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 the most successful results on the set covering problem to this day. Our results demonstrate that the use of dual information significantly improves the efficacy of the heuristic both in terms of 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 just become free for academic use. As of 2012, the latest version of its mixed integer linear programming solver is now on average 50% faster than its earlier releases in the last ten years. Even more impressive, this latest version has improved the performance of the previous version by 15% within merely six months (IBM, 2012).

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 primal-dual heuristics for the 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 take a first step in this direction.

In short, we make the following research contributions: (i) We demonstrate that the dual optimal variables obtained by solving the LP relaxation of the SCP bear important information about the optimal solution of the SCP. (ii) We show that this 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 instances that are widely used in the literature.

This paper is organized as follows: Section 2 summarizes the SCP literature. In Section 3, we discuss the motivation for using dual information to efficiently solve the 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 long known to be $\mathcal{NP}$-hard in the strong sense (Garey and Johnson, 1979). Therefore, many heuristic and enumerative algorithms have been developed to effectively solve the SCP. 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.

Since solving a large SCP with an exact method takes an excessively long time, sacrificing optimality but obtaining fairly good solutions within an acceptable time by means of a heuristic is a compromise. Many researchers list various heuristics and approximation algorithms, and they show that their empirical performance is quite good (Caprara et al., 2000; Gomes et al., 2006; Grossman and Wool, 1997). In the literature, 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 relatively negligible time. 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 appear in (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 the 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 work 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). 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. The basis of the primal-dual approach is finding only a feasible solution to the dual of the LP relaxation of the IP formulation of SCP presented in the next section. Using this solution, an integral solution for the 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).

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 the SCP. Using Definition 1.1, we obtain the integer programming model of the SCP as

$$\text{minimize} \quad \sum_{j \in \mathcal{S}} c_j x_j, \tag{1}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{S}} a_{ij} x_j \geq 1, \qquad i \in \mathcal{U}, \tag{2}$$

$$x_j \in \{0, 1\}, \qquad j \in \mathcal{S}. \tag{3}$$

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; that is, $c_1 = c_2 = \cdots = c_n$ with $n = |\mathcal{S}|$, then the problem is called as the unicost set covering problem. When we consider the LP relaxation of the SCP, the integrality constraints (3) are replaced by simple bounds on the variables 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 the 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 dual optimal 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 the 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.

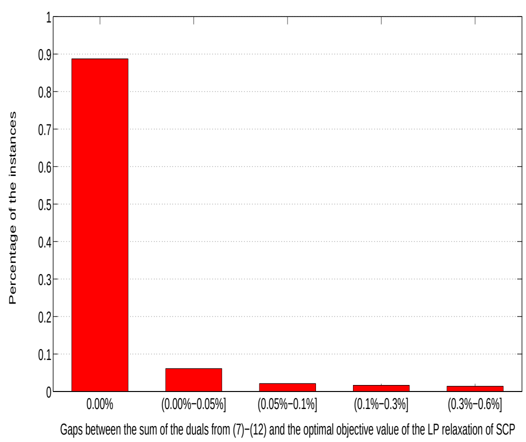
This important relationship between the IP formulation and its LP relaxation has prompted us to concentrate on the best possible result that can be obtained by a primal-dual heuristic that only adds a set to the cover, if the associated reduced cost is zero with respect to a feasible solution of (4)-(6).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

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

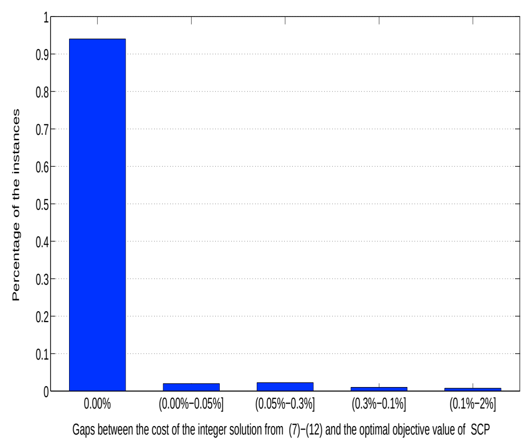
In this model the sets of constraints (8) and (10) ensure that dual and primal problems are feasible, respectively, and the constraints (9) prescribe that a primal variable has a zero reduced cost when it is set to 1.

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 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 (b), (c), and (a) with the exception of the groups *scpnrg* and *scpnrh*. The details of the problem instances are presented in Section 4.

Figure 1(a) shows the average percentage gaps 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). Similarly, Figure 1(b) depicts the average 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). 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. 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 dual optimal solution indeed yields valuable information that could help us select the optimal sets for the SCP. In the subsequent discussion, we concentrate on exploiting this dual information for solving a variety of SCP classes that arise in practice and in the literature.



(a) LP Gap



(b) IP Gap

Figure 1: Average LP-IP percentage gaps: (a) between the optimal objective function value of the LP relaxation and the sum of the duals from the MILP model, (b) between the optimal objective function value of the SCP and the cost of the primal integer solution 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 the 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, 2012) 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, 2012).
- **(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 by (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, 2012).
- **(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. This class is known as the hard cost and coverage correlated set covering problems.
- **(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. Refer to the OR-library (OR-lib, 2012) 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, a group of instances from the highly cited OR-library (a) can be solved to optimality within less than a second on average by a standard IP solver (see next section). Therefore, we focus on hard instances (e-f) and large-scale practical problems (c-d). The Eucledian-type cost and coverage correlated problems (b) are also worth investigating because such problems commonly arise in location, telecommunication, and routing problems (Brusco et al., 1999).

The optimal 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. In 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.

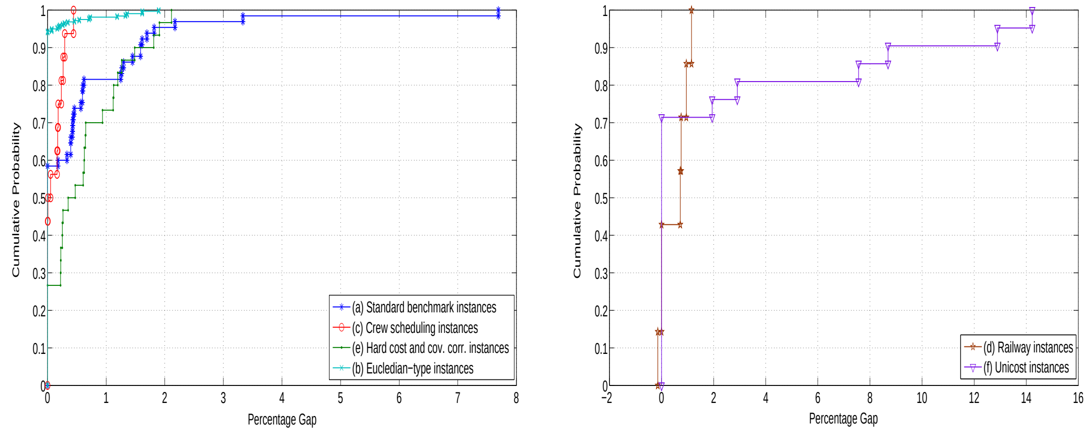
4.2 The Role of Dual Information. We first work with a mathematical programming based heuristic that uses the dual information to extract the most promising columns and then solves the SCP to optimality over those columns. As we discuss in Section 3, this approach is motivated by the MILP model (7)-(12). However, solving this model directly is too demanding, and so we attack it in two phases. 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 at 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.

Tables 1-4 in the appendix report the computational results as well as some statistics for each instance under each category. The optimal/best known objective values are provided in Column 4. The associated solution and root relaxation times appear in the next two columns. The data corresponding to the restricted SCP set up with all sets with zero reduced costs in the optimal solution of the LP relaxation are presented in Columns 7-9. Results for the restricted IP solved over the basic columns in the optimal solution of the LP relaxation follow in Columns 9-12. The time statistics exclude instances that terminate due to the time limit. Table 1 shows the results for the standard benchmark problems (a). These results indicate that the restricted SCP yields integer solutions to within 1% of optimality for a great majority of instances. Moreover, although the performance of the methods varies for different instances, we observe that the overall performance of including only the basic columns is only slightly worse than that of using all columns with zero reduced costs. Table 2 summarizes the results for the crew scheduling problems (c). For these practical problems, the performance of the restricted SCP is better than its performance on standard benchmark instances (a). Table 3 reports the results on another set of practical instances. This time, railway problems (d) are solved. As we observed for the crew scheduling problems, the optimality gaps are lower than those obtained for the standard benchmark instances. Since the sizes of the railway problems are quite large, `CPLEX` cannot solve these instances to optimality within the time limit. In Column 4 we specify the best objective values compiled from the literature to the best of our knowledge. Note that the results retrieved from the restricted SCP are competitive with the best known solutions and even improve that associated with the instance rail2536. In addition, note that even the restricted SCP hits the time limit for three instances, and the corresponding solutions could potentially be improved by increasing the time limit. Table 4 reports the results for unicost problems (f). The structure of the unicost problems is different than those of the other problems given in our study. Except for the group *scpe*, the number of columns is smaller than is the number of rows. The computational results imply that solving the LP relaxation and selecting those columns with 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 the instances in the group *scpe* and a few others. Table 5 summarizes the results for the instances in class (b) and (e). Since the number of instances is quite large in these classes, the individual instances are grouped according to their size, and group averages are reported. The summary statistics are still collected over all instances in a given class. `CPLEX` solves the instances in problem class (b) to optimality in less than 0.5 seconds on average. The heuristic further decreases the computation times with a slight increase in optimality gaps. However, for problem class (e) there is a marked difference between the restricted SCP and the full formulation. The average computation time of set 2 is reduced from 25 seconds to 0.6 seconds.

Figures 2 and 3 depict the empirical cumulative distributions of the optimality gaps and the computation times, respectively, grouped by class. The computational results presented in the appendix indicate that the performance of the heuristic degrades only slightly when the restricted SCP is solved over the basic columns. Therefore, in these figures we only report the results obtained by using the columns with zero reduced costs. In these plots, we can read the cumulative percentage of instances with optimality gaps (or computation times) less than a threshold corresponding to a 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. Except for a very few problem instances, the optimality gap is always within 2%. Only for some of the instances in the sets (a), (e), and (f), the optimality gaps are slightly larger. This can be partly explained by the imposed restriction of 7,200 seconds on the solution time. As we mentioned before, all columns have a zero reduced cost after solving the LP relaxation for the unicost problems (f). Thus, in this case we end up solving the original problem in the second phase which explains the different behavior of our heuristic for this problem class. The negative value on the horizontal axis of Figure 2(b) is due to the instance rail2536 for which we improve the best known objective function value. Figure 3 depicts that the computation time is less than 2 seconds for a great majority of the instances. The relatively longer computation times observed for some of the instances in problem classes (a), (d), and (f) can presumably be decreased at the expense of slightly larger optimality gaps.

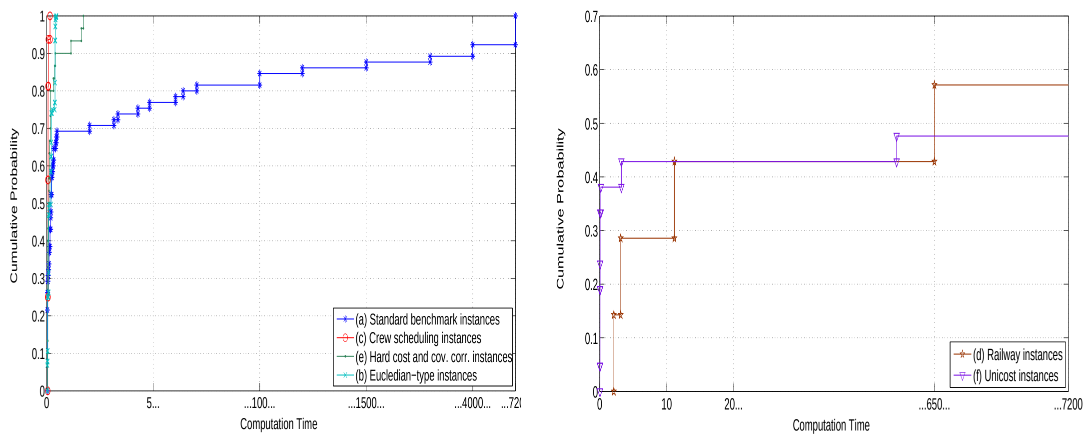
We next investigate how much reduction we obtain in both the problem size and the computation time when we solve the restricted SCP over the zero reduced cost columns. To this end, we collect statistics on the instances in problem sets (a-c) and (e) which are solved to optimality within our time limit. The stacked bar plot in Figure 4 summarizes the results. We report the average percentage reductions in the problem size against the original size, and the average percentage time reductions over the solution times obtained by `CPLEX`. We note that the reported times for our approach also include the time to solve the LP relaxation in the first phase. Figure 4 clearly shows that considerable reduction can be achieved both in solution time and problem size. For example, the use of dual information results in 94.69% size reduction and 68.44% time reduction for the standard benchmark instances (a).

4.3 Improving a Local Search Method with the Dual Information. In the previous section, we show that the dual information is a significant tool to extract the set of promising columns in an SCP problem. Following up on this observation, we aim at showing that the performance of an SCP heuristic



(a) Problem sets (a), (b), (c), and (e) (b) Problem sets (d) and (f)

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



(a) Problem sets (a), (b), (c), and (e) (b) Problem sets (d) and (f)

Figure 3: The empirical cumulative distribution of the computation times for the restricted SCP solved over the columns with zero reduced costs.

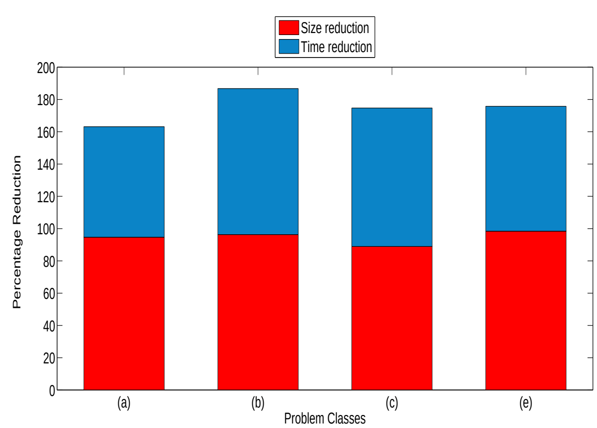
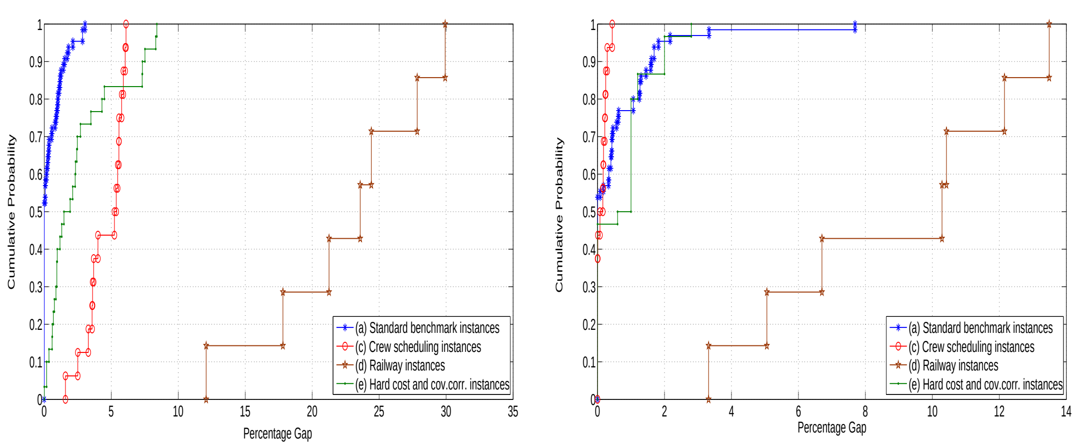


Figure 4: The percentage size and time reductions that we obtain relative to `CPLEX` when we solve the problems by the proposed mathematical programming-based heuristic.

may also be improved when dual information is embedded. To serve this purpose, we use the well-known heuristic Meta-RaPS (Lan et al., 2007) because, to the best of our knowledge, its performance has not been outperformed in the literature. Meta-RaPS is one of the randomized local search heuristics in the literature. The algorithm consists of two phases. In the construction phase, one set is selected randomly among the set of best candidates and added to the cover until a feasible solution is obtained. One of the four priority rules determines the best candidates. The rules that Meta-RaPS uses are as follows: 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. To this end, 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 a feasible solution is restricted to a set of "promising" columns identified during the course of the algorithm in order to enhance the computational speed. The authors refer to this restricted pool of columns as the "core problem" and define it as those columns selected at least once in the candidate list during the construction phase. Lan et al. tested 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 these parameters that they used to solve the standard benchmark instances (a). We also employ the same set of parameter values in our numerical experiments. Our proposed extension, Meta-RaPS-LP, is basically a Meta-RaPS implementation but instead of solving the original problem over all columns, we solve the problem only over the columns with zero reduced costs obtained by solving the LP relaxation of the problem. Note the parallel to the definition of the restricted SCP in the previous section.

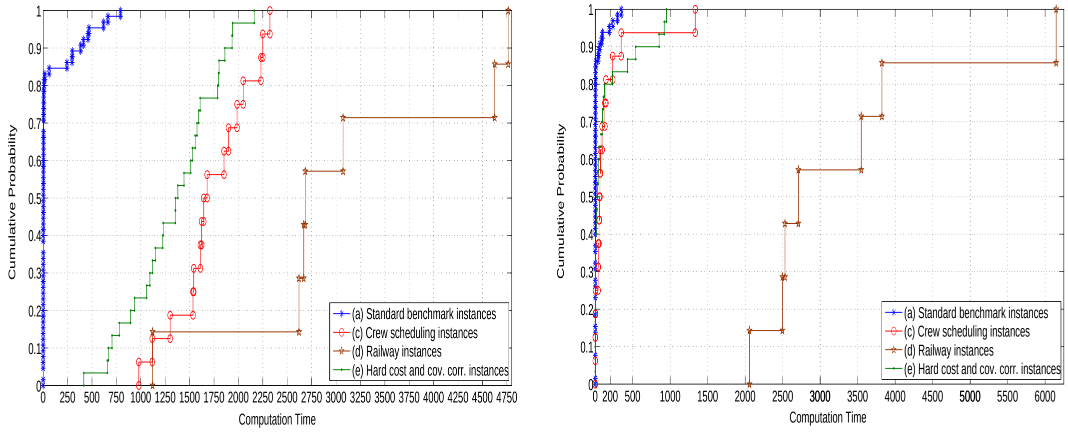
Table 6 in the appendix summarizes the statistics on the optimality gaps and the solution times obtained by both Meta-RaPS and Meta-RaPS-LP on the standard benchmark instances (a). These computational results are reported after taking 5 replications for each instance because the algorithm incorporates randomness. They demonstrate that except for a few instances, Meta-RaPS-LP performs quite well with respect to Meta-RaPS, especially when the set of columns identified by the dual information includes at least one optimal or near-optimal solution. In several instances, however, the optimality gaps turn out to be higher; see for example, the instances scp61, scpb1, scpb3, scpb4, and scpnrf5. 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 6 to observe that the gaps of Meta-RaPS-LP for the instances mentioned above are identical to those of the restricted SCP for these instances. That is to say, Meta-RaPS-LP could not do any better. Table 6 also summarizes the computation times of the heuristic with and without the dual information. Like Lan et al. (2007), we also report the time when the best solution is found for the first time. As previously, the reported times for Meta-RaPS-LP include the time to solve the LP relaxation. The results clearly state that for a minor loss in solution quality we can achieve significant computational savings. Overall, the average solution time is reduced to 22 seconds from 74 seconds if we can tolerate a minor jump in average solution quality from 0.48% to 0.55%. However, when we explore the performance of Meta-RaPS and Meta-RaPS-LP for the the large-scale instances *scpnrg* and *scpnrh* in this class, we observe that with Meta-RaPS-LP we generally attain better solutions with computational savings up to a order of magnitude or more at times.

We next test the effect of embedding the dual information on the solution quality and time for problem classes (c), (d), and (e). Since using the dual information does not have a significant benefit for solving the unicost instances, we have omitted them from our subsequent numerical results. Class (b) is excluded because instances in this class are relatively simple. We set the maximum time limit to 7,200 seconds for set (d) and to 3,600 seconds for sets (c) and (e). The sizes of the instances in these classes are larger than those in the standard benchmark class (a). Thus, we increased the number of iterations spent in the neighborhood search to 500 from its original value of 400. Tables 7-9 in the appendix summarize the results on the problem classes (c), (d), and (e), respectively. Table 7 attests to the value of embedding dual information in Meta-RaPS for the crew scheduling instances. Both the optimality gaps and the computation times are substantially reduced. Note that 6 out of 16 instances are consistently solved to optimality as evident from zero average gaps for Meta-RaPS-LP. The figures in Table 8 correspond to the results for the significantly larger railway instances in class (d). Although the solution times attained by Meta-RaPS and Meta-RaPS-LP are close to each other, both the average and minimum optimality gaps achieved by Meta-RaPS-LP are always less than half of those by Meta-RaPS (the single exception is the average optimality gap for the instance rail516). Finally, Table 9 presents the results for the hard cost and coverage correlated problems. We achieve great reductions in the computational effort expended with better solution quality for this class of problems. While the original algorithm Meta-RaPS solves 2 out of 30 instances to optimality at least once, the corresponding figure for Meta-RaPS-LP is 17. Note that 14 out 17 of these instances are always solved optimally. Figures 5-6 summarize the information presented in Tables 6-9. They depict the empirical cumulative distributions of the optimality gaps and solution times obtained by Meta-RaPS with and without the dual information. Figure 5 shows that the dual information helps to decrease the optimality gaps in almost all problem classes (except for some of the standard benchmark problems). We observe in Figure 6 that when dual information is used, the empirical cumulative distributions of the computation times shift to the left as desired. We note that the significant improvements in the optimality gaps for the railway instances are sometimes attained at the expense of increased solution times.



(a) Meta-RaPS (b) Meta-RaPS-LP

Figure 5: The empirical cumulative distributions of the optimality gaps of Meta-RaPS and Meta-RaPS-LP for classes (a), (c), (d), and (e).

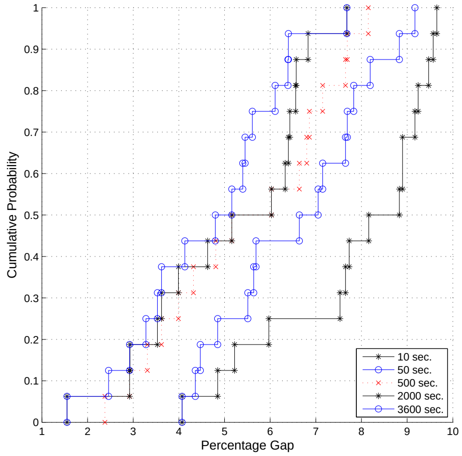


(a) Meta-RaPS (b) Meta-RaPS-LP

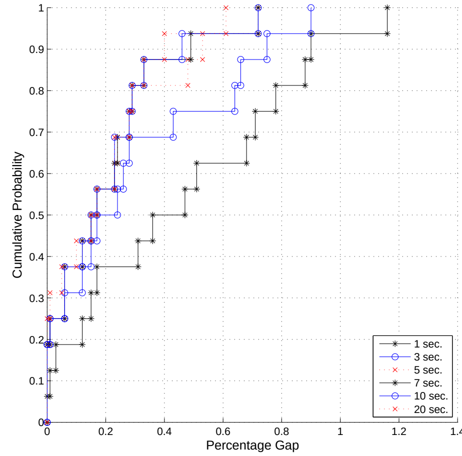
Figure 6: The empirical cumulative distributions of the computation times of Meta-RaPS and Meta-RaPS-LP for classes (a), (c), (d), and (e).

Finally, in Figure 7 we present a detailed analysis of the solution quality versus the solution time for problem sets (c), (d) and (e). As the algorithm progresses, we take snapshots of the optimality gaps at different times. We observe in Figure 7(a) that after 500 seconds, roughly 80% of all instances are within 7.5% of optimality. After 2,000 seconds, however, the optimality gaps are less than 7.5% for all instances. In the same vein, Figure 7(b) depicts the optimality gaps versus the solution time after using the dual information. It shows that after only 5 seconds all instances are solved within 0.65% of optimality. Figures 7(c) and 7(d) summarize the results of a similar analysis for the cost and coverage correlated problems. As illustrated in Figure 7(c), after 10 seconds the optimality gaps are slightly less than 8% for 80% of all instances. In Figure 7(d), the optimality gaps are decreased to 4% for the same percentage of all instances in 10 second when we use the dual information. The last two figures, Figure 7(e) and Figure 7(f), illustrate that for a railway instance we generally do not obtain a considerable decrease in the optimality gap as the solution time increases. However, the benefit of using the dual information stands out clearly when we check the optimality gaps. After 1,000 seconds, the optimality 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 optimality gap decreases to 12% for the same percentage of the instances when we embed the dual information.

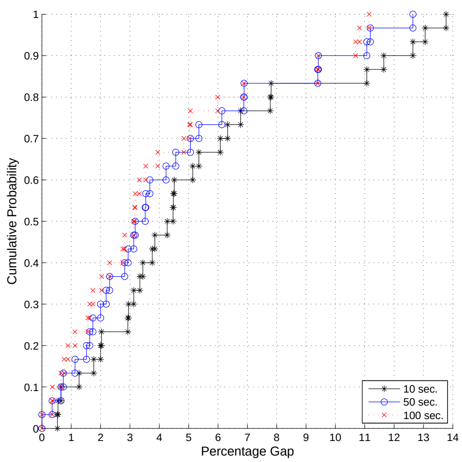
5. Conclusions and Future Research Directions. Our empirical study supports the claim that the dual optimal solution of the LP relaxation of SCP provides an important instrument for tackling this celebrated problem. By using the dual information, significant reductions in problem size and gains in solution quality can be achieved for large-scale instances which otherwise are 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.



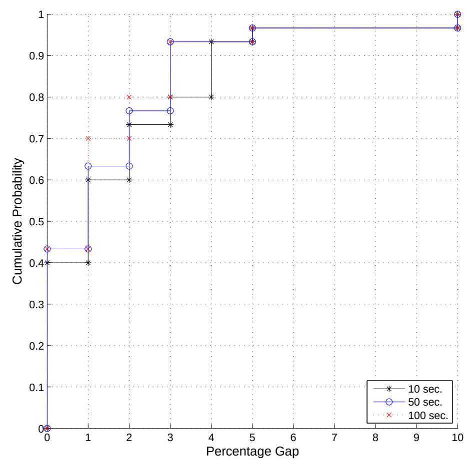
(a) Crew scheduling instances solved by the original Meta-RaPS.



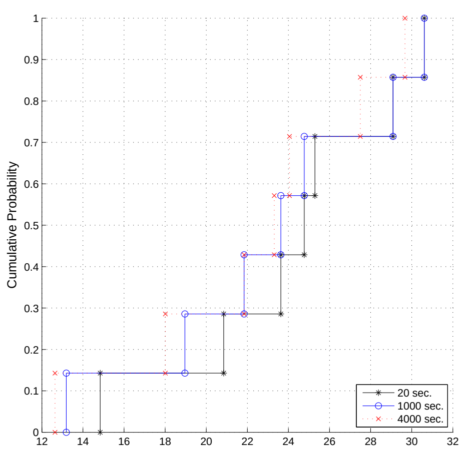
(b) Crew scheduling instances solved by Meta-RaPS-LP.



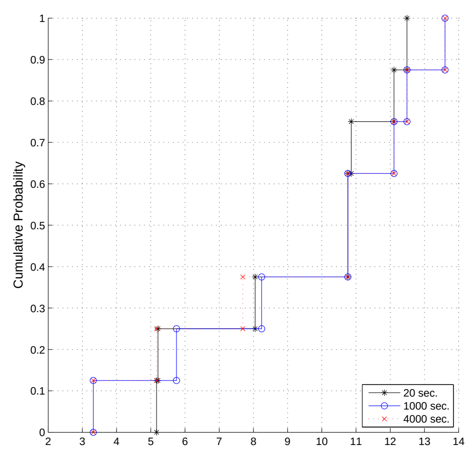
(c) Hard cost and coverage correlated instances solved by the original Meta-RaPS.



(d) Hard cost and coverage correlated instances solved by Meta-RaPS-LP.



(e) Railway instances solved by the original Meta-RaPS.



(f) Railway instances solved by Meta-RaPS-LP.

Figure 7: The progress of the optimality gaps during the course of the algorithm.

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 the 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 dual information improved both the solution time and the solution quality. This improvement is more apparent for large-scale SCP instances. In future research studies, more sophisticated algorithms may be developed that make use of the dual information as proposed here.

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.

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

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.

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.

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 (2012). http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/.

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.

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 (2012). 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.

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 cocering 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.

Appendix A. Tables 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	$\vert\mathcal{U}\vert$	$\vert\mathcal{S}\vert$	Opt.	IPTime	LPTime	OFV	Time	IP Gap(%)	OFV	Time	IP Gap(%)
scp41	200	1,000	429	0.594	0.187	429	0.234	0.00	429	0.140	0.00
scp42	200	1,000	512	0.031	0.031	512	0.031	0.00	512	0.031	0.00
scp43	200	1,000	516	0.016	0.016	516	0.031	0.00	516	0.031	0.00
scp44	200	1,000	494	0.047	0.032	494	0.032	0.00	494	0.031	0.00
scp45	200	1,000	512	0.001	0.016	512	0.031	0.00	512	0.031	0.00
scp46	200	1,000	560	0.078	0.016	561	0.031	0.18	561	0.032	0.18
scp47	200	1,000	430	0.016	0.032	430	0.032	0.00	430	0.031	0.00
scp48	200	1,000	492	0.063	0.016	492	0.031	0.00	492	0.031	0.00
scp49	200	1,000	641	0.047	0.031	641	0.031	0.00	641	0.031	0.00
scp410	200	1,000	514	0.016	0.016	514	0.031	0.00	514	0.031	0.00
		Average		0.091	0.039		0.052	0.02		0.040	0.02
		Minimum		0.001	0.016		0.031	0.00		0.030	0.00
		Maximum		0.594	0.187		0.234	0.18		0.140	0.18
		Median		0.039	0.024		0.031	0.00		0.030	0.00
scp51	200	2,000	253	0.063	0.031	254	0.062	0.40	254	0.047	0.40
scp52	200	2,000	302	0.094	0.031	303	0.047	0.33	304	0.063	0.66
scp53	200	2,000	226	0.031	0.016	226	0.031	0.00	226	0.032	0.00
scp54	200	2,000	242	0.063	0.016	242	0.031	0.00	242	0.031	0.00
scp55	200	2,000	211	0.031	0.031	211	0.031	0.00	211	0.031	0.00
scp56	200	2,000	213	0.031	0.016	213	0.032	0.00	213	0.047	0.00
scp57	200	2,000	293	0.046	0.031	293	0.031	0.00	293	0.032	0.00
scp58	200	2,000	288	0.063	0.015	288	0.047	0.00	288	0.047	0.00
scp59	200	2,000	279	0.015	0.016	279	0.031	0.00	279	0.047	0.00
scp510	200	2,000	265	0.031	0.031	265	0.031	0.00	265	0.032	0.00
		Average		0.047	0.023		0.037	0.07		0.041	0.11
		Minimum		0.015	0.015		0.031	0.00		0.031	0.00
		Maximum		0.094	0.031		0.062	0.40		0.063	0.66
		Median		0.039	0.024		0.031	0.00		0.040	0.00
scp61	200	1,000	138	0.156	0.015	141	0.109	2.17	141	0.094	2.17
scp62	200	1,000	146	0.250	0.016	146	0.094	0.00	146	0.110	0.00
scp63	200	1,000	145	0.156	0.031	145	0.109	0.00	145	0.094	0.00
scp64	200	1,000	131	0.046	0.015	131	0.031	0.00	131	0.046	0.00
scp65	200	1,000	161	0.234	0.032	162	0.125	0.62	162	0.125	0.62
		Average		0.168	0.022		0.094	0.56		0.094	0.56
		Minimum		0.046	0.015		0.031	0.00		0.046	0.00
		Maximum		0.250	0.032		0.125	2.17		0.125	2.17
		Median		0.156	0.016		0.109	0.00		0.094	0.00
scpa1	300	3,000	253	0.422	0.047	254	0.156	0.40	254	0.157	0.40
scpa2	300	3,000	252	0.422	0.063	253	0.156	0.40	253	0.187	0.40
scpa3	300	3,000	232	0.235	0.047	233	0.188	0.43	233	0.156	0.43
scpa4	300	3,000	234	0.140	0.047	235	0.156	0.43	235	0.125	0.43
scpa5	300	3,000	236	0.110	0.046	237	0.078	0.42	237	0.063	0.42
		Average		0.266	0.050		0.147	0.41		0.138	0.41
		Minimum		0.110	0.046		0.078	0.40		0.063	0.40
		Maximum		0.422	0.063		0.188	0.43		0.187	0.43
		Median		0.235	0.047		0.156	0.42		0.156	0.42
scpb1	300	3,000	69	0.469	0.078	70	0.203	1.45	70	0.188	1.45
scpb2	300	3,000	76	0.688	0.078	76	0.281	0.00	76	0.266	0.00
scpb3	300	3,000	80	0.390	0.062	81	0.187	1.25	81	0.188	1.25
scpb4	300	3,000	79	0.844	0.078	80	0.312	1.27	80	0.297	1.27
scpb5	300	3,000	72	0.469	0.062	72	0.203	0.00	72	0.219	0.00
		Average		0.572	0.072		0.237	0.79		0.232	0.79
		Minimum		0.390	0.062		0.187	0.00		0.188	0.00
		Maximum		0.844	0.078		0.312	1.45		0.297	1.45
		Median		0.469	0.078		0.203	1.25		0.219	1.25
scpc1	400	4,000	227	0.484	0.078	228	0.187	0.44	228	0.188	0.44
scpc2	400	4,000	219	0.438	0.094	220	0.234	0.46	220	0.250	0.46
scpc3	400	4,000	243	1.000	0.078	243	0.250	0.00	243	0.250	0.00
scpc4	400	4,000	219	0.594	0.093	219	0.234	0.00	219	0.250	0.00

Continued on next page. . .

			CPLEX			Restricted SCP-ZeroRC			Restricted SCP-BC		
Instance	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{S}\rvert$	**Opt.**	**IPTime**	**LPTime**	**OFV**	**Time**	**IP Gap(%)**	**OFV**	**Time**	**IP Gap(%)**
scpc5	400	4,000	215	0.453	0.093	215	0.203	0.00	215	0.250	0.00
		Average		0.594	0.087		0.222	0.18		0.238	0.18
		Minimum		0.438	0.078		0.187	0.00		0.188	0.00
		Maximum		1.000	0.094		0.250	0.46		0.250	0.46
		Median		0.484	0.093		0.234	0.00		0.250	0.00
scpd1	400	4,000	60	0.750	0.156	60	0.328	0.00	60	0.343	0.00
scpd2	400	4,000	66	3.015	0.125	66	0.453	0.00	66	0.453	0.00
scpd3	400	4,000	72	1.219	0.125	72	0.422	0.00	72	0.500	0.00
scpd4	400	4,000	62	2.625	0.125	63	0.484	1.61	63	0.594	1.61
scpd5	400	4,000	61	0.641	0.125	61	0.328	0.00	61	0.375	0.00
		Average		1.650	0.131		0.403	0.32		0.453	0.32
		Minimum		0.641	0.125		0.328	0.00		0.343	0.00
		Maximum		3.015	0.156		0.484	1.61		0.594	1.61
		Median		1.219	0.125		0.422	0.00		0.453	0.00
scpnre1	500	5,000	29	38.907	0.282	29	6.047	0.00	29	6.078	0.00
scpnre2	500	5,000	30	53.781	0.312	31	15.953	3.33	31	19.328	3.33
scpnre3	500	5,000	27	16.812	0.297	27	4.828	0.00	27	4.031	0.00
scpnre4	500	5,000	28	38.031	0.328	28	6.406	0.00	28	10.453	0.00
scpnre5	500	5,000	28	15.828	0.281	28	3.344	0.00	28	2.563	0.00
		Average		32.672	0.300		7.316	0.67		8.491	0.67
		Minimum		15.828	0.281		3.344	0.00		2.563	0.00
		Maximum		53.781	0.328		15.953	3.33		19.328	3.33
		Median		38.031	0.297		6.047	0.00		6.078	0.00
scpnrf1	500	5,000	14	35.937	0.470	14	4.281	0.00	14	4.015	0.00
scpnrf2	500	5,000	15	13.703	62.000	15	3.156	0.00	15	3.281	0.00
scpnrf3	500	5,000	14	7.765	0.422	14	2.015	0.00	14	2.343	0.00
scpnrf4	500	5,000	14	29.547	0.484	14	7.047	0.00	14	5.937	0.00
scpnrf5	500	5,000	13	28.875	0.484	14	51.984	7.69	14	77.969	7.69
		Average		23.165	12.772		13.697	1.54		18.709	1.54
		Minimum		7.765	0.422		2.015	0.00		2.343	0.00
		Maximum		35.937	62.000		51.984	7.69		77.969	7.69
		Median		28.875	0.484		4.281	0.00		4.015	0.00
scpnrg1	1,000	10,000	176†	⋄	0.891	177	1092.420	0.57	177	715.531	0.57
scpnrg2	1,000	10,000	154†	⋄	0.828	156	310.484	1.30	156	217.438	1.30
scpnrg3	1,000	10,000	166†	⋄	0.890	167	2505.920	0.60	167	2874.750	0.60
scpnrg4	1,000	10,000	168†	⋄	1.047	169	1203.610	0.60	169	2697.560	0.60
scpnrg5	1,000	10,000	168†	⋄	1.063	169	3362.300	0.60	169	4608.250	0.60
		Average			0.944		1694.947	0.73		2222.706	0.73
		Minimum			0.828		310.484	0.57		217.438	0.57
		Maximum			1.063		3362.300	1.30		4608.250	1.30
		Median			0.891		1203.61	0.60		2697.560	0.60
scpnrh1	1,000	10,000	63	⋄	1.203	64	⋄	1.59	63	⋄	0.00
scpnrh2	1,000	10,000	63	⋄	1.422	64	⋄	1.59	64	⋄	1.59
scpnrh3	1,000	10,000	59	⋄	1.359	60	⋄	1.69	59	⋄	0.00
scpnrh4	1,000	10,000	58	⋄	1.156	58	⋄	0.00	58	⋄	0.00
scpnrh5	1,000	10,000	55	⋄	1.141	56	⋄	1.82	56	⋄	1.82
		Average			1.256			1.34			0.68
		Minimum			1.141			0.00			0.00
		Maximum			1.422			1.82			1.82
		Median			1.203			1.59			0.00

†: Best known objective function value.
⋄: Terminated due to time limit.

Table 2: Performance statistics for the restricted SCP on crew scheduling problems (c).

			CPLEX			Restricted SCP-ZeroRC			Restricted SCP-BC		
Instance	$\vert\mathcal{U}\vert$	$\vert\mathcal{S}\vert$	**Opt.**	**IPTime**	**LPTime**	**OFV**	**Time**	**IP Gap(%)**	**OFV**	**Time**	**IP Gap(%)**
aa03	106	8,661	33,155	0.187	0.140	33,157	0.156	0.01	33,157	0.078	0.01
aa04	106	8,002	34,573	0.125	0.047	34,633	0.063	0.17	34,633	0.062	0.17
aa05	105	7,435	31,623	0.110	0.047	31,623	0.063	0.00	31,623	0.047	0.00
aa06	105	6,951	37,464	0.109	0.047	37,464	0.062	0.00	37,464	0.063	0.00
aa11	271	4,413	35,384	0.110	0.063	35,478	0.078	0.27	35,478	0.093	0.27
aa12	272	4,208	30,809	0.110	0.047	30,859	0.062	0.16	30,859	0.078	0.16
aa13	265	4,025	33,211	0.063	0.031	33,211	0.047	0.00	33,211	0.047	0.00
aa14	266	3,868	33,219	0.125	0.047	33,279	0.078	0.18	33,279	0.063	0.18
aa15	267	3,701	34,409	0.141	0.031	34,510	0.047	0.29	34,510	0.063	0.29
aa16	265	3,558	32,752	0.125	0.047	32,769	0.062	0.05	32,858	0.047	0.32
aa17	264	3,425	31,612	0.047	0.047	31,612	0.063	0.00	31,612	0.046	0.00
aa18	271	3,314	36,782	0.125	0.047	36,945	0.062	0.44	36,945	0.063	0.44
aa19	263	3,202	32,317	0.046	0.047	32,317	0.047	0.00	32,317	0.062	0.00
aa20	269	3,095	34,912	0.078	0.031	34,994	0.062	0.23	34,994	0.063	0.23
bus1	454	2,241	27,947	0.031	0.031	27,947	0.047	0.00	27,947	0.031	0.00
bus2	681	9,524	67,760	0.063	0.047	67,760	0.063	0.00	67,760	0.062	0.04
Average				0.100	0.050		0.066	0.11		0.061	0.13
Minimum				0.031	0.031		0.047	0.00		0.031	0.00
Maximum				0.187	0.140		0.156	0.44		0.093	0.44
Median				0.110	0.047		0.062	0.03		0.063	0.10

Table 3: Performance statistics for the restricted SCP on railway problems (d).

			CPLEX			Restricted SCP-ZeroRC			Restricted SCP-BC		
Instance	$\vert\mathcal{U}\vert$	$\vert\mathcal{S}\vert$	**Opt.**	**IPTime**	**LPTime**	**OFV**	**Time**	**IP Gap(%)**	**OFV**	**Time**	**IP Gap(%)**
rail507	507	63,009	174†	$\diamond$	4.203	176	11.156	1.15	177	2.875	1.15
rail516	516	47,311	182†	$\diamond$	1.594	182	2.110	0.00	183	1.781	0.55
rail582	582	55,515	211†	$\diamond$	2.657	211	3.141	0.00	212	2.891	0.47
rail2536	2,536	1,081,841	691†	$\diamond$	1725.800	690	615.680	**-0.14**	696	2776.250	0.72
rail2586	2,586	920,683	948†	$\diamond$	884.515	957	$\diamond$	0.95	963	$\diamond$	1.58
rail4284	4,284	1,092,610	1065†	$\diamond$	5000.520	1073	$\diamond$	0.75	1081	$\diamond$	1.50
rail4872	4,872	968,672	1534†	$\diamond$	3917.510	1545	$\diamond$	0.72	1554	$\diamond$	1.30
Average					1618.114		158.022	0.49		695.949	1.04
Minimum					1.594		2.110	**-0.14**		1.781	0.47
Maximum					5000.520		615.680	1.15		2776.250	1.58
Median					884.515		7.148	0.72		2.883	1.15

$\dagger$: Best known objective function value.
$\diamond$: Terminated due to time limit.

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

			CPLEX			Restricted SCP-ZeroRC			Restricted SCP-BC		
Instance	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{S}\rvert$	**Opt.**	**IPTime**	**LPTime**	**OFV**	**Time**	**IP Gap(%)**	**OFV**	**Time**	**IP Gap(%)**
scpcyc06	240	192	60†	⋄	0.031	60	⋄	0.00	60	1843.280	0.00
scpcyc07	672	448	144†	⋄	0.140	144	⋄	0.00	150	⋄	4.17
scpcyc08	1,792	1,024	344†	⋄	0.453	354	⋄	2.91	357	⋄	3.78
scpcyc09	4,608	2,304	780†	⋄	4.063	839	⋄	7.56	862	⋄	10.51
scpcyc10	11,520	5,120	1792†	⋄	53.266	2047	⋄	14.23	2058	⋄	14.84
scpcyc11	28,160	11,264	4103†	⋄	928.344	4632	⋄	12.89	4719	⋄	15.01
scpclr10	511	210	25†	⋄	0.063	25	44.344	0.00	25	8.765	0.00
scpclr11	1,023	330	23†	⋄	0.312	23	⋄	0.00	23	⋄	0.00
scpclr12	2,047	495	23†	⋄	2.375	23	⋄	0.00	23	⋄	0.00
scpclr13	4,095	715	23†	⋄	8.250	25	⋄	8.70	23	⋄	0.00
scpe1	50	500	5	0.204	0.031	5	0.063	0.00	5	0.062	0.00
scpe2	50	500	5	0.235	0.016	5	0.047	0.00	5	0.047	0.00
scpe3	50	500	5	0.188	0.015	5	0.047	0.00	5	0.062	0.00
scpe4	50	500	5	0.172	0.031	5	0.062	0.00	5	0.063	0.00
scpe5	50	500	5	0.219	0.031	5	0.063	0.00	5	0.047	0.00
STS9	12	9	5†	⋄	0.031	5	0.031	0.00	5	0.031	0.00
STS15	35	15	9†	⋄	0.015	9	0.047	0.00	9	0.032	0.00
STS27	117	27	18†	⋄	0.015	18	0.156	0.00	18	0.156	0.00
STS45	330	45	30†	⋄	0.016	30	3.235	0.00	30	4.109	0.00
STS81	1,080	81	61†	⋄	0.172	61	⋄	0.00	61	⋄	0.00
STS135	3,015	135	103†	⋄	0.062	105	⋄	1.94	105	⋄	1.94
		Average		0.204	47.511		4.809	2.30		168.787	2.39
		Minimum		0.172	0.015		0.031	0.00		0.031	0.00
		Maximum		0.235	928.344		44.344	14.23		1843.280	15.01
		Median		0.204	0.062		0.063	0.00		0.063	0.00

†: Best known objective function value.
⋄: Terminated due to time limit.

Table 5: Performance statistics for the restricted SCP on cost and coverage correlated problems (b) and (e).

				CPLEX		Restricted SCP-ZeroRC		Restricted SCP-BC	
Type	**Instance**	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{S}\rvert$	**IPTime**	**LPTime**	**Time**	**IP Gap(%)**	**Time**	**IP Gap(%)**
(b)	set 1	40	1,560	0.048	0.035	0.042	0.00	0.041	0.00
	set 2	60	3,540	0.154	0.087	0.097	0.04	0.098	0.05
	set 3	80	6,320	0.359	0.201	0.212	0.02	0.214	0.02
	set 4	100	9,900	0.713	0.381	0.395	0.10	0.400	0.11
			Average	0.319	0.176	0.186	0.04	0.188	0.04
			Minimum	0.031	0.031	0.031	0.00	0.031	0.00
			Maximum	0.891	0.437	0.453	1.90	0.672	1.90
			Median	0.312	0.140	0.203	0.00	0.172	0.00
(e)	set1	200	1,000	0.123	0.024	0.058	0.41	0.056	0.49
	set2	200	2,000	25.310	0.055	0.451	0.81	0.60	0.81
			Average	12.717	0.040	0.255	0.61	0.289	0.65
			Minimum	0.015	0.015	0.031	0.00	0.015	0.00
			Maximum	185.765	0.078	1.719	2.11	2.031	2.42
			Median	0.140	0.031	0.102	0.41	0.109	0.41

Table 6: Benchmarking Meta-RaPS and Meta-RaPS-LP on the standard benchmark instances (a).

	Optimality 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**
scp41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.69	1.36	1.20	2.88	0.17	0.65	0.75	1.23
scp42	0.00	0.27	0.20	0.59	0.00	0.00†	0.00	0.00	1.47	2.99	3.13	4.13	0.08	0.09	0.08	0.16
scp43	0.00	0.35	0.39	0.58	0.00	0.00†	0.00	0.00	0.61	2.19	2.27	3.97	0.08	0.17	0.16	0.28
scp44	0.20	0.32	0.40	0.40	0.00†	0.00†	0.00	0.00	1.81	3.48	3.34	5.41	0.08	0.14	0.11	0.28
scp45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	1.39	0.58	4.64	0.08	0.10	0.08	0.19
scp46	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.38	1.76	1.66	3.39	0.08	0.28	0.33	0.44
scp47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.77	0.80	1.49	0.08	0.08	0.08	0.08
scp48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.58	1.88	1.16	3.69	0.16	0.33	0.27	0.58
scp49	0.62	0.81	0.62	1.40	0.00†	0.00†	0.00	0.00	2.67	3.88	4.16	4.61	0.09	0.69	0.63	1.47
scp410	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.41	0.45	0.78	0.08	0.08	0.08	0.09
scp51	0.00†	0.08†	0.00	0.40	0.40	0.40	0.40	0.40	1.03	2.48	2.56	3.69	0.14	0.25	0.22	0.39
scp52	0.66	1.52	1.66	1.99	0.33†	0.33†	0.33	0.33	0.06	4.23	5.00	6.47	0.50	1.74	2.00	2.83
scp53	0.00	0.09	0.00	0.44	0.00	0.00†	0.00	0.00	0.11	1.26	0.74	3.20	0.06	0.10	0.08	0.14
scp54	0.00	0.25	0.41	0.41	0.00	0.00†	0.00	0.00	0.30	0.67	0.59	1.16	0.09	0.19	0.16	0.34
scp55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.41	0.66	0.59	1.09	0.08	0.08	0.08	0.08
scp56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.06	0.06	0.06	0.06	0.07	0.08	0.08
scp57	0.00†	0.00†	0.00	0.00	0.34	0.34	0.34	0.34	0.23	1.15	0.83	2.19	0.09	0.19	0.20	0.27
scp58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19	1.68	1.13	4.25	0.08	0.18	0.17	0.31
scp59	0.00	0.07	0.00	0.36	0.00	0.00†	0.00	0.00	0.06	1.23	0.83	3.28	0.08	0.12	0.08	0.20
scp510	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.22	0.17	0.44	0.08	0.08	0.08	0.08
scp61	0.00†	0.58†	0.72	0.72	2.17	2.17	2.17	2.17	0.19	2.12	2.74	3.34	0.06	0.06	0.06	0.06
scp62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.25	0.17	0.53	0.06	0.07	0.06	0.11
scp63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.16	0.11	0.25	0.06	0.07	0.06	0.08
scp64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.17	0.14	0.27	0.05	0.07	0.06	0.09
scp65	0.00†	0.00†	0.00	0.00	0.62	0.62	0.62	0.62	0.05	0.46	0.39	1.00	0.06	0.11	0.06	0.23
scpa1	0.79	1.11	1.19	0.00	0.40†	0.63†	0.79	0.79	0.80	4.00	4.92	6.47	0.50	2.39	1.47	5.64
scpa2	0.00†	0.08†	0.00	0.40	0.40	0.40	0.40	0.40	0.36	5.08	3.22	10.20	0.13	2.28	1.05	6.77
scpa3	0.43	1.03	0.86	1.72	0.43	0.43†	0.43	0.43	0.89	2.87	2.56	6.13	0.13	0.21	0.14	0.33
scpa4	0.00†	0.17†	0.00	0.43	0.43	0.43	0.43	0.43	0.75	1.36	0.98	3.17	0.28	1.42	1.25	2.89
scpa5	0.85	1.02	0.85	1.27	0.42†	0.42†	0.42	0.42	0.69	3.85	2.30	9.00	0.13	0.39	0.39	0.59
scpb1	0.00†	0.00†	0.00	0.00	1.45	1.45	1.45	1.45	0.11	0.18	0.22	0.24	0.11	0.12	0.11	0.13
scpb2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.27	0.22	0.69	0.13	0.24	0.20	0.50
scpb3	0.00†	0.00†	0.00	0.00	1.25	1.25	1.25	1.25	0.11	0.56	0.72	0.94	0.11	0.11	0.11	0.11
scpb4	0.00†	0.00†	0.00	0.00	1.27	1.27	1.27	1.27	0.13	1.04	0.56	3.28	0.11	0.20	0.19	0.34
scpb5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.16	0.13	0.23	0.11	0.11	0.11	0.11
scpc1	0.44	0.88	0.88	1.32	0.44	0.44†	0.44	0.44	2.34	5.82	3.69	14.89	0.55	1.63	0.94	4.52
scpc2	0.46	0.55	0.46	0.91	0.46	0.46†	0.46	0.46	1.78	8.19	7.17	13.77	0.30	0.70	0.44	1.88

Continued on next page…

	Optimality 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**
scpc3	0.82	0.99	0.82	1.23	0.00†	0.08†	0.00	0.41	2.83	10.40	11.16	17.67	0.31	5.03	5.59	10.80
scpc4	0.91	1.19	1.37	1.37	0.00†	0.00†	0.00	0.00	1.88	4.98	2.42	11.86	2.13	5.28	3.22	12.11
scpc5	0.47	0.93	0.93	1.40	0.00†	0.00†	0.00	0.00	0.19	4.23	1.67	9.33	0.98	5.39	4.20	11.13
scpd1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.20	2.27	1.58	3.63	0.17	0.73	0.74	1.02
scpd2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.74	4.09	3.09	9.44	0.19	0.47	0.41	0.75
scpd3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.11	2.59	1.86	4.22	0.38	1.21	0.81	3.55
scpd4	0.00†	0.00†	0.00	0.00	1.61	1.61	1.61	1.61	0.19	0.33	0.36	0.38	0.16	0.21	0.17	0.27
scpd5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.45	0.36	1.03	0.16	0.17	0.17	0.19
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	0.26	0.48			0.48	0.55			22.32	74.50			2.07	22.35		

†: Indicates the better value compared to its counterpart of the competing algorithm.

Table 7: Benchmarking Meta-RaPS and Meta-RaPS-LP on the crew scheduling instances (c).

	Optimality 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**
aa03	2.79	3.32	3.29	3.69	0.01†	0.01†	0.01	0.01	756.91	1534.99	1470.31	2593.55	0.20	0.79	0.36	2.64
aa04	1.76	2.85	2.51	3.40	0.17†	0.17†	0.17	0.17	53.94	979.97	1381.78	2869.19	0.17	0.22	0.19	0.30
aa05	1.41	2.06	1.59	2.55	0.00†	0.00†	0.00	0.00	522.33	2249.87	2540.32	3489.89	0.16	0.16	0.16	0.17
aa06	1.05	1.44	4.01	1.62	0.00†	0.00†	0.00	0.00	896.66	2322.95	1986.38	3409.23	0.77	3.44	3.28	7.19
aa11	6.40	6.78	5.78	7.01	0.23†	0.25†	0.23	0.27	185.30	1645.41	1679.58	3353.64	0.53	63.15	8.61	212.70
aa12	3.28	4.50	5.58	5.16	0.16†	0.16†	0.16	0.16	58.28	1300.88	1487.88	2561.73	15.78	142.50	64.47	451.63
aa13	6.00	6.92	6.11	7.68	0.00†	0.00†	0.00	0.00	931.70	1852.98	1737.63	3580.80	1.61	55.33	48.89	132.74
aa14	4.57	5.40	5.58	6.32	0.18†	0.18†	0.18	0.18	86.31	1986.37	1758.46	3571.50	3.63	49.88	28.17	160.17
aa15	4.68	5.68	5.91	6.25	0.29†	0.29†	0.29	0.29	127.83	1679.62	1362.01	3427.45	2.97	69.21	3.53	305.49
aa16	5.23	5.87	6.05	6.42	0.15†	0.22†	0.15	0.32	522.97	1623.84	2083.84	3507.72	0.64	40.06	17.14	101.34
aa17	5.10	6.11	5.49	6.99	0.00†	0.00†	0.00	0.00	524.36	2231.48	2196.32	3526.61	3.97	38.54	38.59	72.02
aa18	4.32	5.16	5.38	6.30	0.44†	0.44†	0.44	0.44	320.97	1542.76	1483.62	2218.59	16.83	346.01	84.44	803.69
aa19	5.16	6.06	5.24	6.80	0.00†	0.00†	0.00	0.00	183.42	1117.60	1498.50	2004.56	17.80	91.51	25.42	223.59
aa20	3.43	4.32	3.58	5.32	0.23†	0.23†	0.23	0.23	225.45	2049.34	1728.92	3074.01	4.80	128.92	56.22	471.81
bus1	3.44	3.52	3.61	3.60	0.00†	0.00†	0.00	0.00	387.17	1610.50	1452.43	3541.44	47.83	233.28	181.69	581.64
bus2	3.62	3.71	3.69	3.80	0.06†	0.07†	0.08	0.09	487.20	1898.61	1685.38	3425.78	159.84	1333.26	1198.50	2874.61
Average	3.89	4.61			0.12	0.13			391.93	1726.70			17.35	162.27		

$\dagger$: Indicates the better value compared to its counterpart of the competing algorithm.

Table 8: Benchmarking Meta-RaPS and Meta-RaPS-LP on the railway instances (d).

	Optimality 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	16.67	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	14.85	12.64	4.95†	6.7†	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	22.99	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	27.50	27.84	25.83	27.93	12.45†	13.49†	13.75	13.89	1277.07	2669.72	2020.84	5179.69	3.47	2709.39	1887.17	6199.66
rail2586	23.00	23.59	26.92	24.16	10.23†	10.42†	10.44	10.65	1298.19	2684.40	2779.23	5225.38	1873.67	3824.03	4332.13	5422.95
rail4284	29.67	29.93	27.39	30.23	11.74†	12.15†	12.21	12.49	2055.39	4626.27	5527.35	6339.89	5480.62	6147.49	6213.23	6767.01
rail4872	23.99	24.42	24.45	25.10	9.84†	10.29†	10.43	10.63	2407.99	4763.17	3165.07	7138.33	5.71	3547.47	4921.44	6024.72
Average	21.92	22.42			8.03	8.78			1025.54	3079.69			1303.68	3330.28		

†: Indicates the better value compared to its counterpart of the competing algorithm.

Table 9: Benchmarking Meta-RaPS and Meta-RaPS-LP on the hard cost and coverage correlated instances (e).

	Optimality 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**
rand1	1.88	2.13	2.00	2.50	1.00†	1.00†	1.00	1.00	530.69	935.91	750.05	1306.47	2.73	40.21	42.30	92.59
rand2	1.51†	1.93†	1.38	2.42	2.00	2.80	3.00	3.00	316.53	667.76	381.83	1415.11	4.27	39.46	28.66	113.80
rand3	0.63†	0.94†	1.69	1.25	1.00	1.00	1.00	1.00	128.06	779.37	855.63	2594.05	1.83	5.93	6.00	10.11
rand4	2.13	2.43	1.69	2.74	0.00†	0.00†	0.00	0.00	847.02	1442.95	1525.48	2053.47	8.30	23.48	21.73	40.41
rand5	0.63	0.94	1.04	1.25	0.00†	0.00†	0.00	0.00	1243.77	1608.31	1301.23	2350.05	76.77	107.90	101.00	142.34
rand6	0.68	1.18	0.60	1.36	0.00†	0.00†	0.00	0.00	182.27	895.98	545.48	2262.78	6.73	32.83	13.52	102.56
rand7	0.25	0.35	0.48	0.51	0.00†	0.00†	0.00	0.00	105.22	1353.26	909.94	3241.61	0.14	0.87	0.74	1.88
rand8	0.45	0.59	0.65	0.68	0.00†	0.00†	0.00	0.00	40.94	1093.76	1759.24	1806.88	0.13	0.68	0.36	1.86
rand9	0.43	0.60	0.67	0.86	0.00†	0.00†	0.00	0.00	8.52	2161.87	1722.50	3285.72	0.84	3.00	2.92	6.14
rand10	0.67	0.76	0.43	0.89	0.00†	0.00†	0.00	0.00	124.41	1060.36	1162.04	1883.95	0.48	1.60	0.61	5.24
rand11	0.18	0.18	0.09	0.18	0.00†	0.00†	0.00	0.00	335.44	1119.80	640.17	1674.34	0.11	0.23	0.22	0.41
rand12	0.00	0.00	0.35	0.00	0.00	0.00	0.00	0.00	18.56	416.58	640.17	818.19	0.13	0.18	0.14	0.25
rand13	0.69	0.97	0.53	1.22	0.00†	0.00†	0.00	0.00	277.45	1940.73	1752.61	3058.34	0.14	0.19	0.17	0.25
rand14	0.00	0.18	0.51	0.36	0.00	0.00†	0.00	0.00	8.14	1225.71	997.70	2812.19	0.25	2.56	1.06	7.20
rand15	0.65†	0.68†	4.03	0.81	1.00	1.00	1.00	1.00	434.23	1528.80	1141.18	3098.84	0.22	13.18	17.03	27.95
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
rand23	3.23	3.49	3.97	3.87	1.00†	1.00†	1.00	1.00	10.97	1576.56	1973.78	3244.76	77.88	919.92	531.06	2085.95
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
rand25	1.90	2.47	2.36	2.85	1.00†	1.00†	1.00	1.00	516.88	1355.67	1578.28	2555.55	0.17	96.98	110.06	193.67
rand26	2.12	2.31	1.94	2.59	0.00†	0.00†	0.00	0.00	97.66	1788.02	1697.34	2830.40	120.99	431.54	417.49	1000.09
rand27	0.75	1.31	1.03	1.76	0.00†	0.00†	0.00	0.00	251.89	1512.22	1093.04	2957.19	0.13	0.75	0.27	2.16
rand28	0.51	0.87	1.03	1.03	0.00†	2.00†	0.00	10.00	108.94	657.13	1072.68	1627.44	3.05	77.16	43.69	257.83
rand29	0.79	1.47	1.85	1.84	0.00†	1.00†	0.00	5.00	433.95	1935.24	1453.80	2898.61	0.38	1.01	0.98	2.19
rand30	1.86	2.34	2.39	2.65	0.00†	0.60†	0.00	3.00	446.23	1148.94	1375.43	2161.14	0.00	51.99	33.13	134.97
Average	2.22	2.62			0.53	0.69			323.58	1337.70			38.35	155.23		

†: Indicates the better value compared to its counterpart of the competing algorithm.