The Set Covering Problem Revisited: An Empirical Study of the Value of Dual Information

Belma Yelbay, Ş. İlker Birbil and Kerem Bülbül
Sabancı University, Manufacturing Systems and Industrial Engineering, Orhanlı-Tuzla, 34956 Istanbul, Turkey.
byelbay@sabanciuniv.edu, sibirbil@sabanciuniv.edu, bulbul@sabanciuniv.edu

Abstract: This paper investigates the role of dual information on the performances of heuristics designed for solving the set covering problem. After solving the linear programming relaxation of the problem, the dual information is used to obtain the two main approaches proposed here: (i) The size of the original problem is reduced and then the resulting model is solved with exact methods. We demonstrate the effectiveness of this approach on a rich set of benchmark instances compiled from the literature. We conclude that set covering problems of various characteristics and sizes may reliably be solved to near optimality without resorting to custom solution methods. (ii) The dual information is embedded into an existing heuristic. This approach is demonstrated on a well-known local search based heuristic that was reported to obtain successful results on the set covering problem. Our results demonstrate that the use of dual information significantly improves the efficacy of the heuristic in terms of both solution time and accuracy.

Keywords: Heuristics, set covering, primal-dual heuristic, LP relaxation, dual information

1. Introduction. With the boost in computing technology and the striking advances in linear programming (LP) solvers, many large-scale combinatorial optimization problems can now be solved in a reasonable time. Although the performance of integer programming (IP) solvers is not comparable to that of LP solvers, many moderate-size hard IP problems in academic and industrial contexts are being solved with an increasing success every passing day. Consider for instance the famous state-of-the-art `CPLEX` solver which has recently become free for academic use. Its penultimate release 12.4 provided an average speedup of 40% on challenging instances over the immediately preceding version within a span of six months (IBM, 2013), and the current version is probably at least an order of magnitude faster compared to a decade ago. Realizing these promising developments with the exact methods, we revisit the set covering problem (SCP) and conduct an empirical study on a set of problems that appeared in the literature over the last three decades. This famous problem is defined below.

Definition 1.1 *Given a collection $\mathcal{S}$ of sets over a finite universe $\mathcal{U}$, a set cover $J \subseteq \mathcal{S}$ is a subcollection of these sets, whose union is $\mathcal{U}$. When each set in the collection has an associated cost, then the set covering problem is about finding a set cover J such that the total cost is minimized.*

Our motivation for selecting SCP is two-fold: First, SCP has a wide popularity among researchers and practitioners because a wide range of applications from scheduling to routing, and from manufacturing to telecommunications can be cast (possibly with side constraints) as set covering problems. Second, this wide interest allows us to review a large body of work from the literature as well as access many acknowledged and frequently studied set of problems. To obtain a fairly representative problem set, we have strived to compile from the literature not only the research problems but also some actual problems that arise in practice.

We need to emphasize that the need for fast and efficient heuristic methods persists especially for large-scale combinatorial problems. On this account, SCP is no different. It is fair to say that unless $\mathcal{NP} = \mathcal{P}$, there will always be hard SCP instances that are intractable with the exact methods. On one hand the competition between heuristic and exact methods has become fiercer. On the other hand, it is also known that exact and heuristic methods can be complementary to each other. Leveraging on this idea, we also propose such a complementary approach by considering the LP-IP relationship, particularly, through the use of dual information. Based on our comprehensive empirical study, we shall indeed infer that the dual information is a significant source for designing new heuristics and exact methods with excellent empirical performance. While the primal-dual heuristics for SCP have been thoroughly investigated in approximate as well as exact algorithms, we believe that their full empirical potential is yet to be realized. We also take a step in this direction.

We make the following research contributions: (i) We discuss that the optimal dual variables obtained by solving the LP relaxation of SCP bear important information about the optimal solution of SCP. To support our discussion, we give a new mixed integer linear programming model and motivate its role by solving a set of SCP instances. (ii) We show that the dual information can also be used to increase the performance of the existing local search heuristics. (iii) We support our discussion with a comprehensive computational study on a large set of SCP instances that are widely used in the literature. Our results indicate that the proposed approach performs very well in terms of solution time and quality. We find the best known or optimal solutions for most of the problem instances.

This paper is organized as follows: Section 2 summarizes the SCP literature. In Section 3, we discuss our motivation for using the dual information to efficiently solve SCP. Section 4 starts with a thorough description of the compiled set of problems. Then, the numerical results are reported in two parts. First, we report our results with an integer programming approach based on solving a restricted problem formed by the columns with zero reduced costs in the optimal LP solution. Second, we incorporate the dual information into an existing local search heuristic and compare our results with those of the original heuristic. Section 5 concludes the paper and sets forth several future research directions.

2. Literature Review. SCP is one of the oldest and most studied optimization problems in the mathematical programming literature. Many studies focus on solving SCP to optimality with exact algorithms. Exact algorithms generally rely on the branch-and-bound method to obtain optimal solutions (Balas and Carrera, 1996; Beasley, 1987; Beasley and Jornsten, 1992; Fisher and Kedia, 1990). Beasley (1987) uses subgradient optimization and a heuristic algorithm to bound the problem. Beasley and Jornsten (1992) employ the same method but improve the solution quality through Gomory f-cuts with a better branching strategy. Fisher and Kedia (1990) use a primal and a dual heuristic for bounding. Similarly, Balas and Carrera (1996) use a primal and a dual heuristic and a dynamic subgradient procedure, and iteratively improve the bounds by variable fixing.

SCP is long known to be $\mathcal{NP}$-hard in the strong sense (Garey and Johnson, 1979). Similar to the problems in the same class, many algorithms have been developed to provide approximate solutions with proven performance guarantees. Grossman and Wool (1997); Gomes et al. (2006); Vazirani (2002); Williamson (2002) list various approximation algorithms and they compare their theoretical and empirical performances. Chvatal (1979) proposes a greedy-type algorithm to approximate the SCP with a performance guarantee $\log|\mathcal{S}|$. Hochbaum (1982); Bar-Yehuda and Even (1982); Hall and Vohra (1993) propose approximation algorithms using primal and dual linear programming formulations with a performance guarantee f which is defined as the maximum number of sets that can cover an item. An inapproximability result is presented by Lund and Yannakakis (1994) which is a factor of $c\log|\mathcal{S}|$ for any $c < 1/4$ unless $NP \subseteq DTIME(|\mathcal{S}|^{poly\log|\mathcal{S}|})$. Bertsimas and Vohra (1998) propose a randomized rounding algorithm that obtains the same performance guarantee. Brönnimann and Goodrich (1995) approximate SCP by using the minimum hitting set formulation when the Vapnik-Chervonenkis dimension is d (Vapnik, 1989). Their algorithm ensures that the largest set cover is at most a factor of $O(d\log(ds))$ where s is the cardinality of optimal set cover. Then, Even et al. (2005) improve the bound with a factor of 4 relative to Brönnimann and Goodrich.

There are also heuristics sacrificing optimality but obtaining fairly good solutions within an acceptable time without performance guarantees. Caprara et al. (2000); Gomes et al. (2006); Grossman and Wool (1997) list various heuristics and approximation algorithms and show that those algorithms perform well empirically. There are several approaches to develop a heuristic algorithm. Among these, we have greedy algorithms, randomized search, heuristics based on linear programming and Lagrangian relaxations, and the closely related, primal-dual methods. The simplest algorithms are the greedy algorithms, which can be used to solve large-scale set covering problems in negligible times. However, their myopic nature may easily yield solutions far from optimality. Haouari and Chaouachi (2002), Feo and Resende (1989), as well as Vasko and Wilson (1984) introduce randomness and penalization into the greedy algorithms to improve solution quality. Along this line, three local search heuristics can be mentioned (Lan et al., 2007; Yagiura et al., 2006; Marchiori and Steenbeek, 1998). Finger et al. (2002) conduct an analysis on benchmark instances by measuring the correlation between the cost of a solution and the closeness to the optimal solution. This study gives useful insights to understand the problem structure and develop problem-specific local search algorithms. Several meta-heuristics have also been proposed for SCP. Among these, we can list simulated annealing (Brusco et al., 1999; Jacobs and Brusco, 1995), genetic algorithms (Aickelin, 2002; Beasley and Chu, 1996; Lorena and Lopes, 1997), tabu search (Caserta, 2007; Musliu, 2006; Kinney et al., 2004), ant colony optimization (Ren et al., 2010), and electromagnetism meta-heuristic (Azimi et al., 2010). In a recent study, Muter et al. (2010) devise a generic framework that uses information from the LP relaxation for promoting meta-heuristics to diversify or intensify while searching for the optimum of set covering-type optimization problems. Muter et al. also consider the role of dual information in their numerical study on the vehicle routing problem with time windows. First, they use the dual information for altering the randomized selection mechanism in the meta-heuristic. With this new mechanism, the meta-heuristic is encouraged to generate routes (sets) that are more likely to have negative reduced costs. Second, the dual information is used to reduce the size of the column pool by removing those columns with higher reduced costs. Muter et al. report that the dual information does not increase the effectiveness of their algorithms. However, in this study, we assert the contrary through a fundamentally different setting and implementation.

Similar to our approach in this paper, several studies design heuristics based on the Lagrangian relaxation or the LP relaxation of SCP (Caprara et al., 1999; Ceria et al., 1998; Hochbaum, 1982; Beasley, 1990; Umetani and Yagiura, 2007). The resulting primal-dual approach has been commonly used for approximating $\mathcal{NP}$-hard optimization problems that can be modeled as IP problems, such as the metric traveling salesman problem, the Steiner tree problem, the Steiner network problem, and the set covering problem (Vazirani, 2002). Bar-Yehuda and Even (1981) are the first researchers who have considered a generic primal-dual approach to approximate the set covering problem – later shown to be equivalent to the local ratio technique (Bar-Yehuda and Rawitz, 2005). The basis of the primal-dual approach is finding only a feasible solution to the dual of the LP relaxation of the SCP. Using this solution, an integral solution for SCP is constructed. Although the worst case performance of the primal-dual algorithm of Bar-Yehuda and Even is poor (Hall and Vohra, 1993), its empirical performance turns out to be much more promising. Therefore, several studies have sprung out of the primal-dual approach in the set covering literature (Bertsimas and Vohra, 1998; Melkonian, 2007; Williamson, 2002; Yelbay, 2010).

In the literature, there are many studies that use the dual information from the Lagrangian relaxation of SCP to reduce the size of the large-scale SCP instances by variable fixing. Reduced costs are computed for a current set of Lagrangian multipliers attained by a subgradient procedure. Beasley (1990); Ceria et al. (1998); Caprara et al. (1999); Yagiura et al. (2006) set a variable to zero, whenever its reduced cost is greater than a threshold. The dual information is also used to construct a good feasible solution. The variables with the most negative reduced costs are accepted as good candidates to obtain a feasible solution (Caserta, 2007; Caprara et al., 1999). Unlike these studies, we emphasize the importance of using the optimal dual solution of the LP relaxation of SCP. Since the LP relaxation of SCP can be solved even for very large instances amid the advances in the computational machinery, we do not need to deal with the subgradient procedure to compute the best set of Lagrangian multipliers. The attractiveness of our approach is in a way rooted in its simplicity both from a methodological and also from an implementation point of view.

3. An Overview of Primal-Dual Methods. In this section, we discuss in-depth our motivation for using the relationship between the IP formulation of the set covering model and its LP relaxation. In a nutshell, we gather dual information from the optimal solution of the LP relaxation, and then considerably reduce the problem size so that the resulting SCP can be solved by an IP solver with much less computational effort.

Before delving into the details of this approach, we first give the mathematical model of SCP. Using Definition 1.1, we obtain the integer programming model of SCP as

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in \mathcal{S}} c_j x_j, && && (1)\\ \text{subject to} \quad & \sum_{j \in \mathcal{S}} a_{ij} x_j \geq 1, && i \in \mathcal{U}, && (2)\\ & x_j \in \{0, 1\}, && j \in \mathcal{S}. && (3) \end{aligned}$$

Here $c_j > 0$ is the coverage cost of the jth set; x_j is a binary variable, which is equal to 1, if $j \in J$; a_{ij} is a binary parameter, which is equal to 1, if item i is covered by the jth set. The set of constraints (2) ensures that each item is covered by at least one set, and the constraints (3) impose integrality on the variables. If the cost of coverage is the same for each set, then the problem is called as the *unicost set covering problem*. When we consider the LP relaxation of SCP, the integrality constraints (3) are replaced by non-negativity constraints and a continuum of values is considered for the variables. The dual of the LP relaxation of SCP is then given by

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in \mathcal{U}} y_i, && && (4)\\ \text{subject to} \quad & \sum_{i \in \mathcal{U}} a_{ij} y_i \leq c_j, && j \in \mathcal{S}, && (5)\\ & y_i \geq 0, && i \in \mathcal{U}, && (6) \end{aligned}$$

where the dual variables y_i, $i \in \mathcal{U}$ correspond to the coverage constraints in the LP relaxation of (1)-(3).

As mentioned previously, our motivation is to use the LP information to obtain an integer solution for SCP. A straightforward approach is to solve the LP relaxation and then use the dual information to identify the columns with zero reduced costs. These columns can be considered as promising ones that should likely appear in the IP optimal solution. Along this line, for instance, Hochbaum (1982) solves the dual LP (4)-(6) and constructs a set cover composed of all primal variables with a zero reduced cost. Such approaches fall into the general category of primal-dual methods. Primal-dual methods find a feasible solution for the (primal) IP model (1)-(3) and a feasible solution for the dual LP model (4)-(6). In fact, the optimal dual solution can be obtained easily, since solving the LP model to optimality is not a major concern with the current status of the LP solvers. Using then elementary duality and the relation between the IP model and its LP relaxation, it is easy to see that the objective function values of a feasible IP solution and the optimal LP solution yield a pair of upper and lower bounds for SCP, respectively. Therefore, the main drive behind the primal-dual methods is to find a way to minimize the gap between the objective function value of a feasible IP solution and that of an optimal or feasible dual LP solution.

This important relationship between the IP formulation and its LP relaxation has prompted us to concentrate on the best possible result that can be obtained by a primal-dual heuristic that only adds a set to the cover, if the associated reduced cost is zero with respect to a feasible solution of (4)-(6). This consideration boils down to finding an optimal solution for the following mixed integer linear programming (MILP) model:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j \in \mathcal{S}} c_j x_j - \sum_{i \in \mathcal{U}} y_i, && && (7)\\
\text{subject to} \quad & \sum_{i \in \mathcal{U}} a_{ij} y_i + z_j = c_j, && j \in \mathcal{S} && (8)\\
& 0 \leq z_j \leq (1 - x_j) c_j, && j \in \mathcal{S}, && (9)\\
& \sum_{j \in \mathcal{S}} a_{ij} x_j \geq 1, && i \in \mathcal{U}, && (10)\\
& x_j \in \{0, 1\}, && j \in \mathcal{S}, && (11)\\
& y_i \geq 0, && i \in \mathcal{U}. && (12)
\end{aligned}
$$

In this model the sets of constraints (8) and (10) ensure that dual and primal problems are feasible, respectively, and the constraints (9) prescribe that a primal variable has a zero reduced cost when it is set to 1.

Although the MILP model (7)-(12) nicely encompasses the main idea behind most of the existing primal-dual approaches, it is important to note that solving the MILP model is much more challenging than solving the IP model (1)-(3). However, one may still wish to solve the MILP problem for small-scale instances of SCP, since an assessment of its optimal objective function value and the corresponding optimal solution may give an insight whether a further investigation of applying a primal-dual approach to a particular problem is worthwhile. To test this proposition and support the main motivation of the current study, we first solve the MILP models of a large set of SCP instances *with known optimal solutions* including the problem classes (a), (b), and (c), with the exception of the groups *scpnrg* and *scpnrh* for which the optimal solutions are not available. The details of these problem classes are presented in Section 4. There are in total 426 problems in this compilation. Figure 1 displays the distribution of the percentage gap between the sum of the dual variables $\sum_{i \in \mathcal{U}} y_i$ in the optimal solution of the MILP model (7)-(12) and the optimal objective function value of the dual LP (4)-(6). This is denoted by "LP" in the legend. Figure 1 also depicts the distribution of the percentage gap between the cost of the primal integer solution $\sum_{j \in \mathcal{S}} c_j x_j$ in the optimal solution of the MILP model and the optimal objective function value of (1)-(3). This value is denoted by "IP" in the legend.

Figure 1 has a very important implication; the feasible IP solution obtained from the proposed MILP coincides with the optimal IP solution in almost all problem instances. To be exact, 94.4% of all problems can be solved to optimality when the IP solutions from MILP are used. Furthermore, the sum of the dual variables resulting from solving the proposed MILP is equal, or very close, to the objective value of the optimal LP solution. The results obtained with the proposed MILP show that the optimal dual solution indeed yields valuable information that could help us select the optimal sets for SCP. In the subsequent discussion, we concentrate on exploiting this dual information.

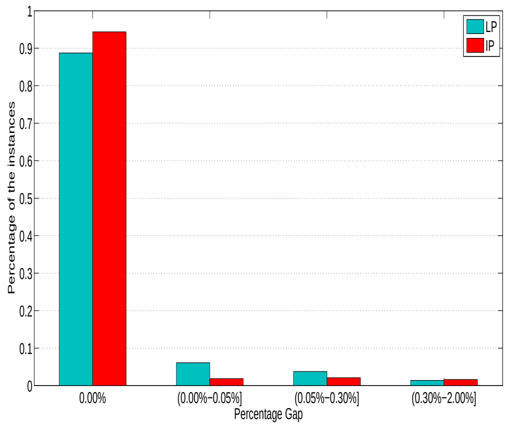


Figure 1: The distributions of the optimality gaps of the sum of the duals ("LP gap") and the primal integer solution ("IP gap") from the MILP model.

4. Computational Study. In this section, we conduct a set of experiments to support our idea that the dual information may be used to develop a mathematical programming based heuristic as well as to improve the performance of local search heuristics. We first define the problem classes and the experimental setup. Then, we present the results of a heuristic that uses the dual information to extract the most promising columns and then solves SCP optimally over those columns. Finally, we also incorporate the dual information into a well-known local search heuristic and observe that its performance indeed improves significantly.

4.1 Problem Classes and Experimental Setup. Here are the problem sets and the testing environment that we used in our empirical study. Instances not available in the OR-library (OR-lib, 2013) can be downloaded from `http://people.sabanciuniv.edu/sibirbil/scp/`.

- ◇ **(a) Standard benchmark problems from the OR-library (65 instances):** This class includes randomly generated non-unicost instances used widely in the literature (OR-lib, 2013).

- ◇ **(b) Eucledian-type cost and coverage correlated problems (320 instances):** This problem class was first introduced in (Yelbay, 2010) motivated by the presence of a set covering structure in multicast routing in wireless ad hoc networks (Oliveira and Pardalos, 2005). In this problem setting, items are points to be covered in two dimensional Euclidean space. Each point (a potential transmitter) is also associated with a set of concentric circles covering neighboring points. That is, each circle corresponds to a set in a set covering instance, and all points in the corresponding circle are covered when the set is selected. The cost of a set is typically modeled as a power function of the Euclidean distance between the center and the farthest point in the circle.
- ◇ **(c) Crew scheduling problems (16 instances):** Fourteen of these are medium-scale real-world airline crew scheduling problems from American Airlines, and two of them are bus driver scheduling problems as described in (Balas and Carrera, 1996).
- ◇ **(d) Railway problems (7 instances):** These are large-scale real-world railway crew scheduling instances from Italian railways and are available in the OR-library (OR-lib, 2013).
- ◇ **(e) Hard cost and coverage correlated problems (30 instances):** These are randomly generated instances based on the method given in (Rushmeier and Nemhauser, 2010). This method ensures that each row and column has at least two nonzero entries. The cost of a set is generated proportionally to the number of items included in the set.
- ◇ **(f) Unicost problems (21 instances):** This class includes various types of combinatorial optimization problems modeled as unicost set covering problems. Unicost problems are generally assumed to be more challenging relative to their non-unicost counterparts. The Steiner triple instances (labeled as "STS") are regarded as the toughest problem set in this class. We refer to the OR-library (OR-lib, 2013) for a more detailed description of the instances in this class.

It is fair to state that some of the problems that we include in the compilation are not as widely studied as those repeatedly used in the literature. This is in fact necessary because most of the standard benchmark problems solved in many past studies should be considered as relatively easy for the current state-of-the-art solvers. For example, the groups of instances *scp*4, *scp*5, *scp*6, *scpa*, *scpb*, *scpc*, *scpd* in problem set (a) from the highly cited OR-library can be solved to optimality within less than a second by a standard IP solver. Similarly, one group of instances in problem set (e), the group *scpe* in problem set (f), and all of the instances in problem sets (b) and (c) are relatively easy. Consequently, in the sequel we focus on and report numerical results for the remaining hard instances only.

The LP and IP solutions in this study were obtained by `ILOG IBM CPLEX 12.1` running on a personal computer with an Intel Core i5 processor and 4 GB of RAM. For all problem instances, the upper limit on the solution time is set to 7,200 seconds. The batch processing of the instances is carried out through simple C++ scripts.

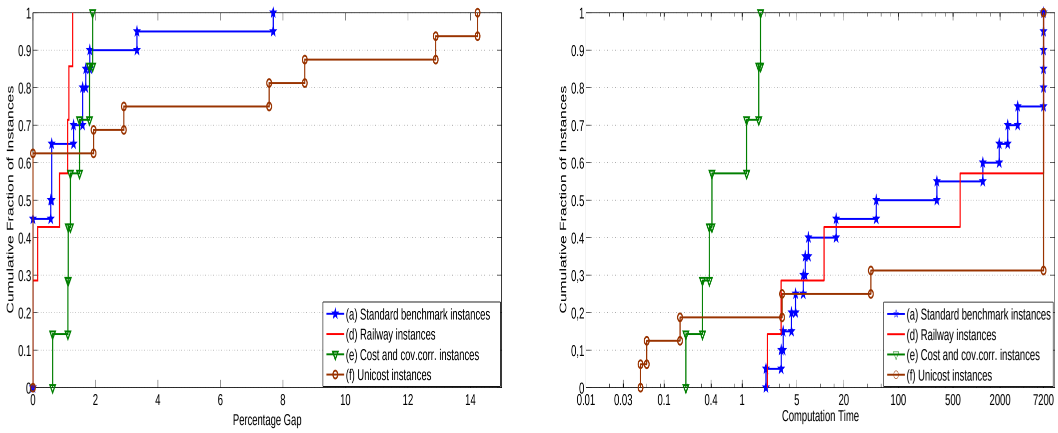
4.2 The Role of Dual Information. As we discussed in Section 3, the computational experiments obtained by solving the MILP model (7)-(12) indicate that the dual information has a potential to obtain good feasible solutions. This observation motivates us to develop a two-phase mathematical programming based heuristic that uses the dual information. In the first phase, we solve the LP relaxation of (1)-(3) and identify the most promising columns. In the second phase, we obtain an integer feasible solution to SCP by solving (1)-(3) over these columns only. We refer to this IP as the *restricted IP* or the *restricted*

SCP. One option to construct the restricted problem is to use the columns with zero reduced costs in the first phase. Even though we reduce the size of the problem with this approach, we may still end up solving a large restricted IP. This is indeed a valid concern as we observed with some of the unicost problems, for which the size of the restricted IP is identical to that of the original problem. Therefore, as an alternate method, we propose to solve a restricted IP only over the columns that are basic in the optimal solution of the LP relaxation. There are two great benefits of this approach: we can reduce the size of the restricted IP considerably and we know its exact size in advance.

The elimination of variables to obtain the restricted IP implies that the associated optimal solution may be suboptimal for SCP. Clearly, if allotted sufficient time and memory `CPLEX` can solve (1)-(3) to optimality. However, in some applications, it is not practical to wait for an optimal solution. Therefore, obtaining a good feasible solution in a reasonable time may be crucial. Yelbay et al. (2013) study such an application that is related to graph query processing on graph databases. They observed that solving the restricted IP results in better solutions than those obtained with `CPLEX` (Yelbay et al., 2013, page 9) at the time limit.

Tables 1-4 in the appendix report the computational results as well as some summary statistics. The optimal or best known objective values from the literature are provided in Column 4. The results retrieved from `CPLEX` by solving (1)-(3) are displayed in the next four columns. Observe that for the last two groups of instances *scpnrg* and *scpnrh*, `CPLEX` only returns a feasible solution when it hits the time limit of two hours. The data corresponding to the restricted SCP set up with all sets with zero reduced costs (Restricted SCP-ZeroRC) in the optimal solution of the LP relaxation of SCP are presented in Columns 9-11. Results for the restricted IP solved over the basic columns (Restricted SCP-BC) in the optimal solution of the LP relaxation of SCP follow in Columns 12-14. In these tables, "OFV" stands for "objective function value", and "T^{LP}" and "T^{IP}" denote the computation times for solving a root relaxation and the total solution time, respectively. For Restricted SCP-ZeroRC and Restricted SCP-BC, the total times reported include the effort spent while solving the LP relaxation of SCP. The gaps reported under "Gap^{IP}" are calculated with respect to the best known results from the literature. Table 1 displays the results for the hard instances among the standard benchmark problems (a). These results indicate that the restricted SCP yields integer solutions to within 1% of the best known solutions for most of the instances. There are only two instances where a single unit difference in the objective value results in a gap larger than 2% due to the small magnitude of the best known objective value. Moreover, for these problem instances there is no gain in solution quality from incorporating all columns with zero reduced costs into the restricted SCP. For the group *scpnrh*, Restricted SCP-BC even outperforms Restricted SCP-ZeroRC. Results from a set of real-world large-scale railway crew scheduling instances are summarized in Table 2. Remarkably, the percentage gaps are lower than those obtained for the standard benchmark instances. `CPLEX` does never produce an optimal solution for these instances within the time limit but terminates with feasible solutions that are competitive with the best known solutions. Restricted SCP-ZeroRC identifies the same high-quality feasible solutions in significantly less time for four out of seven instances. Table 3 depicts the results for the instances in class (e). For this problem class, the restricted SCP reduces the solution time from an average of 53.8 seconds to under 1 second with an average gap of 1.33%. Table 4 reports the results for the unicost problems (f). The structure of the unicost problems is different from that of the other problems given in our study. The number of columns is smaller than the number of rows. The computational results imply that solving the LP relaxation and selecting those columns with a zero reduced cost does not decrease the problem size. Therefore, we do not observe any noteworthy benefit of using dual information for unicost problems, except for in a few instances.

Figure 2 depicts the empirical cumulative distributions of the percentage gaps and the computation times, respectively. The percentage gaps are calculated with respect to the best known solutions in the literature as in Tables 1-4. These computational results reveal that there is only a slight difference between Restricted SCP-ZeroRC and Restricted SCP-BC. Therefore, in these figures we only report the results obtained by Restricted SCP-ZeroRC. In these plots, each point indicates the cumulative fraction of the instances with percentage gaps (or computation times) no more than the corresponding value on the $x-$axis. These figures clearly illustrate that extremely good results can be obtained when the IP formulation (1)-(3) is solved over the columns with zero reduced costs. For problem sets (a), (d), (e) with a non-unicost structure, the percentage gap does not exceed 2% except in a small fraction of the instances. The corresponding number for the unicost instances in the problem set (f) is higher. However, this different behavior is easily accounted for by observing that for these instances the size of the restricted SCP is generally identical to that of the original problem. If any reduction is present, it is small in magnitude. Consequently, we observe that the proposed mathematical programming based heuristic is more effective for non-unicost SCP instances. The relatively longer computation times observed in Figure 2(b) for some of the instances in problem classes (a), (d), and (f) can presumably be decreased at the expense of slightly larger percentage gaps.



(a) Percentage gaps (b) Computation times

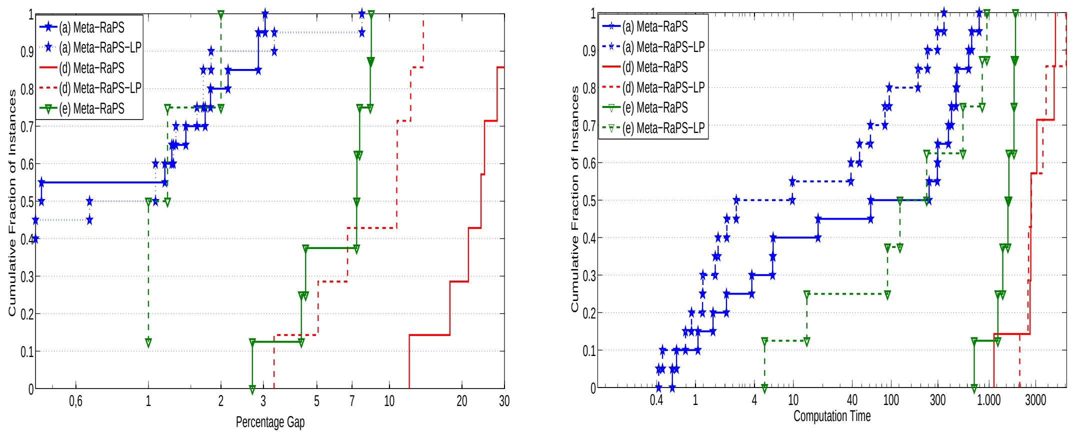
Figure 2: The empirical cumulative distribution of the percentage gaps and computation times for the restricted SCP solved over the columns with zero reduced costs.

4.3 Improving a Local Search Method with the Dual Information. In the previous section, we demonstrated that the dual information embedded in the LP relaxation of SCP is a significant tool to extract the set of promising columns in an SCP problem. In this section, we rely on the same dual information to enhance the performance of an existing SCP heuristic. Ultimately, we would like to conclude that one may leverage on the dual information to design both simple mathematical programing based heuristics (see Section 4.2) and to elicit better feasible solutions from local search algorithms. To serve this purpose, we use a randomized local search heuristic Meta-RaPS (Lan et al., 2007). Meta-RaPS consists of two phases applied several times. In the construction phase, one set is randomly selected among the set of best candidates and added to the set cover. This process is applied iteratively until all items are covered and a feasible solution to SCP is obtained. While determining the best candidates, Meta-RaPS uses one of the following four priority rules: c_j/k_j^2, c_j/k_j, $\sqrt{c_j}/k_j$ and $c_j/\sqrt{k_j}$, where k_j is the number of currently uncovered items that could be covered by set j. Each construction phase is followed by a neighborhood search phase starting from the current solution. Here, some of the sets in the current feasible solution are removed from the cover, and feasibility is restored as in the construction phase. However, in this phase the search for candidates to be inserted into the set cover is restricted to a set of "promising" columns identified during the course of the algorithm in order to speed up the computations. The authors refer to this restricted pool of columns as the "core problem" and define it as the set of columns added to the candidate list during the construction phase. Lan et al. test different versions of Meta-RaPS on the standard benchmark instances (a). We implemented the best performing one, labeled as "Meta-RaPS with randomized priority rules and core problem definition." The algorithm uses a set of parameters, such as the number of iterations performed in the construction and neighborhood search phases, the percentage of the feasible solutions that are removed during the neighborhood search, and so on. Lan et al. present the values of all relevant parameters that they use to solve the standard benchmark instances (a). We also employ the same set of parameter values in our numerical experiments. Our implementation of Meta-RaPS follows the original form described in Lan et al. (2007). However, to argue the value of dual information, we solve the problem only over the sets with zero reduced costs with respect to the optimal dual solution of the LP relaxation of SCP. Observe that this approach, referred to as Meta-RaPS-LP, is consistent with the definition of the restricted SCP in the previous section and allows us to test the value of dual information on a common ground.

Table 5 in the appendix summarizes the statistics on the percentage gaps and the solution times obtained by both Meta-RaPS and Meta-RaPS-LP on the standard benchmark instances (a). Due to the randomness inherent in the algorithms, we run them 5 times on each instance and report statistics for each instance. The computational results demonstrate that except for in a few instances Meta-RaPS-LP performs on a par with Meta-RaPS in terms of the solution quality at a much less computational effort. This observation implies that the set of columns identified by the dual information often includes at least one optimal or near-optimal solution. Meta-RaPS yields a lower minimum gap in 5 instances while the corresponding figure for Meta-RaPS-LP is 4. However, Meta-RaPS-LP beats Meta-RaPS in 8 instances in terms of the average gap. The respective number for Meta-RaPS is 4. Table 5 also summarizes the computation times of the heuristic with and without the dual information. Like Lan et al. (2007), we report the time when the best solution is encountered for the first time. As previously, the reported times for Meta-RaPS-LP include the time to solve the LP relaxation of SCP. The results clearly state that we can achieve significant computational savings. Overall, the average solution time is reduced to 70.94 seconds from 237.14 seconds if we can tolerate a minor jump in average solution quality from 0.93% to 1.17%. Moreover, observe that this difference in solution quality is mainly attributed to the two instances *scpnre*2 and *scpnrf*5. For these instances, the set of columns with zero reduced costs in the optimal LP solution lacks some crucial columns. Compare Table 1 with Table 5 to observe that the gaps of Meta-RaPS-LP for *scpnre*2 and *scpnrf*5 are identical to those of the restricted SCP for these instances. That is to say, Meta-RaPS-LP could not do any better. If we ignore *scpnre*2 and *scpnrf*5, then the average percentage gap of Meta-RaPS jumps to 1.04% while that of Meta-RaPS-LP is reduced to 0.69%.

We next test the effect of embedding the dual information on the solution quality and time for problem classes (d) and (e). In both of these problem sets, the sizes of the instances are larger relative to those in the standard benchmark set (a). Furthermore, the problem set (d) boasts some very large set covering instances. Thus, we increase the number of iterations spent in the neighborhood search to 500 from its original value of 400 and set the maximum time limit to 7,200 and 3,600 seconds for problems sets (d) and (e), respectively. Since using the dual information does not have a significant benefit for solving the unicost instances (f), we omit them from our subsequent numerical results. Tables 6-7 in the appendix summarize the results on the problem classes (d) and (e), respectively. Although the solution times attained by Meta-RaPS and Meta-RaPS-LP for the railway instances in Table 6 are close to each other, both the average and minimum percentage gaps achieved by Meta-RaPS-LP are always less than half of those obtained by Meta-RaPS (the only exception is the average percentage gap for the instance *rail*516). Finally, Table 7 presents the results for the hard cost and coverage correlated problems (e). We achieve great reductions in the computational effort expended with better solution quality for this class of problems. As indicated by a † sign, Meta-RaPS-LP finds a better solution relative to Meta-RaPS for all of the instances.

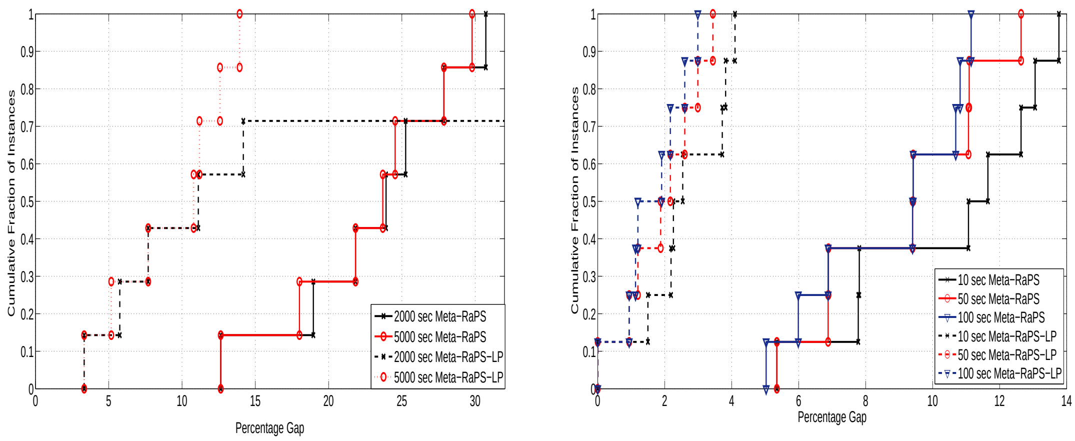
Figure 3 illustrates the information presented in Tables 5-7. It depicts the empirical cumulative distributions of the percentage gaps and solution times obtained by Meta-RaPS with and without the dual information. Figure 3(a) shows that the dual information helps to decrease the percentage gaps in almost all instances (except for some of the standard benchmark problems). We observe in Figure 3(b) that when dual information is used, the empirical cumulative distributions of the computation times shift to the left as desired. The significant improvements in the percentage gaps for the railway instances are sometimes attained at the expense of increased solution times.



(a) (b)

Figure 3: The empirical cumulative distributions of the percentage gaps and computation times of Meta-RaPS and Meta-RaPS-LP for problem classes (a), (d), and (e).

Finally, in Figure 4 we present a detailed analysis of the solution quality versus the solution time for problem sets (d) and (e). As the algorithm progresses, we take snapshots of the percentage gaps at different times. Figure 4(a) illustrates that we generally do not obtain a considerable decrease in the percentage gap as the solution time increases for the railway instances. However, the benefit of using the dual information stands out clearly when we check the percentage gaps. After 5,000 seconds, the percentage gaps are less than 25% in 70% of the instances when we apply Meta-RaPS without the dual information. Within the same duration, the maximum percentage gap decreases to 13% for the same percentage of the instances when we embed the dual information. Note that the final horizontal piece on the curve for Meta-RaPS-LP after 2,000 seconds is explained by observing that solving the LP relaxation of SCP takes more than 2,000 seconds for two out of the seven railway instances (see Table 2). Figure 4(b) summarizes the results of a similar analysis for the hard cost and coverage correlated problems. Meta-RAPS achieves a percentage gap of slightly less than 13% for 80% of all instances after 10 seconds. The percentage gaps are decreased to less than 4% again in the same percentage of all instances in 10 seconds when we use the dual information.



(a) Railway instances (b) Hard cost and coverage correlated instances

Figure 4: The progress of the percentage gaps during the course of the algorithm.

5. Conclusions and Future Research Directions. Our empirical study supports the claim that the optimal dual solution of the LP relaxation of SCP provides an important instrument for tackling this celebrated problem. By using the dual information, significant reductions in problem size and gains in solution quality and speed can be achieved for large-scale instances that are, otherwise, out of reach for off-the-shelf solvers. As our results demonstrate, there is a trade-off between incorporating all columns with zero reduced costs in the restricted SCP versus solving this IP over the basic columns only. Clearly, the former yields integer solutions of higher quality and suggests that an algorithm may benefit from visiting alternate optimal solutions of the LP relaxation of SCP. It is yet to be determined which of these multiple optimal solutions plays a more significant role in improving the IP solution. This may be an interesting path to explore for simple primal-dual heuristics as well as for more sophisticated local search methods proposed to solve SCP.

It is well-known that in many practical applications, such as vehicle routing, scheduling and so on, a large-scale SCP is solved within a branch-and-bound or a branch-and-price setting. In such a setting, the proposed approach here may be used to solve the integer programming problem formed by the columns of the restricted master problem. This approach may then give a better incumbent solution that could speed up the overall convergence of the optimal algorithm.

We also observed that a large class of standard benchmark instances for SCP can be solved very efficiently by standard exact methods. There is a clear need for gathering new problem sets for benchmarking purposes. However, we emphasize that most of the unicost problem instances remain hard for off-the-shelf solvers.

Finally, we embedded the dual information into a well known local search method and demonstrated that the dual information improves both the solution time and quality. This improvement is more apparent for large-scale SCP instances. Although we base our study on SCP, our approach of using dual information is quite generic and can be applied to other $\mathcal{NP}$-hard problems. Therefore, in the future more sophisticated algorithms may be developed that make use of the dual information as proposed in this work.

References

Aickelin, U. (2002). An indirect genetic algorithm for set covering problems. *Journal of the Operations Research*, 53(10):1118–1126.

Azimi, Z. N., Toth, P., and Galli, L. (2010). An electromagnetism metaheuristic for the unicost set covering problem. *European Journal of Operational Research*, 205:290–300.

Balas, E. and Carrera, M. C. (1996). A dynamic subgradient-based branch-and-bound procedure for set covering. *Operations Research*, 44(6):875–890.

Bar-Yehuda, R. and Even, S. (1981). A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2:198–203.

Bar-Yehuda, R. and Even, S. (1982). On approximating a vertex cover for planar graphs. In *14th ACM Symposium on Theory of Computing*, pages 303–309, San Francisco, California.

Bar-Yehuda, R. and Rawitz, D. (2005). On the equivalence between the primal-dual schema and the local ratio technique. *SIAM Journal on Discrete Mathematics*, 19:762–797.

Beasley, J. E. (1987). An algorithm for set covering problem. *European Journal of Operational Research*, 31:85–93.

Beasley, J. E. (1990). A Lagrangian heuristic for set covering problems. *Naval Research Logistics*, 37:151–164.

Beasley, J. E. and Chu, P. C. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94:392–404.

Beasley, J. E. and Jornsten, K. (1992). Enhancing an algorithm for set covering problems. *European Journal of Operational Research*, 58:293–300.

Bertsimas, D. and Vohra, R. (1998). Rounding algorithms for covering problems. *Mathematical Programming*, 80:63–89.

Brönnimann, H. and Goodrich, M. (1995). Almost optimal set covers in finite vc-dimension. *Discrete and Computational Geometry*, 14:463–479.

Brusco, M. J., Jacobs, L. W., and Thompson, G. M. (1999). A morphing procedure to supplement a simulated annealing heuristic for cost- and coverage-correlated set-covering problems. *Annals of Operations Research*, 86:611–627.

Caprara, A., Fischetti, M., and Toth, P. (1999). A heuristic method for the set covering problem. *Operations Research*, 47(5):730–743.

Caprara, A., Toth, P., and Fischetti, M. (2000). Algorithms for the set covering problem. *Annals of Operations Research*, 98:353–371.

Caserta, M. (2007). *Metaheuristics: progress in complex systems optimization*, pages 43–63. Springer, Berlin.

Ceria, S., Nobili, P., and Sassano, A. (1998). A Lagrangian-based heuristic for large-scale set covering problems. *Mathematical Programmimg*, 81:215–228.

Chvatal, V. (1979). A greedy-heuristic for the set covering problem. *Mathematics of Operations Research*, 4:233–235.

Even, G., Rawitz, D., and S.Shahar (2005). Hitting sets when the vc-dimension is small. *Information Processing Letters*, 95:358–362.

Feo, T. A. and Resende, M. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71.

Finger, M., Stützle, T., and Lourenço, H. (2002). Exploiting fitness distance correlation of set covering problems. *Lecture Notes in Computer Science*, 2279:61–71.

Fisher, M. L. and Kedia, P. (1990). Optimal solution of set covering/partitioning problems using dual heuristics. *Management Science*, 36(6):674–687.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York.

Gomes, F. C., Meneses, C. N., Pardalos, P. M., and Viana, G. V. R. (2006). Experimental analysis of approximation algorithms for the vertex cover and set covering problems. *Computers & Operations Research*, 33:3520–3534.

Grossman, T. and Wool, A. (1997). Computational experience with aproximation algorithms for the set covering problem. *European Journal of Operational Research*, 101:81–92.

Hall, N. and Vohra, R. V. (1993). Pareto optimality and a class of set covering heuristics. *Annals of Operations Research*, 43:279–284.

Haouari, M. and Chaouachi, J. S. (2002). A probabilistic greedy search algorithm for combinatorial optimization with application to the set covering problem. *Journal of the Operational Research Society*, 53:792–799.

Hochbaum, D. S. (1982). Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on Computing*, 11:555–556.

IBM (2013). `IBM ILOG CPLEX` Optimizer performance benchmarks.

Jacobs, L. W. and Brusco, M. J. (1995). A local search heuristic for large set-covering problems. *Naval Research Logistics*, 42:1129–1140.

Kinney, G., Barnes, J. W., and Colleti, B. (2004). A group theoretic tabu search algorithm for set covering problems. Technical report, Graduate Program in Operations Research and Industrial Engineering, The University of Texas, Austin, Texas 78712.

Lan, G., DePuy, G. W., and Whitehouse, G. E. (2007). An effective and simple heuristic for the set covering problem. *European Journal of Operational Research*, 176:1387–1403.

Lorena, L. A. N. and Lopes, L. S. (1997). Genetic algorithms applied to computationally difficult set covering problems. *Journal of Operational Research Society*, 48:440–445.

Lund, C. and Yannakakis, M. (1994). On the hardness of approximating minimization problems. *Journal of ACM*, 5:960–981.

Marchiori, E. and Steenbeek, A. (1998). An iterated heuristic algorithm for the set covering problem. In *Proceedings WAE'98 Saarbrücken*, pages 1–3.

Melkonian, V. (2007). New primal-dual algorithms for steiner tree problems. *Computers & Operations Research*, 34:2147–2167.

Musliu, N. (2006). Local search algorithm for unicost set covering problem. *Lecture Notes in Artificial Intelligence*, 4031:302–311.

Muter, I., Birbil, S. I., and Sahin, G. (2010). Combination of metaheuristic and exact algorithms for solving set covering-type optimization problems. *INFORMS Journal on Computing*, 22:603–619.

Oliveira, C. A. and Pardalos, P. M. (2005). A survey of combinatorial optimization problems in multicast routing. *Computers & Operations Research*, 32(8):1953–1981.

OR-lib (2013). http://people.brunel.ac.uk/~mastjjb/jeb/info.html.

Ren, Z. G., Feng, Z. R., Ke, L. J., and Zhang, Z. J. (2010). New ideas for applying ant colony optimization to the set covering problem. *Computers & Industrial Engineering*, 58:774–784.

Rushmeier, R. A. and Nemhauser, G. L. (2010). Experiments with parallel branch-and-bound algorithms for the set covering problem. *Computers & Industrial Engineering*, 58:774–784.

Umetani, S. and Yagiura, M. (2007). Relaxation heuristics for the set covering problem. *Journal of the Operations Research Society of Japan*, 50:350–375.

Vapnik, V. (1989). *Statistical Learning Theory*. Wiley-Interscience.

Vasko, F. J. and Wilson, G. R. (1984). An efficient heuristic for large set covering problems. *Naval Research Logistics Quarterly*, 31:163–171.

Vazirani, V. V. (2002). *Theoretical aspects of computer science*, pages 198–207. Springer, Verlag LNCS 2292.

Williamson, D. P. (2002). The primal-dual method for approximation algorithms. *Mathematical Programming*, 91:447–478.

Yagiura, M., Kishida, M., and Ibaraki, T. (2006). A 3-flip neighborhood local search for the set covering problem. *European Journal of Operational Research*, 172:472–499.

Yelbay, B. (2010). Primal-dual heuristics for solving the set covering problem. Master's thesis, Sabancı University, Istanbul,Turkey.

Yelbay, B., Birbil, S., Bülbül, K., and Jamil, H. (2013). Trade-offs computing minimum hub cover toward optimized graph query processing. http://arxiv.org/abs/1311.1626.

Appendix A. Tables that are referred to in Sections 4.2 and 4.3

Table 1: Performance statistics for the restricted SCP on standard benchmark problems (a).

				CPLEX				Restricted SCP-ZeroRC			Restricted SCP-BC		
Instance	$\|\mathcal{U}\|$	$\|\mathcal{S}\|$	**OFV**	**OFV**	T^{IP}	T^{LP}	**Gap$^{\mathrm{IP}}$ (%)**	**OFV**	T^{IP}	**Gap$^{\mathrm{IP}}$ (%)**	**OFV**	T^{IP}	**Gap$^{\mathrm{IP}}$ (%)**
scpnre1	500	5,000	29*	29	38.91	0.28	0.00	29	6.05	0.00	29	6.08	0.00
scpnre2	500	5,000	30*	30	53.78	0.31	0.00	31	15.95	3.33	31	19.33	3.33
scpnre3	500	5,000	27*	27	16.81	0.30	0.00	27	4.83	0.00	27	4.03	0.00
scpnre4	500	5,000	28*	28	38.03	0.33	0.00	28	6.41	0.00	28	10.45	0.00
scpnre5	500	5,000	28*	28	15.83	0.28	0.00	28	3.34	0.00	28	2.56	0.00
			Average		32.67	0.30	0.00		7.32	0.67		8.49	0.67
scpnrf1	500	5,000	14*	14	35.94	0.47	0.00	14	4.28	0.00	14	4.02	0.00
scpnrf2	500	5,000	15*	15	13.70	0.51	0.00	15	3.16	0.00	15	3.28	0.00
scpnrf3	500	5,000	14*	14	7.77	0.42	0.00	14	2.01	0.00	14	2.34	0.00
scpnrf4	500	5,000	14*	14	29.55	0.48	0.00	14	7.05	0.00	14	5.94	0.00
scpnrf5	500	5,000	13*	13	28.88	0.48	0.00	14	51.98	7.69	14	77.97	7.69
			Average		23.17	0.47	0.00		13.70	1.54		18.71	1.54
scpnrg1	1,000	10,000	176†	177$^\bullet$	$\diamond$	0.89	0.57	177	1952.42	0.57	177	715.53	0.57
scpnrg2	1,000	10,000	154†	156$^\bullet$	$\diamond$	0.83	1.30	156	310.48	1.30	156	217.44	1.30
scpnrg3	1,000	10,000	166†	167$^\bullet$	$\diamond$	0.89	0.60	167	2505.92	0.60	167	2874.75	0.60
scpnrg4	1,000	10,000	168†	169$^\bullet$	$\diamond$	1.05	0.60	169	1203.61	0.60	169	2697.56	0.60
scpnrg5	1,000	10,000	168†	169$^\bullet$	$\diamond$	1.06	0.60	169	3362.30	0.60	169	4608.25	0.60
			Average			0.94	0.73		1866.95	0.73		2222.71	0.73
scpnrh1	1,000	10,000	63†	64$^\bullet$	$\diamond$	1.20	1.59	64$^\bullet$	$\diamond$	1.59	63$^\bullet$	$\diamond$	0.00
scpnrh2	1,000	10,000	63†	64$^\bullet$	$\diamond$	1.42	1.59	64$^\bullet$	$\diamond$	1.59	64$^\bullet$	$\diamond$	1.59
scpnrh3	1,000	10,000	59†	60$^\bullet$	$\diamond$	1.36	1.69	60$^\bullet$	$\diamond$	1.69	59$^\bullet$	$\diamond$	0.00
scpnrh4	1,000	10,000	58†	58$^\bullet$	$\diamond$	1.16	0.00	58$^\bullet$	$\diamond$	0.00	58$^\bullet$	$\diamond$	0.00
scpnrh5	1,000	10,000	55†	56$^\bullet$	$\diamond$	1.14	1.82	56$^\bullet$	$\diamond$	1.82	56$^\bullet$	$\diamond$	1.82
			Average			1.26	1.34			1.34			0.68

$\dagger$: Best known objective function value in the literature.

$\diamond$: Terminated due to time limit.

$\star$: Optimal objective function value.

$\bullet$: Incumbent solution.

Table 2: Performance statistics for the restricted SCP on railway problems (d).

				CPLEX				Restricted SCP-ZeroRC			Restricted SCP-BC		
Instance	$\|\mathcal{U}\|$	$\|\mathcal{S}\|$	**OFV**	**OFV**	$T^{\mathbf{IP}}$	$T^{\mathbf{LP}}$	$\mathbf{Gap}^{\mathbf{IP}}$ (%)	**OFV**	$T^{\mathbf{IP}}$	$\mathbf{Gap}^{\mathbf{IP}}$ (%)	**OFV**	$T^{\mathbf{IP}}$	$\mathbf{Gap}^{\mathbf{IP}}$ (%)
rail507	507	63,009	174†	176$^\bullet$	$\diamond$	4.20	1.15	176	11.16	1.15	176	5.42	1.15
rail516	516	47,311	182*	182	$\diamond$	1.59	0.00	182	2.11	0.00	183	1.78	0.55
rail582	582	55,515	211*	211	$\diamond$	2.66	0.00	211	3.14	0.00	212	2.89	0.47
rail2536	2,536	1,081,841	689*	690$^\bullet$	$\diamond$	468.09	0.15	690	615.68	0.15	696	2776.25	1.02
rail2586	2,586	920,683	945†	957$^\bullet$	$\diamond$	884.52	1.27	957$^\bullet$	$\diamond$	1.27	963$^\bullet$	$\diamond$	1.90
rail4284	4,284	1,092,610	1064†	1073$^\bullet$	$\diamond$	5000.52	0.85	1073$^\bullet$	$\diamond$	0.85	1081$^\bullet$	$\diamond$	1.60
rail4872	4,872	968,672	1528†	1545$^\bullet$	$\diamond$	3917.51	1.11	1545$^\bullet$	$\diamond$	1.11	1554$^\bullet$	$\diamond$	1.70
			Average			1468.44	0.65			0.65			1.20

$\dagger$: Best known objective function value in the literature.

$\diamond$: Terminated due to time limit.

$\star$: Optimal objective function value.

$\bullet$: Incumbent solution.

Table 3: Performance statistics for the restricted SCP on hard cost and coverage correlated problems (e).

				CPLEX				Restricted SCP-ZeroRC			Restricted SCP-BC		
Instance	$\|\mathcal{U}\|$	$\|\mathcal{S}\|$	**OFV**	**OFV**	$T^{\mathbf{IP}}$	$T^{\mathbf{LP}}$	$\mathbf{Gap}^{\mathbf{IP}}$ (%)	**OFV**	$T^{\mathbf{IP}}$	$\mathbf{Gap}^{\mathbf{IP}}$ (%)	**OFV**	$T^{\mathbf{IP}}$	$\mathbf{Gap}^{\mathbf{IP}}$ (%)
rand16	200	2,000	276*	276	28.30	0.08	0.00	281	0.41	1.81	281	0.41	1.81
rand17	200	2,000	269*	269	42.55	0.06	0.00	273	1.14	1.49	273	1.61	1.49
rand18	200	2,000	262*	262	74.19	0.08	0.00	267	1.72	1.91	267	2.03	1.91
rand19	200	2,000	266*	266	36.36	0.08	0.00	269	0.38	1.13	269	0.44	1.13
rand20	200	2,000	268*	268	185.77	0.08	0.00	271	1.63	1.12	271	1.78	1.12
rand22	200	2,000	334*	334	2.64	0.06	0.00	338	0.19	1.20	338	0.22	1.20
rand24	200	2,000	320*	320	6.97	0.05	0.00	322	0.31	0.63	322	0.41	0.63
			Average		53.82	0.07	0.00		0.82	1.33		0.98	1.33

$\star$: Optimal objective function value.

Table 4: Performance statistics for the restricted SCP on unicost problems (f).

				CPLEX				**Restricted SCP-ZeroRC**			**Restricted SCP-BC**		
Instance	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{S}\rvert$	**OFV**	**OFV**	$T^{\mathbf{IP}}$	$T^{\mathbf{LP}}$	**Gap$^{\mathbf{IP}}$** (%)	**OFV**	$T^{\mathbf{IP}}$	**Gap$^{\mathbf{IP}}$** (%)	**OFV**	$T^{\mathbf{IP}}$	**Gap$^{\mathbf{IP}}$** (%)
scpcyc06	240	192	$60^\dagger$	$60^\bullet$	$\diamond$	0.03	0.00	$60^\bullet$	$\diamond$	0.00	60	1843.28	0.00
scpcyc07	672	448	$144^\dagger$	$144^\bullet$	$\diamond$	0.14	0.00	$144^\bullet$	$\diamond$	0.00	$150^\bullet$	$\diamond$	4.17
scpcyc08	1,792	1,024	$344^\dagger$	$354^\bullet$	$\diamond$	0.45	2.91	$354^\bullet$	$\diamond$	2.91	$357^\bullet$	$\diamond$	3.78
scpcyc09	4,608	2,304	$780^\dagger$	$839^\bullet$	$\diamond$	4.06	7.56	$839^\bullet$	$\diamond$	7.56	$862^\bullet$	$\diamond$	10.51
scpcyc10	11,520	5,120	$1{,}792^\dagger$	$2{,}047^\bullet$	$\diamond$	53.27	14.23	$2{,}047^\bullet$	$\diamond$	14.23	$2{,}058^\bullet$	$\diamond$	14.84
scpcyc11	28,160	11,264	$4{,}103^\dagger$	$4{,}632^\bullet$	$\diamond$	928.34	12.89	$4{,}632^\bullet$	$\diamond$	12.89	$4{,}719^\bullet$	$\diamond$	15.01
scpclr10	511	210	$25^\dagger$	25	44.28	0.06	0.00	25	44.34	0.00	25	8.77	0.00
scpclr11	1,023	330	$23^\dagger$	$23^\bullet$	$\diamond$	0.31	0.00	$23^\bullet$	$\diamond$	0.00	$23^\bullet$	$\diamond$	0.00
scpclr12	2,047	495	$23^\dagger$	$23^\bullet$	$\diamond$	2.38	0.00	$23^\bullet$	$\diamond$	0.00	$23^\bullet$	$\diamond$	0.00
scpclr13	4,095	715	$23^\dagger$	$25^\bullet$	$\diamond$	8.25	8.70	$25^\bullet$	$\diamond$	8.70	$23^\bullet$	$\diamond$	0.00
STS9	12	9	$5^\star$	5	0.03	0.03	0.00	5	0.06	0.00	5	0.06	0.00
STS15	35	15	$9^\star$	9	0.03	0.02	0.00	9	0.05	0.00	9	0.05	0.00
STS27	117	27	$18^\star$	18	0.14	0.02	0.00	18	0.16	0.00	18	0.16	0.00
STS45	330	45	$30^\star$	30	3.22	0.02	0.00	30	3.24	0.00	30	3.24	0.00
STS81	1,080	81	$61^\dagger$	$61^\bullet$	$\diamond$	0.17	0.00	$61^\bullet$	$\diamond$	0.00	$61^\bullet$	$\diamond$	0.00
STS135	3,015	135	$103^\dagger$	$105^\bullet$	$\diamond$	0.06	1.94	$105^\bullet$	$\diamond$	1.94	$105^\bullet$	$\diamond$	1.94
			Average			62.35	2.83			2.83			3.14

$\dagger$: Best known objective function value.
$\diamond$: Terminated due to time limit.
$\star$: Optimal objective function value.
$\bullet$: Incumbent solution.

Table 5: Benchmarking Meta-RaPS and Meta-RaPS-LP on the standard benchmark instances (a).

	Percentage Gap (%)								**Computation Time**							
	Meta-RaPS				**Meta-RaPS-LP**				**Meta-RaPS**				**Meta-RaPS-LP**			
Instance	**min**	**average**	**median**	**max**	**min**	**average**	**median**	**max**	**min**	**average**	**median**	**max**	**min**	**average**	**median**	**max**
scpnre1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.45	0.64	0.53	0.89	0.41	0.42	0.42	0.44
scpnre2	**0.00**†	**0.00**†	0.00	0.00	3.33	3.33	3.33	3.33	9.33	17.89	13.10	38.27	0.45	0.79	0.52	1.50
scpnre3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.97	1.51	1.02	2.34	0.42	0.91	1.03	1.38
scpnre4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.23	3.76	2.89	6.03	0.49	1.59	1.30	3.20
scpnre5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.42	0.58	0.50	1.05	0.44	0.46	0.45	0.47
scpnrf1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.94	2.07	2.14	3.45	1.13	1.70	1.22	3.17
scpnrf2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.94	1.06	0.95	1.25	1.06	1.19	1.09	1.36
scpnrf3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.28	6.21	6.14	9.42	1.11	2.09	1.75	3.39
scpnrf4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	6.14	5.39	12.16	1.98	2.61	2.83	3.27
scpnrf5	**0.00**†	**0.00**†	0.00	0.00	7.69	7.69	7.69	7.69	6.94	61.68	48.97	154.13	1.13	1.18	1.13	1.38
scpnrg1	0.57	1.25	1.14	1.70	0.57	**0.57**†	0.57	0.57	160.91	459.53	358.44	916.16	6.17	187.98	166.33	461.00
scpnrg2	**0.65**†	**1.17**†	1.30	1.30	1.30	1.30	1.30	1.30	21.03	413.19	253.97	858.20	2.78	38.79	15.84	98.70
scpnrg3	**0.60**†	1.81	1.81	2.41	1.20	**1.69**†	1.81	1.81	118.38	469.47	481.86	826.22	6.98	47.46	29.58	121.61
scpnrg4	1.19	2.14	2.38	2.38	**0.60**†	**1.07**†	1.19	1.19	438.80	618.63	625.84	875.13	14.97	86.34	97.89	158.31
scpnrg5	0.60	1.43	1.19	2.38	0.60	**1.07**†	1.19	1.19	444.33	663.96	542.59	936.83	2.88	61.31	52.80	127.83
scpnrh1	1.59	2.86	3.17	3.17	**0.00**†	**1.27**†	1.59	3.17	19.02	243.92	207.55	584.39	16.02	235.19	135.38	630.33
scpnrh2	1.59	2.86	3.17	3.17	1.59	**1.59**†	1.59	1.59	1.77	299.14	46.11	1263.20	13.11	95.47	75.14	240.58
scpnrh3	1.69	3.05	3.39	3.39	**0.00**†	**1.69**†	1.69	3.39	23.20	792.47	1212.06	1311.40	8.14	346.43	223.09	747.52
scpnrh4	1.72	1.72	1.72	1.72	**0.00**†	**0.34**†	0.00	1.72	80.97	384.43	376.67	728.26	41.39	297.02	228.89	692.97
scpnrh5	**0.00**†	**0.36**†	0.00	1.82	1.82	1.82	1.82	1.82	87.80	296.49	103.75	1042.22	3.91	9.77	7.88	21.33
Average	0.51	0.93	0.96	1.17	0.93	1.17	1.19	1.44	71.13	237.14	214.52	478.55	6.25	70.94	52.23	165.99

†: A bold cell marked with this sign indicates a better value compared to its counterpart of the competing algorithm.

Table 6: Benchmarking Meta-RaPS and Meta-RaPS-LP on the railway instances (d).

	Percentage Gap (%)								**Computation Time**							
	Meta-RaPS				**Meta-RaPS-LP**				**Meta-RaPS**				**Meta-RaPS-LP**			
Instance	**min**	**average**	**median**	**max**	**min**	**average**	**median**	**max**	**min**	**average**	**median**	**max**	**min**	**average**	**median**	**max**
rail507	20.69	21.26	21.26	21.84	**4.60**†	**5.06**†	5.17	5.75	3.06	1121.02	924.95	2671.31	1078.22	2530.17	2647.53	4587.73
rail516	11.54	12.09	12.09	12.64	**4.95**†	**6.70**†	7.14	7.69	136.00	2621.12	1891.95	5756.98	255.53	2496.43	1500.11	6291.38
rail582	17.06	17.82	18.01	18.48	**2.37**†	**3.32**†	3.32	3.79	1.06	3072.16	1969.40	5935.33	428.53	2056.97	2225.38	3559.73
rail2536	27.14	27.98	28.30	28.30	**12.77**†	**13.82**†	14.08	14.22	1277.07	2669.72	2020.84	5179.69	3.47	2709.39	1887.17	6199.66
rail2586	23.39	23.98	24.02	24.55	**10.58**†	**10.77**†	10.79	11.01	1298.19	2684.40	2779.23	5225.38	1873.67	3824.03	4332.13	5422.95
rail4284	29.79	30.06	29.98	30.36	**11.84**†	**12.25**†	12.31	12.59	2055.39	4626.27	5527.35	6339.89	5480.62	6147.49	6213.23	6767.01
rail4872	24.48	24.79	24.93	25.00	**10.27**†	**10.72**†	10.86	11.06	2407.99	4763.17	3165.07	7138.33	5.71	3547.47	4921.44	6024.72
Average	21.92	22.42	22.73	22.91	8.03	8.78	8.92	9.27	1025.54	3079.69	2611.25	5463.84	1303.68	3330.28	3389.57	5550.45

†: A bold cell marked with this sign indicates a better value compared to its counterpart of the competing algorithm.

Table 7: Benchmarking Meta-RaPS and Meta-RaPS-LP on the hard cost and coverage correlated instances (e).

	Percentage Gap (%)								**Computation Time**							
	Meta-RaPS				**Meta-RaPS-LP**				**Meta-RaPS**				**Meta-RaPS-LP**			
Instance	**min**	**average**	**median**	**max**	**min**	**average**	**median**	**max**	**min**	**average**	**median**	**max**	**min**	**average**	**median**	**max**
rand16	7.25	8.41	8.37	9.42	**2.00**†	**2.00**†	2.00	2.00	497.94	1593.46	1399.55	3176.91	17.14	230.31	80.66	579.05
rand17	7.43	8.33	8.02	9.29	**1.00**†	**1.20**†	1.00	2.00	107.91	1229.70	1701.87	1963.70	221.67	540.94	616.08	792.41
rand18	6.49	7.33	7.52	8.02	**2.00**†	**2.00**†	2.00	2.00	105.64	1557.40	716.73	2982.92	74.41	851.16	925.19	1523.56
rand19	6.77	7.52	7.49	7.89	**1.00**†	**1.00**†	1.00	1.00	117.06	705.44	1012.20	2251.72	29.17	91.68	110.13	131.84
rand20	6.72	7.31	5.09	8.21	**1.00**†	**1.20**†	1.00	2.00	656.52	1802.18	1276.54	2861.06	454.13	949.66	981.66	1428.39
rand21	1.89	2.70	3.83	3.46	**0.00**†	**0.00**†	0.00	0.00	570.95	1379.92	1181.41	3147.05	2.38	5.08	4.09	9.59
rand22	4.19	4.49	4.03	4.79	**1.00**†	**1.00**†	1.00	1.00	1088.78	1861.70	2050.10	2390.92	0.30	13.74	3.83	50.25
rand24	4.06	4.31	3.46	4.69	**1.00**†	**1.00**†	1.00	1.00	94.78	1796.35	1578.28	3083.15	45.61	122.81	142.05	211.28
Average	5.60	6.30	5.97	6.97	1.13	1.18	1.13	1.38	404.95	1490.77	1364.58	2732.18	105.60	350.67	357.96	590.80

†: A bold cell marked with this sign indicates a better value compared to its counterpart of the competing algorithm.