuals about different cardinalities to interchange the learned knowledge through genetic operations during the optimization process. Figure 1 exhibits the basic solution procedure of an MSCCOP.

In the current study, we assume that the cardinalities of an MSCCOP are presumed to compose an integer interval, i.e., $\{c_1, c_2, \ldots, c_k\} = [L, R]$, where L and R are the lower (or left) and upper (or right) bounds respectively. Denote the optimum of (1) as $F^*(x, c)$. The Mucard strategy is based on the two properties that are possessed by most of the CCOPs.

Definition 1 (Non-increasing monotonicity)**.** If $c_1 > c_2$, then $F^*(x, c_1) \leqslant F^*(x, c_2)$.

Definition 2 (Decreasing monotonicity)**.** If $c_1 > c_2$, then $F^*(x, c_1) < F^*(x, c_2)$.

For example, in the p-median problem, increasing p (i.e., increasing the number of sites selected for building new schools) reduces the optimal overall commuting distances of the students. Therefore, the p-median problem possesses the decreasing monotonicity property.

If $\mathcal{P} = \bigcup\{\text{CCOPs possessing the non-increasing monotonicity property}\}$ and $\mathcal{Q} = \bigcup\{\text{CCOPs possessing the decreasing monotonicity property}\}$, then $\mathcal{P} \supseteq \mathcal{Q}$. The present version of Mucard can handle only $\mathcal{P}$; however, we hope to extend the problem Mucard to future versions.

A multi-scenario CCOP from $\mathcal{P}$, which takes the form of (2), can be transformed into

$$\begin{cases} \min\ \text{obj1} = \text{card}(x), \\ \min\ \text{obj2} = F(x) + G(\text{card}(x)) \cdot \eta, \end{cases} \tag{3}$$

where x is the decision variable, $G(\text{card}(x))$ is the decreasing function of $\text{card}(x)$, and η is a scaling factor ($\eta \geqslant 0$) that cannot be set to zero when F is not increasing. In this study, we simply set $G(\text{card}(x)) = (L - \text{card}(x))$, although $G()$ can be set to any other form that maintains its decreasing property versus $\text{card}(x)$.

Eq. (3) works because when $\text{card}(x)$ increases, $F(x)$ is non-increasing under the assumptions of $\mathcal{P}$, and $(L - \text{card}(x)) \cdot \eta$ also decreases. Therefore, obj2 decreases with increasing obj1, which causes conflict between the two objectives obj1 and obj2. Such a conflict is considered to be a prerequisite for MOEAs.

Eq. (3) successfully blends (2) into one MOP, enabling simultaneous solving of the multiple scenarios in (2). When the individuals gain knowledge of each cardinality and interchange that knowledge in real-time during evolution, they learn a superior solution structure that potentially promotes the results of all cardinalities. The correctness of this approach will be verified and proved in later sections.

The main contributions of this study are summarized below.

• A specific transformation strategy, Mucard, is proposed (Section 4). The transformation is suitable for MSCCOPs with the non-increasing monotonicity property. The correctness of the transformation is theoretically proved and experimentally validated (Section 5) using two MSCCOPs, the multi-scenario p-median problem, and the multi-scenario cardinality-constrained portfolio optimization problem.

• A class of problems, MSCCOPs, is formally proposed and defined (Section 1). The concept of transforming MSCCOPs into low-dimensional MOPs is a new contribution to the literature.

• An adapted MOEA (also named as Mucard) is proposed for the transformed MOP. Although it is based on NSGA-II, Mucard possesses unique characteristics. Furthermore, the MOEA obtained new benchmark results in the two test problems that are mentioned above (Section 5). The advantages of the designed MOEAs are demonstrated in comprehensive experimental results.

2 Preliminaries

Mathematically, an unconstrained MOP can be expressed as

$$\min \ \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})), \tag{4}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_D) \in X \subset \mathbb{R}^D$ is the decision vector (or variable vector) and $\boldsymbol{f}(\boldsymbol{x}) \in Y \subset \mathbb{R}^M$ is the objective vector. D and M are the dimensionalities of the decision space and the objective space, respectively.

The objectives of an MOP are always conflicting [11]. Under this condition, there usually does not exist a single solution (decision vector) that simultaneously minimizes all the objective functions [11]. Instead, an MOP computes a set of tradeoff solutions (called Pareto optimal solutions). The basic definitions in MOO are given below.

Definition 3 (Pareto dominance)**.** For two decision vectors $\boldsymbol{x}_u$ and $\boldsymbol{x}_v$, $\boldsymbol{x}_u$ is said to Pareto-dominate $\boldsymbol{x}_v$ (therefore $\boldsymbol{x}_v$ is dominated by $\boldsymbol{x}_u$), denoted as $\boldsymbol{x}_u \prec \boldsymbol{x}_v$, whenever $\forall i \in \{1, \ldots, M\}$, $f_i(\boldsymbol{x}_u) \leqslant f_i(\boldsymbol{x}_v)$, and $\exists j \in \{1, \ldots, M\}$, $f_j(\boldsymbol{x}_u) < f_j(\boldsymbol{x}_v)$.

Definition 4 (Pareto optimal solutions)**.** If a decision vector $\boldsymbol{x}_u$ cannot be Pareto-dominated by any other decision vectors, $\boldsymbol{x}_u$ is a Pareto optimal solution.

Definition 5 (Pareto set)**.** The set of all Pareto optimal solutions can be referred to as the Pareto optimal set (PS).

Definition 6 (Pareto front)**.** The image of the PS can be referred to as the Pareto optimal front (PF); that is, PF=$\{\boldsymbol{f}(\boldsymbol{x})|\boldsymbol{x} \in \text{PS}\}$.

3 Related work

This section provides an overview of the existing studies that are related to CCOPs on which the MSCCOPs are based. Because general studies of the MSCCOP class are rare, we review the existing work on each MSCCOP that will be tested in the corresponding experimental sections.

Prior to reviewing the existing studies, we highlight the difference between our framework and a recently proposed concept called multi-factorial optimization (MFO) [12, 13]. MFO is a composite evolutionary paradigm that harnesses the implicit parallelism of population-based search to solve multiple, diverse optimization problems concurrently using a single population of evolving individuals. Each optimization problem can be considered to be an additional factor that influences the evolutionary process. Through transfer learning, the optimization processes properly harness the useful information between tasks/problems, aiming for effective future problem-solving. In contrast, our method adopts the multi-objectivization concept [14] and transforms a series of single-objective optimization problems into a conflicting low-dimensional MOP. The information reuse and interchange in our method occurs during the current problem-solving. An obvious advantage of our method over MFO is the automated knowledge sharing between separate problems, whereas MFO requires the design of certain mechanisms to achieve knowledge sharing. Further, it has to be mentioned that for MFO the conflicting relation required in MOO is considered to be not necessary in MFO.

While applying MOEAs to non-MOPs, we were inspired by [11,15]. The authors of [11,15] used MOEAs to simultaneously find all the roots of a non-linear equation system and all the minima of a multi-modal problem, respectively. Existing studies on CCOPs can be loosely divided into two categories, cardinality constraint-preserved optimization and cardinality constraint-eliminated optimization.

3.1 Cardinality-constraint-preserved work

Owing to the neat mathematical formulas of CCOPs, CCOPs can be tackled via mathematical programming. In a pioneering study, the authors of [16] reformulated (1) as a mixed-integer programming problem. They replaced the cardinality constraint by a surrogate constraint and solved the problem using a tailored branch-and-bound algorithm. Following this research line, researchers have developed other mathematical programming approaches, including Lagrangian relaxation. Readers can refer to [17, 18] for details. However, designing mathematical programming requires a high mastery of mathematical techniques, and the runtime of a mathematical program is unpredictable because the integer program is also NP-hard.

Many CCOPs are solved by meta-heuristic approaches instead of mathematical programming. The performances of CCOPs run using different types of meta-heuristics, including the genetic algorithm (GA), simulated annealing, Tabu search, and ant colony algorithm, have been reviewed in [19, 20]. Chang et al. [21] solved the cardinality constrained portfolio problem using three types of meta-heuristics and compared the results. However, these existing algorithms are often problem-dependent and cannot be generalized to the MSCCOP class.

3.2 Cardinality-constraint-eliminated work

If the cardinality constraint of a CCOP was eliminated or implicitly satisfied, the CCOP would be easy to solve. The cardinality constraint can be eliminated using at least two approaches, by adding slack variables and by applying penalty-based techniques.

3.2.1 *Slack variable-based techniques*

This technique was used by Volgenant [22], who solved the k-cardinality linear assignment problem (k-LAP) by assigning k (out of m) rows to k (out of n) columns in a matrix to minimize the summed costs.

Although the addition of slack variables eliminates the cardinality constraint, it is not readily generalized to obtain a simultaneous solving for multi-scenario CCOPs.

3.2.2 *Penalty-based techniques*

The cardinality constraint in (1) can also be eliminated by combining the objective function and the cardinality constraint into one criterion [20]. In (1), this combination gives

$$\min_x \ F(x) + \rho \cdot |\mathrm{card}(x) - c|, \tag{5}$$

where ρ is the penalty parameter. The drawback is the difficulty associated with controlling ρ. A large ρ will trap the search in a local optimum, whereas a small ρ risks obtaining an invalid solution x (i.e., $\mathrm{card}(x) \neq c$). In addition, the landscape of (5) is rather rugged, which increases the difficulty of searching.

4 Mucard details

4.1 Working principle

The working principle (see Appendix A) is a typical MOP evolution procedure. This procedure differs largely from the scenario-by-scenario single objective optimization procedure, which is also illustrated in Appendix A.

4.2 Proof

Subsequently, we prove the correctness of the transformation (3). Theorem 1 implies a one-to-one correspondence between an MSCCOP and its MOP through Mucard transformation[4)].

4) The developed proof for bi-objective problems can be easily extended to high-dimensional problems.

Theorem 1. Every point (o_1, o_2) on the Pareto front of (3) corresponds to the optimal result of the scenario $c_i = o_1$ (Eq. (2)). Meanwhile, every optimal result $F^*(x, c_i)$ about a scenario c_i, $i = 1, 2, \ldots, k$ has a corresponding point $(c_i, F^*(x, c_i) + (L - c_i) \cdot \eta)$ on the Pareto front.

A detailed proof is provided in Appendix B. The proof mainly exploits the non-increasing monotonicity of obj2 over obj1. We have endeavored to make the proof easy to understand.

4.3 Algorithm framework

Once the objective functions (Eq. (3)) of an MSCCOP have been determined, we can solve the transformed MOP by the adapted MOEA based on NSGA-II. The Mucard algorithm that embeds the Mucard strategy, shown in Algorithm 1, adopts the $\mu + \mu$ elitist framework [8–10].

Algorithm 1 Mucard: an MOEA algorithm for MSCCOPs.

input: The prescribed interval bounds L and R; an archive data-structure Arch to store the non-dominated individuals through the evolutionary history; the stopping criteria; the population size pop.
output: The set of PF (corresponding to $F^*(x, L), F^*(x, L + 1), \ldots, F^*(x, R)$) and the PS.
1: $P_0 \leftarrow$ generate the initial population randomly or incorporating specific knowledge of the problem;
2: Evaluate these individuals according to Eq. (3);
3: **while** stopping criteria not reached **do**
4: $P_1 \leftarrow$ selection_and_crossover_and_mutation(P_0);
5: $P_1' \leftarrow$ repair undesired individuals, and/or improve desired individuals in P_1;
6: $P_2 \leftarrow P_0 \bigcup P_1'$;
7: Insert non-dominated individuals of P_2 into Arch, and remove dominated individuals from Arch;
8: $P_0 \leftarrow$ non_dominated_sort_and_select(P_2);
9: **end while**
10: Output the restored results from the Arch.

In Algorithm 1, an individual is usually represented by a set or a binary string. Thus, each individual has cardinality, and each individual will evolve to address the scenario corresponding to that cardinality. In Steps 1 and 4, the individuals' cardinalities would better fall within the interval $[L, R]$ (for easy understanding, readers can refer to panels (b1)–(b3) in Appendix A), which will concentrate the optimization search within the interval of interest. An individual whose cardinality is outside the range $[L, R]$ can be referred to as an "undesired" individual that requires repairs. In the reproduction step (Step 4), one possible operator that produces two feasible individuals A' and B' from two already feasible parental individuals A and B, such that $\mathrm{card}(A') = \mathrm{card}(A)$ and $\mathrm{card}(B') = \mathrm{card}(B)$ can be the random respectful recombination (RRR) operator [23]. For example, suppose there are two parental individuals A_{00111} and B_{01001}. The RRR operator will produce two new individuals A'_{01011} and B'_{00011}, where $\mathrm{card}(A') = \mathrm{card}(A)$ and $\mathrm{card}(B') = \mathrm{card}(B)$. The details of RRR is in [23]. After applying the RRR operator, we can skip the repair operation in Step 5 because the cardinalities of the newly generated individuals are already within $[L, R]$ if the parents' cardinalities are within $[L, R]$.

The mutation operator in Step 4 can be implemented by simply swapping an element in an individual with an element that is not in that individual (or by exchanging two bits when the individual is encoded as a binary string). For example B'_{00011} can be mutated to B''_{10010}. Step 7 of the algorithm stores the non-dominated individuals throughout the evolving procedure in an archive. Without Arch, some good individuals may be lost from the final population because of the limited population size. The individuals in Arch are currently unused in the reproduction; however, exploiting them in future studies is an enticing prospect. Steps 6 and 8 are the classic NSGA-II steps. Step 10 restores the optimum objective value and solution of cardinality c_i from $(c_i, F^*(x, c_i) + (L - c_i) \cdot \eta)$, $i = 1, \ldots, k$.

5 Experimental studies

The accuracy and efficiency of the Mucard strategy were evaluated using two test-bed problems, the multi-scenario p-median problem and the multi-scenario cardinality constrained portfolio optimization problem (MSCCPOP, cautious not be confused with MSCCOP). Because the untransformed MSCCPOP is already

a bi-objective problem, it becomes tri-objective after the transformation. To simplify the experiments, we predetermined the lower and upper bounds of the cardinality. In real-life applications, the bounds can be acquired from decision-makers or can be computed (or estimated) out by heuristics. The present study focuses on effective solving MSCCOP and considers the determination of the cardinality bounds for conducting future studies.

5.1 Multi-scenario *p*-median problem

5.1.1 *Problem definition*

The p-median problem is a typical problem in operations research. The problem was informally introduced in the municipal government example of the previous section and can be formally stated as follows. Given a graph $G = (V, E, W)$, where W refers to the edge weights, we have to find a $V_p \subseteq V$ such that $\mathrm{card}(V_p) = p$, where p is a known constant, and that the sum of the shortest distances from the vertices in V to each one's nearest vertex in V_p is minimized, that is

$$\min \sum_{i \in V} \min_{j \in V_p} d(i, j),$$

where d is the distance function, V_p is called the selected site-set.

5.1.2 *Literature review*

The p-median problem has been extensively studied for its importance in real-life applications, such as for selecting facility locations. In 1979, Kariv [24] showed that the p-median problem is NP-hard on general graphs. Therefore, it is unlikely to find the optimal solution by using a low complexity deterministic algorithm. Hence, approximate approaches that can efficiently find high-quality solutions, such as mathematical programming, classic heuristics, and meta-heuristics, are of practical interest. Here, we review GA methods on which our method is based. Other approximate approaches are surveyed elsewhere [25–27].

Hosage and Goodchild [28] provided the first GA for the p-median problem. They noted that even though trivial GAs are unlikely to reach the efficiency of some existing heuristics, they are highly general and can be fine-tuned to maximize the computational efficiency for any specific problem classes. The GA design for p-median problems began flourishing during the 2000s. Alp et al. [29] proposed an efficient GA, which incorporates a greedy repair procedure in the reproduction of new individuals. An enhanced version of that GA was proposed in [30], and several new heuristic recombination operators for a fixed-length subset were devised and compared by Lim and Xu [31]. They concluded that a GA with the T-H-RAR operator obtains the best solutions to most of the problems. Correa et al. [32] introduced a new genetic operator that can be referred to as heuristic hypermutation, which improves the fitness of a certain percentage of genes; in [33, 34], statistical comparison of several types of the GAs was performed and compared to a variable neighborhood search meta-heuristic. In summary, the above GAs were applied to the p-median problem with a fixed p.

Two state-of-the-art approaches for p-median problems are Lagrangian relaxation and the specially tailored GA [35, 36]. The Lagrangian relaxation approach provides exact optimal solutions when the time limit is sufficiently large, whereas the special GA provides well-approximated solutions. Both the approaches have been packed into the software SITATION that will be used in our comparison study.

5.1.3 *Experimental settings*

The algorithm was tested using two datasets with p intervals. Both the datasets are well-known in the field of p-median problems and are available at the website[5)].

5) http://www.bus.ualberta.ca/eerkut/testproblems/.

Table 1 Configurations of the two algorithms compared in this study

Settings	SITATION GA	Mucard
Representation	Set	Set
Crossover	Uniform crossover	RRR operator
Mutation	Greedy interchange	Greedy interchange
Repair	Yes	No
Reproduction selection	Tournament	Tournament
Survival selection	Elitist	Crowding distance operator

• Galvão [37]. This dataset contains two graphs, Galvão100 ($|V| = 100$, $|E| = 4950$, sum$(W) = 8643$) and Galvão150 ($|V| = 150$, $|E| = 11175$, sum$(W) = 15682$). Here, sum(W) is the sum of the edge weights.

• Köerkel [38]. This dataset records a complete graph with 1000 nodes. The summed weight of the 1000×1000 matrix entries is 88500790044.

As the base comparator, we applied the SITATION software [35,36][6], which encodes several approaches such as GAs and Lagrangian relaxation. In all the testing scenarios, we compared the approximate solutions with the exact optimal solution obtained using Lagrangian relaxation. Because the time cost of Lagrangian relaxation is unpredictable (and can be unknowingly long), we imposed a cutoff time of 1500 s. In a preliminary investigation, most of the instances were satisfactorily solved within 300 s; therefore, the 1500 s cutoff time was deemed to be sufficient.

The primary goal was to compare the results of the Mucard strategy and the GA in SITATION (denoted as SITATION GA), both against the time-limited Lagrangian relaxation results. The configurations of SITATION GA and Mucard, designed to control the minimum difference of settings, are listed in Table 1. The RRR operator, which aims to produce two feasible offspring from two feasible parents, has been briefly described before. The greedy interchange operator has a complexity of $O(|V| \times p \times (|V| - p))$ [36].

5.1.4 *Results and discussion*

Because the p-median problem possesses the decreasing monotonicity property, the Mucard strategy can be used to compute the (near-)optima of different cardinalities when η in (3) is set to 0. Thus, the derived MOP is

$$\begin{aligned} &\min\ \ \mathrm{card}(x), \\ &\min\ \ \sum_{i \in V} \min_{j \in x} d(i,j), \end{aligned} \tag{6}$$

where x is the selected site-set.

In the SITATION GA, all scenarios were handled individually; the results of each scenario of 20 times were subjected to statistical analyses. In the Mucard, all the scenarios were handled simultaneously. In the present study, each algorithm was run 20 times on the three test networks. The results of the Galvão100 and Galvão150 networks are plotted in Figure 2 (the Köerkel dataset was omitted because Lagrangian relaxation in SITATION cannot solve p-median problems with more than 300 nodes but will be included in later comparisons). Here, p was varied from 1 to 20, and the population size and the maximum number of evolution generations were set to 100 and 1000, respectively, in each run. Figure 2 confirms that the Mucard algorithm yields better average results than that obtained by SITATION GA: in all the 20 runs, even the worst result of the Mucard algorithm outperformed the average SITATION GA result.

The time complexity of SITATION GA about each p is obtained by multiplying the population size, the maximum generation number, and the time complexities of reproduction operations (crossovers and mutations); thus, the time complexity of each p is $O(\text{popsize} \times \text{generation} \times (|V| + |V| \times p \times (|V| - p)))$. Further, the time complexity of the Mucard strategy is slightly increased because of the non-dominated sorting operation. However, because all scenarios of p can be solved simultaneously, the time complexity

6) http://daskin.engin.umich.edu/software/.

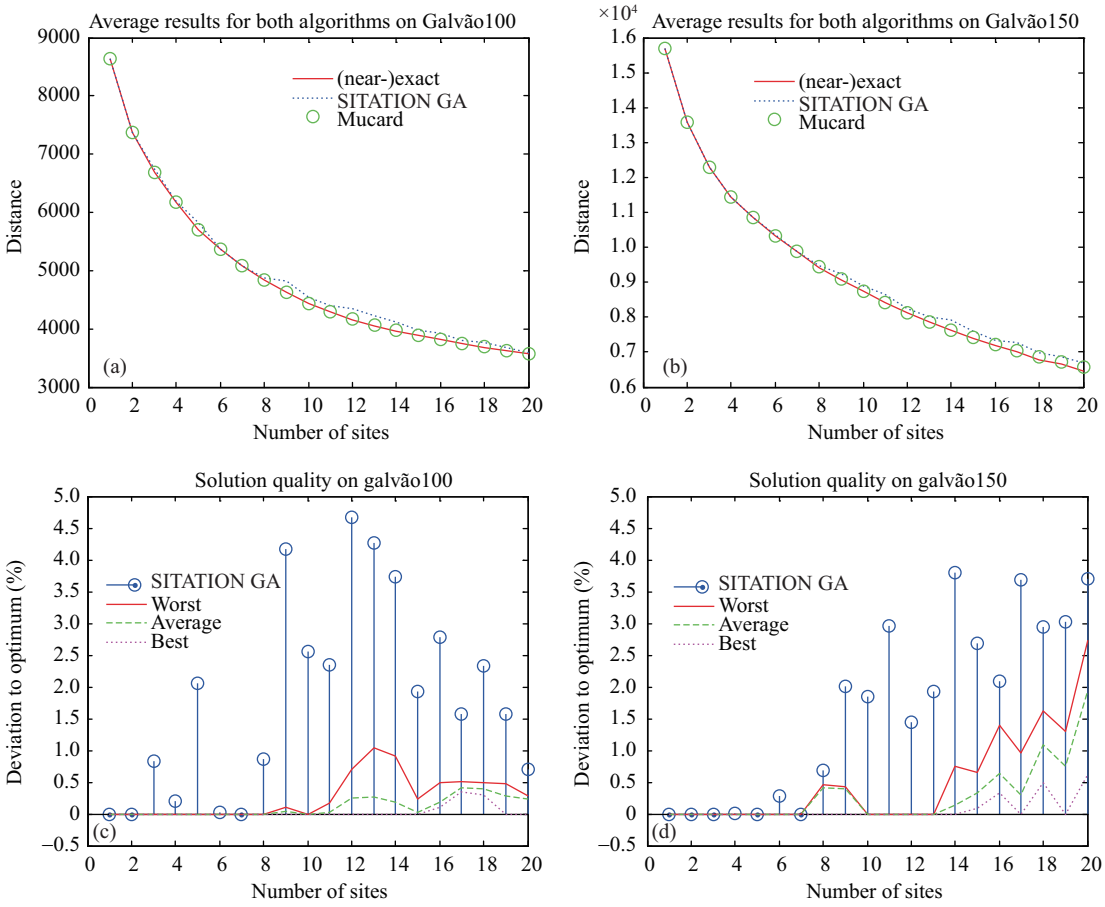

Figure 2 (Color online) Results of Mucard and SITATION GA on the Galvão100 and Galvão150 datasets. (a) and (b) average objective values of the algorithms on the two datasets. (c) and (d) average deviation percentages of SITATION GA (stems) and Mucard (lines) from the Lagrangian exact results. The red solid, green dashed, and magenta dotted lines denote the worst, average, and best results of Mucard among the 20 runs, respectively. The x-axis represents the number of chosen sites in V_p.

of the Mucard strategy is averaged among the p interval. Therefore, the time complexity of each p in Mucard strategy is $O(\text{popsize}\times\text{generation}\times(|V|+|V|\times p\times(|V|-p)+\text{popsize}\times\text{popsize}\times M))/(R-L+1)$ (where $M=2$). While evolving for the same given number of individuals and when the interval size is sufficiently large, the Mucard strategy is theoretically almost $(R-L+1)$ times faster than SITATION GA.

Figure 3 depicts the convergence behavior of Mucard. This figure traces the changes of optimal result while selecting 10 sites on each dataset. To conserve space, the changes for $p=11,\ldots,20$ are summarized as tables in Appendix C. On each network, each setting (depicted by a rounded rectangle in Figure 3) shows the averages and standard deviations of the optimal results for 20 runs for $p=10$. After obtaining the averages and standard deviations of the (population size, generation number) settings (20, 1), and (20, 100), we can trace the evolution of the optimal result for $p=10$. For example, "0.74, 0.12" indicates that the average of the optimal $p=10$ results in (20, 100) is 0.74 times that of (20, 1), and the standard deviation is 0.12 times. It means that when the problem is minimization, the optimal $p=10$ result improves (becomes smaller) as the evolution generation number increases. Therefore, increasing the number of generations from one to several hundred reduces both the average and standard deviation, with a considerable decrease in the latter. As also confirmed in Figure 3 also shows that, enlarging the population size and evolution generation changes the average after (20, 100) nominally (the "0.99"). That means the algorithm rapidly converges to the (near-)optimum solution. In the later stages of the evolution (>100 generations), the Mucard strategy mainly intended to stabilize the (near-)optimum result.

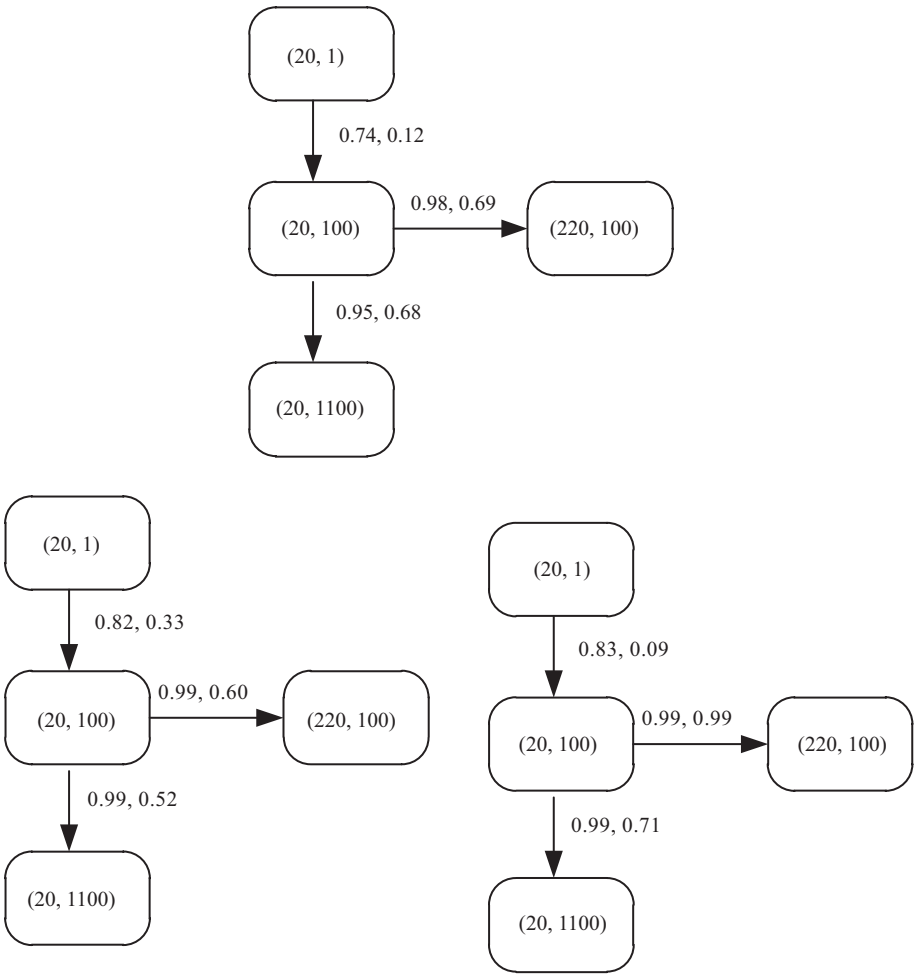

Figure 3 Statistical relations of the optimal results obtained by the Mucard strategy under different (population size, maximum generation) settings on the Köerkel, Galvão100 and Galvão150 datasets while selecting 10 sites. The numbers within parentheses denote the population size and evolution generation number, respectively. The numbers beside the arrows denote the multiplication factors of the averages and standard deviations of the results, respectively.

5.1.5 *Summary*

This subsection examines the performance (accuracy, complexity, and convergence behaviors) of the Mucard strategy on a typical MSCCOP, which can be referred to as the multi-scenario p-median problem. The main observations are as follows: (1) because it computes the optima for all p values in a single run, the Mucard strategy is more efficient than SITATION GA in average; (2) the results obtained by Mucard are much more accurate than those obtained by SITATION GA.

5.2 MSCCPOP

The portfolio optimization problem (POP) is a prevalent problem in quantitative finance. The problem devotes a fixed quantity of funding to several potential assets, thereby attempting to maximize the profit (return) and minimize the risk (variance). The POP model becomes cardinality constrained when only a specific number of assets can receive funding. To verify whether our method can effectively and efficiently solve the tri-objective problems, we verified it using an MSCCPOP.

5.2.1 *Problem definition*

The standard (i.e., cardinality unconstrained) Markowitz mean-variance (MV) model of POP traces an efficient frontier (EF, i.e., a Pareto front), which is a continuous curve that illustrates the tradeoff between

profit and risk [21]. Mathematically, the MV model can be expressed as follows:

$$\begin{aligned} &\min\ \text{risk} = \sum_{i=1}^{N}\sum_{j=1}^{N} x_i\sigma_{ij}x_j,\\ &\max\ \text{return} = \sum_{i=1}^{N} x_i\mu_i,\\ &\text{s.t.}\ \sum_{i=1}^{N} x_i = 1,\\ &\quad x_i \geqslant 0, \quad i = 1,\ldots,N, \end{aligned} \tag{7}$$

where N is the total number of assets, x_i is the quantity of funding devoted to asset i, σ_{ij} is the covariance between asset i and asset j, and μ_i is the expected return of asset i. The funding quantity $x_i,\ i = 1,\ldots,N$ is the variable. The first constraint indicates that only one unit of fund is available. The other constraints impose a zero lower bound of the fund that is devoted to each asset.

The EF is polynomial-time solvable if the matrix σ is positive semi-definite [21], which is usually the case. Thus, the frontier can be easily found by methods such as quadratic programming. The EF of (7), called "unconstrained EF", is continuous and convex.

However, the complexity of the problem increases with the number of constraints. For instance, one can fix the number of assets and impose lower- and-upper-bounded funds for each selected asset. These constraints cause NP-hardness, non-continuity, and non-convexity of the EF [39], besides reducing the efficiency of mathematical programming.

When the number of selected assets K takes a series of values, such as the values in an interval $[K_L, K_R]$, the problem becomes an MSCCPOP. One run of a bi-objective algorithm can obtain only one EF for each cardinality. To obtain all EFs for the MSCCPOP, the bi-objective algorithm requires several runs. In contrast, the Mucard strategy converts the MSCCPOP into a tri-objective problem, and the Mucard algorithm obtains all the EFs of interest in a single algorithmic run.

$$\begin{aligned} &\min\ \text{risk},\\ &\min\ \ -\text{return} + (K_L - \text{card}(y))\cdot\eta,\\ &\min\ \text{card}(y),\\ &\text{s.t.}\ \sum_{i=1}^{N} x_i = 1,\\ &\quad \epsilon_i y_i \leqslant x_i \leqslant \delta_i y_i, \quad i = 1,\ldots,N,\\ &\quad y_i \in \{0,1\}, \quad i = 1,\ldots,N, \end{aligned} \tag{8}$$

where y_i indicates whether the asset i is being funded and x_i is the quantity of funding devoted to asset i. The funding allocated to asset i is lower-bounded by ϵ_i and upper-bounded by δ_i. The variables return and risk were defined above. By convention, in this study, the second objective in (8) has been made as minimization.

An adjustment term (the term after "+") is added to the second objective in (8). The adjustment term (following the + sign in the second objective of (8)) can be altered to any form that prevents undesired Pareto dominance (as discussed in Section 4). The additional adjustment term is visually explained in Figure 4. In Figure 4(a), the two EFs (the return was negated) for $K = 2$ and $K = 3$ were separately obtained by a bi-objective algorithm. The magnified portion in Figure 4(a) reveals several equal risk-return portfolios in the two EFs; for example, (0.0048, 0.0108). Without the adjustment term, these equal portfolios for different cardinalities are not always maintained in the tri-objective algorithm because (0.0048, 0.0108, 3) is Pareto-dominated by (0.0048, −0.0108, 2). After adding an adjustment term $\eta = 0.0001$, the adjusted portfolio for (0.0048, 0.0108, 3) becomes (0.0048, 0.0109, 3), which is no longer dominated by (0.0048, 0.0108, 2) or by any other EF portfolios with cardinalities of 2 and 3. Ultimately, the adjusted (8) achieved (0.0048, 0.0108, 3) and (0.0048, 0.0108, 2) without losing either.

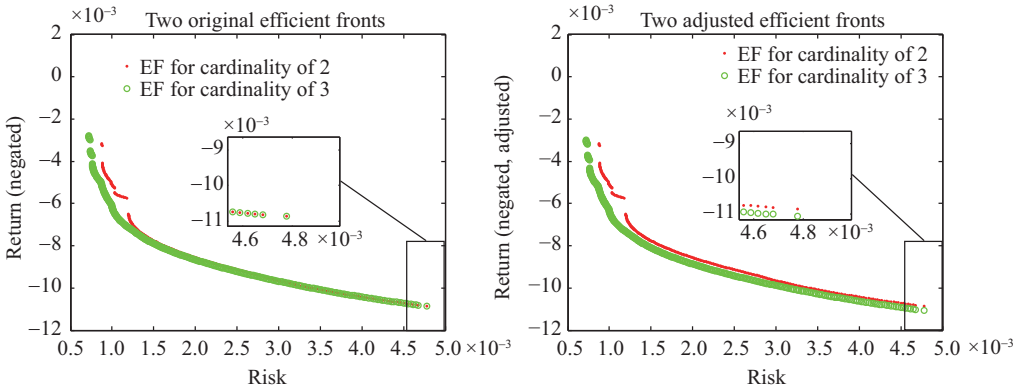

Figure 4 (Color online) Impact of the adjustment term in the proposed model. After the adjustment, the green-circled portfolios (including those in the magnified part) are no longer Pareto-dominated by the red-dotted portfolios in the three-dimensional (3D) objective space. (a) EFs for $K = 2$ and $K = 3$ without the adjustment term. Some portfolios on the two EFs are equal. (b) EFs for $K = 2$ and $K = 3$ after adding an adjustment term $\eta = 0.0001$. The originally equal portfolios no longer have dominance relations.

5.2.2 *Literature review*

As reviewed in [40, 41], a considerable amount of POP studies can be observed in the literature. The seminal model in [42] can be exactly solved by quadratic programming techniques; however, when imposed with extra constraints POP becomes difficult, and even NP-hard.

To approximate the EFs of a cardinality-constrained POP (CCPOP), one can resort to meta-heuristics such as genetic algorithms. In fact, the first POPs imposed with cardinality constraints were solved by GA and other meta-heuristics [21]. Notably, Fieldsend et al. [43] modified the cardinality constraint into an objective function, similar to our concept. However, unlike [43], the Mucard strategy creates requisite conflicts among multiple objectives. Furthermore, their model reproduces new individuals only by the mutation operator. Anagnostopoulos and Mamanis [44] also treated the cardinality constraint as an additional objective. However, when two portfolios have the same risk and return, their model simply selects the lower-cardinality portfolio without adjusting the second objective in (8). Furthermore, their algorithm is not designed for the prescribed cardinality intervals. Several other heuristics for portfolio optimization algorithms have also been proposed. For example, portfolio optimization problems with cardinality constraints have been solved using a hybrid GA and quadratic programming approach [45].

5.2.3 *Experimental settings*

Experiments were conducted on five datasets[7)], denoted as port1, port2, ..., port5. The benchmark results[8)] of these datasets are provided in [39, 46]. In the benchmark results of each dataset, the best-known risk is provided when the return and cardinality are provided. The cardinality value in the datasets does not exceed 10. As references, we also considered the results from [21].

5.2.4 *Results and discussion*

We evaluated the Mucard strategy using five datasets (port1 to port5) with $L = 1$, $R = 10$, $\eta = 1$, $\epsilon_i = 0.01$, and $\delta_i = 1$. Mucard was configured as the conventional bi-objective GA in [21]. In both Mucard and the conventional bi-objective GA [21], an individual was encoded as a set of asset indices along with their weights (related to their devotion fractions). Both the algorithms selected the parent individuals for mating by binary tournament selection and generated new individuals by a uniform crossover operator and a simple mutation operator. In both the algorithms, the number of function evaluations (equal to the number of total individuals) in each run was set to $50000N$ (where N is the number of assets), and the population size was set to 100.

7) http://people.brunel.ac.uk/~mastjjb/jeb/info.html.
8) http://host.uniroma3.it/docenti/cesarone/datasetsw3_tardella.html.

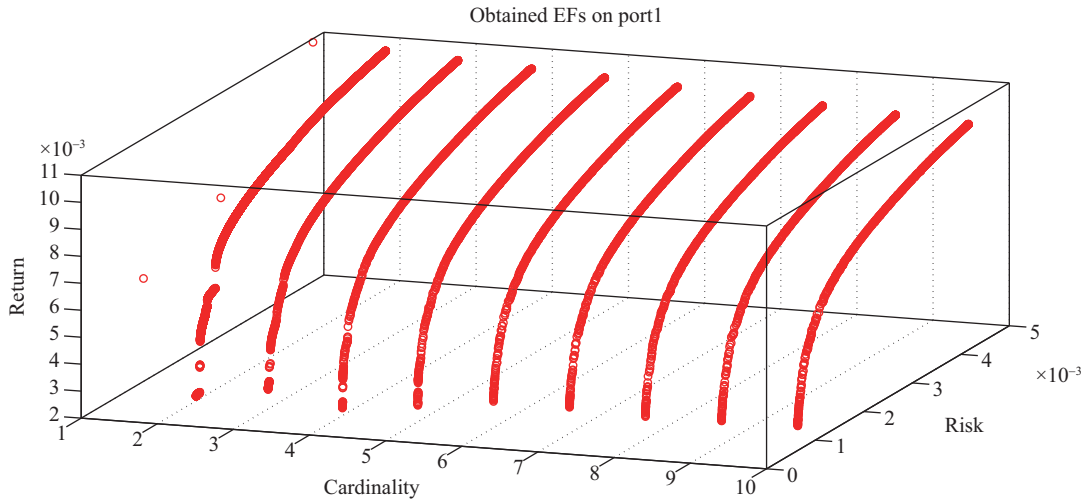

Figure 5 (Color online) EFs obtained in one run of the Mucard algorithm. Each cardinality has an EF. The EF with cardinality 1 exhibits only three portfolios.

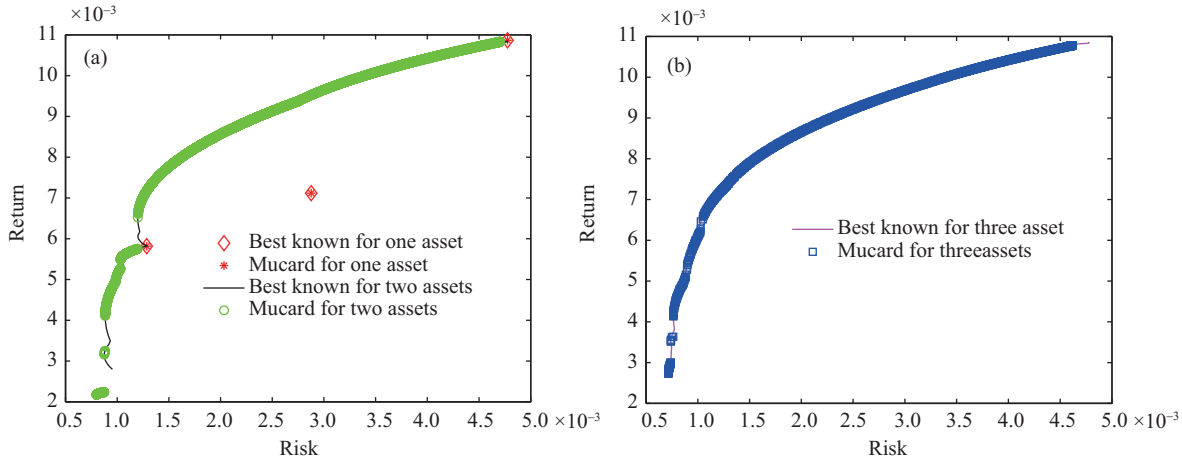

Figure 6 (Color online) Performance of Mucard. The results converge well. Furthermore, Mucard found some valid portfolios in the bottom-left regions of both plots that were not found elsewhere. The black and magenta lines plot the reference results for cardinality 2 and cardinality 3, respectively. (a) The Mucard and benchmark data when cardinality is 1 and 2; (b) the Mucard and benchmark data when cardinality is 3.

The obtained EFs on port1 in an arbitrary run are depicted in Figure 5. Mucard obtained 10 EFs in one run. Each EF comprised several discontinuous curves. To conserve space, the EFs on the other datasets are provided in the supplementary material (Appendix D).

For clarity, Figure 6 plots the port1 results with cardinalities of 1, 2, and 3, although the EFs of all cardinalities (1–10) on all five datasets were obtained. The benchmark results of Cesarone et al. [39, 46] (where the maximum return for a given risk was computed using a mathematical method) are also plotted as references.

As depicted in Figure 6, the Mucard algorithm converged closely to the non-dominated return-risk points in the optimal benchmark. In addition, it computed considerably uniform non-dominated portfolios and found some new portfolios for this problem (the lowest left regions of both the plots) which have not been previously reported in the literature, not even in the adopted benchmark data [39, 46]. The newly detected data may be explained by the inappropriate large lower bound of the return that was set in the previous benchmark's computations.

To quantify the convergence and efficiency of Mucard, we calculated the metric 'percentage deviation error' [21] of the EF with a cardinality of 10. This metric is similar to the epsilon indicator $I_\epsilon(A, B)$ [47]

Table 2 Comparison of the mean percentage deviation errors between Mucard and the algorithm in [21]. The small values in the first three rows are favorable

	Port1	Port2	Port3	Port4	Port5
Row 1	0.01097	0.02524	0.01108	0.01933	0.00796
Row 2	0.01560	0.03616	0.01680	0.03365	0.01066
Row 3	0.01095	0.02464	0.00738	0.01672	0.00504
Row 4	0.00463	0.01092	0.00573	0.01432	0.00270
Row 5	−0.00002	−0.00060	−0.00369	−0.00260	−0.00292

in evolutionary MOO. Specifically, it measures the percentage by which the portfolio (a, b, K) deviates from the actual EF of cardinality K. The 'percentage error' is the minimum of the horizontal and vertical deviations. Occasionally, a closed formula for the actual EF for a cardinality is unavailable due to discontinuity but it is described by a set of discrete points on the actual EF [39, 46]. In such cases, the percentage deviation error was calculated by interpolating these points to approximate the actual EF curve.

The mean percentage deviation errors between 20 runs of Mucard and the conventional bi-objective GA [21] are presented in Table 2. The results are broken down as follows:

• Row 1. Results for $K = 10$ computed by the GA in [21].

• Row 2. Results of Mucard after evolving $50000N$ individuals, similar to that in [21] for one cardinality. For the same number of individuals, Mucard generates non-dominated portfolios for cardinalities ranging from 1 to 10.

• Row 3. Results of Mucard after evolving $10 \times 50000N$ individuals. The average number of individuals for each cardinality was now equal to that of row 1.

• Row 4. Difference between rows 2 and 1. All results are small and positive, indicating that Mucard performs slightly worse than Chang et al.'s algorithm when one cardinality in [21] and that 10 cardinalities in Mucard, respectively, are assigned the same number of individuals [21]. However, the difference is less than 2%.

• Row 5. Difference between rows 3 and 1. The mean percentage deviation error is consistently lower in Mucard than in [21]'s algorithm [21] on these datasets, indicating that the Mucard algorithm consistently outperforms [21]'s algorithm when each cardinality is assigned the same number of individuals.

5.2.5 *Summary*

This subsection generalized the Mucard algorithm to high-dimensional MSCCOPs. Experiments confirmed that Mucard can simultaneously obtain several EFs and that it can outperform the conventional bi-objective GA in [21] in terms of mean percentage deviation error. Furthermore, Mucard discovered new valid portfolios that have not been previously reported.

6 Conclusion

MSCCOPs regularly arise in real-world applications; however, they have received very little attention. This study presents the first attempt to solve MSCCOPs effectively by an MOEA that can be referred to as the Mucard algorithm. Unlike the traditional scenario-by-scenario optimization process, the Mucard algorithm, which is embedded with a Mucard transformation strategy to MSCCOPs, can simultaneously solve all the scenarios. The transformation guarantees that each of the Pareto optimal solutions corresponds to an optimum of a scenario. The advantage is that the Mucard algorithm enables genetic individuals addressing different cardinalities to reuse and interchange their useful in-process knowledge, which can hopefully improve the quality of their solutions. The aforementioned expectations were confirmed using experimental results. For the same given number of individuals, the Mucard algorithm could obtain all the optima in a more efficient manner than the traditional method that requires multiple runs because it optimizes multiple scenarios simultaneously.

Apart from the future research ideas mentioned in the previous sections, our future work could design more effective MOEAs for MSCCOPs, which utilize the solutions that are stored in the archive. Second, because the predefined interval of the cardinality values may be difficult to estimate in practice, we should incorporate the decision-making theory into MSCCOPs. Third, the stability of Mucard algorithm in problems with many cardinality constraints deserves further attention. Finally, we highlight that solving MSCCOPs using the multi-objectivization approach is still in its infancy and should motivate a lot of interesting studies.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61751208, 61502510, 61773390), Outstanding Natural Science Foundation of Hunan Province (Grant No. 2017JJ1001), and Special Program for the Applied Basic Research of National University of Defense Technology (Grant No. ZDYYJCYJ20140601).

References

1 Stephan R. Cardinality constrained combinatorial optimization: complexity and polyhedra. Discrete Optim, 2010, 7: 99–113
2 Karp R M. Reducibility among combinatorial problems. In: Proceedings of Complexity of Computer Computations, 1972. 85–103
3 Banfield R E, Hall L O, Bowyer K W, et al. Ensemble diversity measures and their application to thinning. Inf Fusion, 2005, 6: 49–62
4 Moghaddam B, Weiss Y, Avidan S. Spectral bounds for sparse pca: exact and greedy algorithms. In: Proceedings of Advances in Neural Information Processing Systems, 2005. 915–922
5 Bruckstein A M, Donoho D L, Elad M. From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev, 2009, 51: 34–81
6 Zhou X, Huaimin W, Bo D. How many robots are enough: a multi-objective genetic algorithm for the single-objective time-limited complete coverage problem. In: Proceedings of IEEE International Conference on Robotics and Automation, 2018. 2380–2387
7 Chai R, Li H P, Meng F Y, et al. Energy consumption optimization-based joint route selection and flow allocation algorithm for software-defined networking. Sci China Inf Sci, 2017, 60: 040306
8 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput, 2002, 6: 182–197
9 Wang R, Purshouse R C, Fleming P J. Preference-inspired coevolutionary algorithms for many-objective optimization. IEEE Trans Evol Comput, 2013, 17: 474–494
10 Wang R, Zhou Z B, Ishibuchi H, et al. Localized weighted sum method for many-objective optimization. IEEE Trans Evol Comput, 2018, 22: 3–18
11 Wang Y, Li H X, Yen G G, et al. MOMMOP: multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems. IEEE Trans Cybern, 2015, 45: 830–843
12 Gupta A, Ong Y S, Feng L. Multifactorial evolution: toward evolutionary multitasking. IEEE Trans Evol Comput, 2016, 20: 343–357
13 Gupta A, Ong Y S, Feng L, et al. Multiobjective multifactorial optimization in evolutionary multitasking. IEEE Trans Cybern, 2017, 47: 1652–1665
14 Knowles J D, Watson R A, Corne D W. Reducing local optima in single-objective problems by multi-objectivization. In: Proceedings of International Conference on Evolutionary Multi-Criterion Optimization, 2001. 269–283
15 Song W, Wang Y, Li H X, et al. Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization. IEEE Trans Evol Comput, 2015, 19: 414–431
16 Bienstock D. Computational study of a family of mixed-integer quadratic programming problems. Math Program, 1996, 74: 121–140
17 Burdakov O, Kanzow C, Schwartz A. On a reformulation of mathematical programs with cardinality constraints. In: Proceedings of Advances in Global Optimization, 2015. 3–14
18 Sun X L, Zheng X J, Li D. Recent advances in mathematical programming with semi-continuous variables and cardinality constraint. J Oper Res Soc China, 2013, 1: 55–77
19 Rifki O, Ono H. A survey of computational approaches to portfolio optimization by genetic algorithms. In: Proceedings of the 18th International Conference Computing in Economics and Finance, 2012
20 Ruiz-Torrubiano R, García-Moratilla S, Suárez A. Optimization problems with cardinality constraints. In: Proceedings of Computational Intelligence in Optimization, 2010. 105–130
21 Chang T J, Meade N, Beasley J E, et al. Heuristics for cardinality constrained portfolio optimisation. Comput Oper Res, 2000, 27: 1271–1302
22 Volgenant A. Solving the k-cardinality assignment problem by transformation. Eur J Oper Res, 2004, 157: 322–331
23 Radcliffe N J, George F A. A study in set recombination. In: Proceedings of the 5th International Conference on Genetic Algorithms, 1993. 23–30
24 Kariv O, Hakimi S L. An algorithmic approach to network location problems. SIAM J Appl Math, 1979, 37: 539–560
25 Reese J. Solution methods for the p-median problem: an annotated bibliography. Networks, 2006, 48: 125–142

26 Mladenović N, Brimberg J, Hansen P, et al. The p-median problem: a survey of metaheuristic approaches. Eur J Oper Res, 2007, 179: 927–939
27 ReVelle C S, Eiselt H A, Daskin M S. A bibliography for some fundamental problem categories in discrete location science. Eur J Oper Res, 2008, 184: 817–848
28 Hosage C M, Goodchild M F. Discrete space location-allocation solutions from genetic algorithms. Ann Oper Res, 1986, 6: 35–46
29 Alp O, Erkut E, Drezner Z. An efficient genetic algorithm for the p-median problem. Ann Oper Res, 2003, 122: 21–42
30 Li X, Xiao N C, Claramunt C, et al. Initialization strategies to enhancing the performance of genetic algorithms for the p-median problem. Comput Ind Eng, 2011, 61: 1024–1034
31 Lim A, Xu Z. A fixed-length subset genetic algorithm for the p-median problem. In: Proceedings of Genetic and Evolutionary Computation Conference, 2003. 1596–1597
32 Correa E S, Steiner M T A, Freitas A A, et al. A genetic algorithm for solving a capacitated p-median problem. Numer Algorithm, 2004, 35: 373–388
33 Alba E, Domínguez E. Comparative analysis of modern optimization tools for the p-median problem. Stat Comput, 2006, 16: 251–260
34 Hansen P, Mladenović N. Complement to a comparative analysis of heuristics for the p-median problem. Stat Comput, 2008, 18: 41–46
35 Daskin M S, Maass K L. The p-median problem. In: Location Science. Berlin: Springer, 2015. 21–45
36 Daskin M S. Network and Discrete Location: Models, Algorithms, and Applications. Hoboken: John Wiley & Sons, 2013
37 Galvão R D, ReVelle C. A Lagrangean heuristic for the maximal covering location problem. Eur J Oper Res, 1996, 88: 114–123
38 Körkel M. On the exact solution of large-scale simple plant location problems. Eur J Oper Res, 1989, 39: 157–173
39 Cesarone F, Scozzari A, Tardella F. Efficient algorithms for mean-variance portfolio optimization with hard real-world constraints. In: Proceedings of the 18th AFIR Colloquium: Financial Risk in a Changing World, 2008
40 Ponsich A, Jaimes A L, Coello C A C. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. IEEE Trans Evol Comput, 2013, 17: 321–344
41 Metaxiotis K, Liagkouras K. Multiobjective evolutionary algorithms for portfolio management: a comprehensive literature review. Expert Syst Appl, 2012, 39: 11685–11698
42 Markowitz H. Portfolio selection. J Financ, 1952, 7: 77–91
43 Fieldsend J E, Matatko J, Peng M. Cardinality constrained portfolio optimisation. In: Proceedings of International Conference on Intelligent Data Engineering and Automated Learning, 2004. 788–793
44 Anagnostopoulos K P, Mamanis G. A portfolio optimization model with three objectives and discrete variables. Comput Oper Res, 2010, 37: 1285–1297
45 Ruiz-Torrubiano R, Suarez A. Hybrid approaches and dimensionality reduction for portfolio selection with cardinality constraints. IEEE Comput Intell Mag, 2010, 5: 92–107
46 Cesarone F, Scozzari A, Tardella F. A new method for mean-variance portfolio optimization with cardinality constraints. Ann Oper Res, 2013, 205: 213–234
47 Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput, 2003, 7: 117–132

Appendix A Working principle

Figure A1 illustrates the working principles of the conventional EAs and the Mucard transformation strategy on MSCCOPs. Panels (a1) and (a2) depict the actions of the conventional evolutionary algorithm (EA) on a CCOP with cardinality $k1$. First, the algorithm initializes some individuals with a cardinality that is equal to the specified value of $k1$. After the evolution, the algorithm finds one (near-)optimal solution for that cardinality. Conventional EAs solve the MSCCOP in separate passes. Panels (b1)–(b3) depict the manner in which the MSCCOP solution strategy of MOEA is embedded in the proposed transformation strategy. First, the algorithm initializes some individuals with cardinalities between L and R (panel (b1)). It further obtains a set of new individuals by reproduction (panel (b2)). After the evolution, it obtains a set of Pareto optimal individuals that are optimized to the constructed bi-objective problem (panel (b3)). Simultaneously, the optimal solutions for different CCOPs with different cardinality constraints are retrieved.

Appendix B Proof

The proof mainly exploits the non-increasing monotonicity of obj2 over obj1.

Theorem 1. Every point (o_1, o_2) on the Pareto front of (3) corresponds to an optimal result in scenario $c_i = o_1$ (Eq. (2)). Meanwhile, every optimal result $F^*(x, c_i)$ in scenario $c_i, i = 1, 2, \ldots, k$ exhibits a corresponding point $(c_i, F^*(x, c_i) + (L - c_i) \cdot \eta)$ on the Pareto front.

Proof. To prove the first statement, we note that for every point (o_1, o_2) (with an assumed decision vector of x') on the Pareto front, o_2 is minimized when obj1 is fixed at o_1; that is, no value of obj2 is smaller than o_2. Now, because $o_2 = F(x') + (L - \mathrm{card}(x')) \cdot \eta$ is minimized, $F(x')$ is also minimized because $\mathrm{card}(x') = o_1$ and the term following the

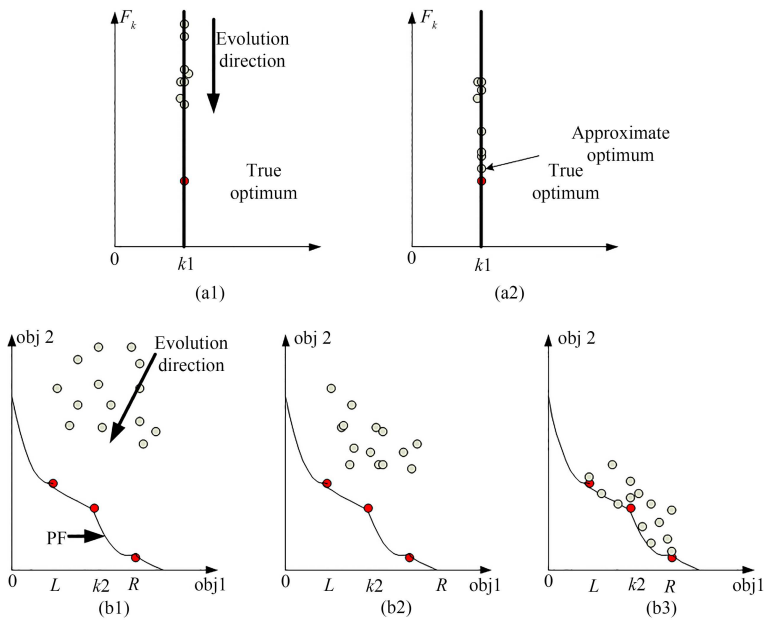

Figure A1 (Color online) Solution procedures of conventional EAs and Mucard. The red circles are the real optima; the other circles are the objective values of the EA individuals. (a1) Initialization and evolution direction of conventional EAs; (a2) final result of conventional EAs; (b1) initialization and evolution direction of Mucard; (b2) evolution of Mucard; (b3) final results of Mucard.

Table C1 Multiplication factors of the average results with different running settings and different numbers of selected sites on each dataset. The values highlighted in bold font are plotted in Figure 3

p	Köerkel			Galvão100			Galvão150		
	A	B	C	A	B	C	A	B	C
10	**0.74**	**0.98**	**0.95**	**0.82**	**0.99**	**1.00**	**0.83**	**1.00**	**1.00**
11	0.80	0.98	0.94	0.82	1.00	1.00	0.83	1.00	1.00
12	0.76	0.98	0.94	0.82	1.00	1.00	0.81	1.00	1.00
13	0.74	0.98	0.93	0.82	1.00	1.00	0.81	1.00	1.00
14	0.76	0.98	0.93	0.83	1.00	1.00	0.81	1.00	1.00
15	0.72	1.00	0.93	0.83	1.00	1.00	0.83	1.00	1.00
16	0.78	0.98	0.92	0.83	1.00	1.00	0.82	1.00	1.00
17	0.76	0.99	0.93	0.87	1.00	1.00	0.81	1.00	0.99
18	0.77	0.99	0.94	0.85	1.00	1.00	0.80	1.00	0.99
19	0.78	0.99	0.93	0.87	1.00	1.00	0.81	0.99	0.99
20	0.78	0.98	0.92	0.87	1.00	1.00	0.81	0.99	0.99

A: setting (20, 1) to setting (20, 100); B: setting (20, 100) to setting (220, 100); C: setting (20, 100) to setting (20, 1100).

+ sign is constant. Therefore, from (o_1, o_2) of the Pareto Front, we obtain the optimality function $F^*(x, o_1) = F(x') = o_2 - (L - \text{card}(x')) \cdot \eta$ in scenario o_1, and the optimal solution for scenario o_1 is also x'.

The second statement is proved by an exclusion method. Assuming that the statement is wrong, i.e., there exists c_j such that $(c_j, F^*(x, c_j) + (L - c_j) \cdot \eta)$ Pareto-dominates $(c_i, F^*(x, c_i) + (L - c_i) \cdot \eta)$, making the latter not Pareto soptimal. In the following, we demonstrate that this assumption is impossible.

• If $c_j > c_i$, then any $(c_j, F^*(x, c_j) + (L - c_j) \cdot \eta)$ will not Pareto- dominate $(c_i, F^*(x, c_i) + (L - c_i) \cdot \eta)$ once its first dimension exceeds that of the latter.

• If $c_j = c_i$ and dominance exists, the second dimension of the two points must satisfy $F^*(x, c_j) + (L - c_j) \cdot \eta < F^*(x, c_i) + (L - c_i) \cdot \eta$. Under this condition, we have $F^*(x, c_j) < F^*(x, c_i)$, which cannot be true because $F^*(x, c_i)$ is already the optimal (minimal) result of cardinality c_i.

• if $c_j < c_i$, then $F^*(x, c_j) \geqslant F^*(x, c_i)$ by the non-increasing property. We also have $(L - c_j) \cdot \eta > (L - c_i) \cdot \eta$ when $\eta > 0$. Adding the two inequalities gives $F^*(x, c_j) + (L - c_j) \cdot \eta > F^*(x, c_i) + (L - c_i) \cdot \eta$. Because the points in the second dimension do not satisfy "<" relation, point $(c_j, F^*(x, c_j) + (L - c_j) \cdot \eta)$ does not Pareto-dominate point $(c_i, F^*(x, c_i) + (L - c_i) \cdot \eta)$. Upon close examination, if a CCOP possesses a non-increasing property and η is accidentally set to 0, the Mucard algorithm cannot push the optimal results of every scenario onto the Pareto front; further, it is ambiguous whether the missing results in a certain scenario are caused by Pareto dominance or by the algorithm itself. Thus, η in (6) cannot be zero when

Table C2 Multiplication factors of the standard deviations for different running settings and different numbers of selected sites on each dataset. The values highlighted in bold are plotted in Figure 3

p	Köerkel			Galvão100			Galvão150		
	A	B	C	A	B	C	A	B	C
10	**0.12**	**0.49**	**0.68**	**0.33**	r**0.10**	**0.52**	**0.09**	**1.11**	**0.71**
11	0.08	1.31	0.41	0.13	0.10	0.41	0.07	1.12	1.13
12	0.11	0.72	0.62	0.37	0.38	0.36	0.07	0.85	0.81
13	0.07	0.87	0.53	0.18	0.14	1.02	0.13	0.77	0.64
14	0.07	0.66	0.34	0.08	0.04	0.16	0.20	0.41	0.24
15	0.12	0.87	0.40	0.06	0.22	0.06	0.07	0.93	0.69
16	0.18	0.46	0.26	0.04	0.34	0.42	0.08	0.71	1.16
17	0.16	0.96	0.58	0.03	1.52	2.40	0.24	0.51	0.56
18	0.16	0.81	0.90	0.02	2.09	1.67	0.18	0.70	0.70
19	0.22	0.82	0.61	0.03	1.10	0.86	0.15	1.21	1.01
20	0.26	0.40	0.60	0.09	0.75	0.36	0.31	0.91	0.73

A: setting (20, 1) to setting (20, 100); B: setting (20, 100) to setting (220, 100); C: setting (20, 100) to setting (20, 1100).

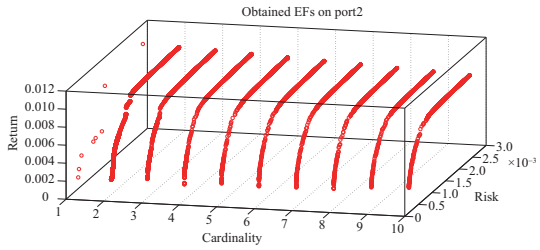

Figure D1 (Color online) EFs obtained during one run of Mucard algorithm on port2.

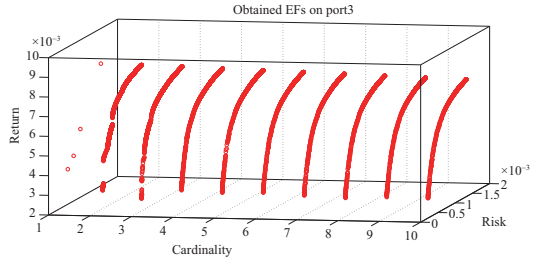

Figure D2 (Color online) EFs obtained during one run of Mucard algorithm on port3.

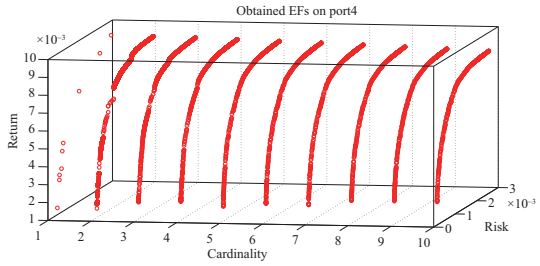

Figure D3 (Color online) EFs obtained during one run of Mucard algorithm on port4.

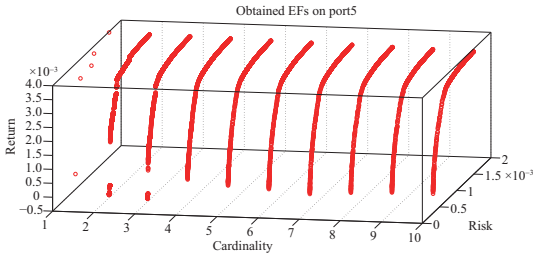

Figure D4 (Color online) EFs obtained during one run of Mucard algorithm on port5.

MSCCOP possesses the non-increasing property and when the optimum is pursued in every scenario.

From the preceding discussion, it can be concluded that no $(c_j, F^*(x, c_j)+(L-c_j)\cdot\eta)$ can Pareto-dominate $(c_i, F^*(x, c_i)+(L-c_i)\cdot\eta)$. Therefore, the second statement is true.

This statement ends the proof.

Appendix C Convergence data

Tables C1 and C2 contain statistical relations of the optimal results that are obtained by different (population size, maximum generation) settings on the Köerkel, Galvão100 and Galvão150 datasets for different numbers of selected sites.

Appendix D EFs on port2–port5

Figures D1–D4 depict the 10 EFs of interest obtained by the Mucard algorithm during one run on port2–port5.