



ON A NEW CONSTRAINT REDUCTION HEURISTIC USING IMPROVED BISECTION METHOD FOR MIXED INTEGER LINEAR PROGRAMMING

HANDE GÜNAY AKDEMİR*

Giresun University, Faculty of Arts and Sciences, Department of Mathematics, Giresun, Turkey

*Corresponding author: hande.akdemir@giresun.edu.tr

ABSTRACT. In this study, we develop a surrogate relaxation-based procedure to reduce mixed-integer linear programming (MILP) problem sizes. This technique starts with one surrogate constraint which is a non-negative linear combination of multiple constraints of the problem. At this initial step, we calculate optimal Lagrangian multipliers from LP relaxation of the problem and use them as initial surrogate multipliers. We incorporate the improved bisection method (IBM) (B. Gavish, F. Glover, and H. Pirkul, Surrogate Constraints in Integer Programming, *J. Inform. Optim. Sci.* 12(2) (1991), 219–228.) into our algorithm. This simple heuristic algorithm is designed to iteratively generate a new surrogate cut that is to guarantee to satisfy the most violated two constraints of the corresponding iteration. The performance of the heuristic is tested using both some problems from the OR libraries and randomly generated ones.

1. INTRODUCTION

The objective here is to attempt to reduce the number of constraints of MILP type of problems. In the surrogate relaxation, a subset of constraints is substituted by a linear combination of these constraints, that is, we multiply these constraints with non-negative multipliers, and then aggregate them together to generate one combined constraint. With the optimal surrogate multipliers, this combined constraint serves as a proxy for the others and captures useful information. By doing that, we may obtain infeasible but near-optimal solutions while reducing processing time and complexity relative to the original model.

Received October 26th, 2020; accepted November 19th, 2020; published December 4th, 2020.

2010 *Mathematics Subject Classification.* 90C10, 90C27, 90C46, 49N15.

Key words and phrases. mixed integer linear programming; surrogate relaxation; surrogate constraints; Lagrangian multipliers; improved bisection method; heuristics; reduction.

©2021 Authors retain the copyrights of their papers, and all open access articles are distributed under the terms of the Creative Commons Attribution License.

Besides, the use of surrogate constraints is known to provide better bounds relative to Lagrangian bounds. In the proposed heuristic, as initial surrogate multipliers, we simply use the optimal dual values (Lagrangian multipliers) of the LP relaxation to get the *strongest* initial surrogate constraint. We incorporate the IBM [12], which is an updated version of the bisection algorithm [13], into our heuristic. At each iteration, the procedure evaluates two infeasible constraints at a time, and uses the IBM to find the optimal surrogate multipliers while upper (or lower) bound decreases (or increases). Furthermore, we generate not one but several surrogate constraints and only relax inequality constraints. So, we solve a sequence of computationally easy relaxed and reduced problems to reach the optimal solution in a small amount of time. Our proposed heuristic also enables us to examine which ratio of the problem constraints are redundant and which ones are critical. The heuristic is implemented in the MATLAB environment. In order to test the algorithm, we use problems available for download from Beasley's OR Library [4] and the Mixed Integer Programming Library [23]. We also conduct a set of testings using randomly generated instances. All of the computational experiments are done on a PC with an Intel Core i5-7400 CPU (3.00 GHz) and 4 GB of RAM.

Consider the following MILP problem:

$$(1.1) \quad \begin{aligned} (P) \quad & \min \quad c^T x \\ & \text{s.t.} \quad Ax \leq b \\ & \quad \quad x \in X \end{aligned}$$

where $c, x \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$ and X is a discrete set that may be defined by some linear equalities and bound constraints of the decision variables.

By moving the constraints into the objective function, we generate Lagrange relaxation:

$$(1.2) \quad \begin{aligned} & \min \quad c^T x + \lambda^T (Ax - b) \\ & \text{s.t.} \quad x \in X \end{aligned}$$

where Lagrange multipliers vector $\lambda \geq 0$.

Analogously, by assembling multiple constraints into a single new surrogate constraint, we generate surrogate relaxation:

$$(1.3) \quad \begin{aligned} & \min \quad c^T x \\ & \text{s.t.} \quad \mu^T (Ax - b) \leq 0 \\ & \quad \quad x \in X \end{aligned}$$

where surrogate multipliers vector $\mu \geq 0$.

Both techniques enlarge the feasible region and provide a lower bound on the optimal objective value of Problem (P). But, surrogate lower bound is tighter than the Lagrangian lower bound [14, 17].

Relaxation-based search or dual algorithms, both exact and heuristics, have been extensively used in finding bounds for integer programming (IP). Let us now survey the related literature which investigates surrogate relaxation-based heuristics to solve combinatorial optimization problems. In [15], the author proposed a class of surrogate constraint heuristics that provide a variety of supplementing new alternatives and independent solution strategies. Lorena and Narciso [31, 39] proposed six heuristics based on both the surrogate and Lagrangian relaxations and a subgradient search algorithm for large scale generalized assignment problems. The authors showed that the use of procedures based on surrogate constraint analysis is effective for satisfiability problem in [30]. Applications of a classical combinatorial optimization problem called the set covering were given in [1, 11]. The former used the Surrogate constraint normalization technique to create appropriate weights for surrogate constraint relaxations, and the latter compared heuristics based on Lagrangian and surrogate relaxations for the Maximal Covering Location Problem. Karabati et al. [22] handled a class of discrete problems with min-max-sum objectives by using line search and surrogate procedures to obtain optimal surrogate multipliers. The combined use of surrogate and Lagrangian relaxation was considered for traveling salesman problem in [26, 32, 36]. The reduction is a preprocessing technique that is fundamental for developing efficient integer programming methods and based on dynamic programming and upper bounds. For a comprehensive literature review of reduction techniques, see [3, 10, 40]. Riberio and Lorena [46] examined an IP problem called cartographic label placement by using Lagrangian/surrogate heuristics. For a detailed review on solution techniques based on surrogate relaxations for p -median and facility location problems, see [43]. For a review of the approaches which combine metaheuristics with exact IP techniques, see [42]. In [27], a critical event tabu search heuristic was presented to solve the multidimensional knapsack problems (MKPs) with generalized upper bound constraints. Osorio et al. [41] considered a combined cutting and surrogate constraint analysis strategy for MKPs, see also [19]. The authors introduced problem-size reduction heuristics for MKPs in [21]. For heuristics and metaheuristics for MKPs and their variants, see recent works of [2, 5, 6, 8, 18, 20, 24, 25, 28, 29, 33, 35, 47–50], and references therein. For graph theory applications, such as graph coloring, weighted maximum clique, and shortest path problems see [9, 16, 37, 38]. Choi and Choi [7] proposed a redundancy identification method that is based on surrogate constraints. The relaxation adaptive memory programming approach based on surrogate constraints was proposed for combinatorial optimization problems in [44] and for capacitated minimum spanning tree problems in [45]. To review the scatter search method which was conceived as an extension of surrogate constraint relaxation, see [34].

The rest of this paper is organized as follows. Section 2 gives a brief version of the IBM, followed by the proposed heuristic is introduced in Section 3. Section 4 describes computational experiments. Section 5 concludes the paper.

2. COMPUTING THE OPTIMAL SURROGATE MULTIPLIERS BY IBM

Now, we consider problems with only two constraints. We start by determining which constraint is tighter, namely, it produces a lower (greater) objective value for maximization (minimization) problem considering only one constraint and ignoring the other one. Renumerate the tighter constraint as Constraint 1, and the other as Constraint 2. The surrogate multiplier of Constraint 1 is constant and equals to 1. Other multiplier is μ which is going to be updated. The algorithm can be summarized as follows:

Step 1. Initial values of μ_L and μ_H are as following. The solution of the problem with multipliers $(1, \mu_L)$ and $(1, \mu_H)$ satisfies and does not satisfy Constraint 1, respectively.

Step 2. Let $\mu = (\mu_L + \mu_H)/2$. Solve the problem with the multipliers $(1, \mu)$. If

(i) both constraints are satisfied stop, $(1, \mu)$ is the optimal multipliers, and the optimal solution of the problem is obtained.

(ii) only Constraint 1 is satisfied let $\mu_L = f/g$ where f is the amount of oversatisfaction of Constraint 1 and g is the amount of undersatisfaction of Constraint 2.

(iii) only Constraint 2 is satisfied let $\mu_H = f/g$ where f is the amount of undersatisfaction of Constraint 1 and g is the amount of oversatisfaction of Constraint 2.

Step 3. If $\mu_L < \mu_H$ go to Step 2, otherwise Stop, $(1, \mu)$ is the optimal multipliers. Note that, we can also Stop if the objective function does not improve, or if the number of iterations reached an upper limit.

3. CONSTRAINT REDUCTION HEURISTIC

To determine the surrogate multipliers of the most violated two constraints of the current step, we apply the following iterative process which finds appropriate surrogate constraints.

Initial solution: We start by calculating optimal Lagrangian multipliers from LP relaxation of the problem and use them as initial surrogate multipliers. This surrogate constraint is our first surrogate constraint. Then, we solve our problem with this constraint and integrality restrictions.

Adding a new surrogate constraint (generating a cut): We first determine the most violated two constraints. Let

$$g_i(x) = \sum_{j=1}^n a_{ij}x_j - b_i, \quad i \in I = \{1, 2, \dots, m\},$$

and

$$\begin{aligned} \max_{i \in I} g_i(x) &= g_t(x), \\ \max_{i \in I - \{t\}} g_i(x) &= g_k(x). \end{aligned}$$

Add a new surrogate constraint as $\mu_t g_t(x) + \mu_k g_k(x) \leq 0$ where $\mu_t = \mu_k = 1$. Solve the problem adding this new surrogate constraint to prior surrogate constraint(s). If $g_t(x)$ and $g_k(x)$ are non-positive while $\mu_t = \mu_k = 1$, add another surrogate constraint. Otherwise, apply the IBM supposing the positive one as

Constraint 1. If both constraints are satisfied then add a new surrogate constraint. Repeat these steps until all constraints are satisfied (within a tolerance value). Pseudo-code is given in Algorithm 3.1.

4. COMPUTATIONAL RESULTS

This section presents the experimental results obtained with the proposed simple heuristic. Both test problems from the OR libraries [4, 23] and randomly generated ones are used to conduct the computational experiment. We only consider MILP problem cases which have non-negative Lagrange multipliers and at least one of them is non-zero. If this is not the case, we can assume that all of the initial surrogate multipliers are 1.

We denote the number of constraints by m , and the number of variables by n . For randomly generated 0-1 MKPs, we choose $m \in \{3000, 4000, 5000\}$ and $n \in \{150, 200\}$.

The constraint coefficients a_{ij} and objective function coefficients c_j are integers and randomly generated from the discrete uniform distributions $U\{1, 2, \dots, 500\}$ and $U\{1, 2, \dots, 100\}$, respectively. It is assumed that there are no correlations between the objective function and constraint coefficients. The right hand side values b_i are set to equal to

$$s_i \sum_{j=1}^n a_{ij},$$

where s_i is a slackness ratio and drawn from uniform distribution between 0.65 and 0.95. For each combination of (m, n) , we generate 10 problems. Table 1 gives, for each combination of $m \times n$, minimum, average (rounded off) and maximum computing times for both the intlinprog solver and the proposed heuristic. The heuristic performs better when the methods are compared by their average execution times which are less except for only one case. Table 1 indicates average execution times are reduced for almost every uncorrelated instance in which expected slackness ratio is 0.80.

In addition to the MKP instances, we test the proposed heuristic on a few instances from OR libraries [4, 23]. Refer to Table 2 for the results.

Note that, in all our calculations, the tolerance value are fixed to 10^{-6} , the algorithm terminates if the upper bound does not improve 30 times, and for the IBM, the iteration upper limit is set to 10.

5. CONCLUSIONS

In this paper, we try to provide equivalent formulations with a fewer number of constraints for combinatorial optimization problems. To do that, we present a surrogate cut generation procedure based on the IBM. Experiments performed on problems with lots of redundant constraints have shown that the proposed heuristic gives reduced models which can be solved significantly faster than original models.

Algorithm 3.1 Constraint Reduction Heuristic

INPUT: Coefficient matrices and vectors of the MILP Problem (P) (A, b, c , and the set X)

OUTPUT: Optimal integer solution x ; number of constraints k after reduction; constraint coefficient matrix Sur after reduction; Right hand side vector $SurRhs$ after reduction; Optimal surrogate weights matrix μ

- 1: Solve the LP relaxation of Problem (P) and calculate Lagrange multipliers, assign the corresponding Lagrange multipliers into the row vector μ ($\mu(1, i) = \lambda(i)$ for $i \in I = \{1, 2, \dots, m\}$),
- 2: $ConstNum \leftarrow$ the number of inequality constraints (m), $ToL \leftarrow 1e - 6$, $IterUpLim \leftarrow 10$, $lowBound1 \leftarrow 0$, $lowBound2 \leftarrow 0$, $w \leftarrow 0$ and $k \leftarrow 1$.
- 3: **function** SOLVESUR(A, b, c, X, μ)
- 4: Solve

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & \mu Ax \leq \mu b \\ & x \in X \end{array}$$
- 5: $e \leftarrow Ax - b$
- 6: **Return** x, e
- 7: **end function**
- 8: **function** LOWBOUND(k, c, x)
- 9: **if** k is odd **then**
- 10: $lowBound1 = c^T x$
- 11: **else**
- 12: $lowBound2 = c^T x$
- 13: **end if**
- 14: **Return** $lowBound1, lowBound2$
- 15: **end function**
- 16: SOLVESUR(A, b, c, X, μ)
- 17: LOWBOUND(k, c, x)
- 18: **while** $max(e) > ToL$ **and** $k \leq ConstNum$ **and** $w \leq 30$ **do**
- 19: **if** $lowBound1 == lowBound2$ **then**
- 20: $w ++$
- 21: **end if**
- 22: $r \leftarrow 0$
- 23: **for** $i \leftarrow 1, ConstNum$ **do**
- 24: $f(i, 1) = e(i)$
- 25: $f(i, 2) = i$
- 26: **end for**
- 27: Sort the rows of f in ascending order based on the elements in the first column and assign it to d
- 28: $ind1 = d(ConstNum, 2)$
- 29: $ind2 = d(ConstNum - 1, 2)$
- 30: $mLow = 0$
- 31: $mHigh = 1$
- 32: $k ++$
- 33: **if** $e(ind1) > ToL$ **and** $e(ind2) > ToL$ **then**
- 34: $\mu(k, ind1) = 1$
- 35: $\mu(k, ind2) = 1$
- 36: SOLVESUR(A, b, c, X, μ)

```

37:   if  $e(ind1) > 0$  and  $e(ind2) < 0$  then
38:     if  $e(ind1) > ToL$  then
39:        $\mu(k, ind2) = 0.5$ 
40:       SOLVESUR( $A, b, c, X, \mu$ )
41:       while ( $e(ind1) > ToL$  or  $e(ind2) > ToL$ ) and  $r \leq IterUpLim$  do
42:         if  $e(ind1) > 0$  and  $e(ind2) \neq 0$  then
43:            $mhigh = |e(ind1)/e(ind2)|$ 
44:           if  $mlow \leq mhigh$  then
45:              $\mu(k, ind2) = (mlow + mhigh)/2$ 
46:             SOLVESUR( $A, b, c, X, \mu$ )
47:              $r++$ 
48:             if  $mlow == mhigh$  then
49:               Break
50:             end if
51:           else
52:             Break
53:           end if
54:         else if  $e(ind2) > 0$  and  $e(ind2) \neq 0$  then
55:            $mhigh = |e(ind1)/e(ind2)|$ 
56:           if  $mhigh \leq mlow$  then
57:              $\mu(k, ind2) = (mhigh + mlow)/2$ 
58:             SOLVESUR( $A, b, c, X, \mu$ )
59:              $r++$ 
60:             if  $mhigh == mlow$  then
61:               Break
62:             end if
63:           else
64:             Break
65:           end if
66:         end if
67:         LOWBOUND( $k, c, x$ )
68:       end while
69:     end if
70:   else if  $e(ind2) > 0$  and  $e(ind1) < 0$  then
71:     if  $e(ind2) > ToL$  then
72:        $\mu(k, ind1) = 0.5$ 
73:       SOLVESUR( $A, b, c, X, \mu$ )
74:       LOWBOUND( $k, c, x$ )
75:       while ( $e(ind1) > ToL$  or  $e(ind2) > ToL$ ) and  $r \leq IterUpLim$  do
76:         if  $e(ind1) > 0$  and  $e(ind1) \neq 0$  then
77:            $mhigh = |e(ind2)/e(ind1)|$ 
78:           if  $mhigh \leq mlow$  then
79:              $\mu(k, ind1) = (mhigh + mlow)/2$ 
80:             SOLVESUR( $A, b, c, X, \mu$ )
81:              $r++$ 
82:             if  $mhigh == mlow$  then
83:               Break
84:             end if

```

```
85:         else
86:             Break
87:         end if
88:     else if  $e(ind2) > 0$  and  $e(ind1) \neq 0$  then
89:          $mhigh = |e(ind2)/e(ind1)|$ 
90:         if  $mlow \leq mhigh$  then
91:              $\mu(k, ind1) = (mlow + mhigh)/2$ 
92:             SOLVESUR( $A, b, c, X, \mu$ )
93:              $r++$ 
94:             if  $mlow == mhigh$  then
95:                 Break
96:             end if
97:         else
98:             Break
99:         end if
100:     end if
101:     LOWBOUND( $k, c, x$ )
102: end while
103: end if
104: end if
105: else if  $e(ind1) > Tol$  and  $e(ind2) < Tol$  then
106:      $\mu(k, ind1) = 1$ 
107:     SOLVESUR( $A, b, c, X, \mu$ )
108:     LOWBOUND( $k, c, x$ )
109: end if
110: end while
111:  $Sur = \mu A$ 
112:  $SurRhs = \mu b$ 
```

TABLE 1. Computational results of randomly generated 0-1 MKPs.

CPU time (seconds)									
$m \times n$	intlinprog			Heuristic			Number of Const.		
	min	ave	max	min	ave	max	min	ave	max
3000×150	85	232	520	34	145	538	17	33	62
3000×200	425	1050	3203	168	960	3860	31	40	60
4000×150	59	257	626	31	207	805	18	36	76
4000×200	66	528	1391	30	443	2043	20	28	41
5000×150	149	861	2051	21	942	2761	26	45	83
5000×200	670	2000	2994	209	1390	4296	27	42	58

TABLE 2. Computational results for instances from OR libraries [4, 23].

Problem Key	Number of			
	Inequality Constraints	Equality Constraints	Variables	Ineq. Const. After Reduction
ST1	30	0	60	13
ST2	30	0	60	14
PB6	30	0	40	13
PB7	30	0	37	15
f2gap40400	40	0	400	21
f2gap401600	40	0	1600	27
f2gap801600	80	0	1600	47
app2-1	1038	0	3283	2
app2-2	335	0	1226	1
supportcase14	127	107	304	4
misc07	177	35	206	136
blend2	185	89	353	64
bppc8-02	39	20	232	33
BeasleyC1	1250	500	2500	628

Data Availability: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The author declares that there are no conflicts of interest regarding the publication of this paper.

REFERENCES

- [1] J.H. Ablanedo-Rosas and C. Rego, Surrogate Constraint Normalization for the Set Covering Problem, *Eur. J. Oper. Res.* 205(3) (2010), 540–551.
- [2] B. Alidaee, V.P. Ramalingam, H. Wang, and B. Kethley, Computational Experiment of Critical Event Tabu Search for the General Integer Multidimensional Knapsack Problem, *Ann. Oper. Res.* 269(1-2) (2018), 3–19.
- [3] S. Balev, N. Yanev, A. Fréville, and R. Andonov, A Dynamic Programming based Reduction Procedure for the Multidimensional 0–1 Knapsack Problem, *Eur. J. Oper. Res.* 186(1) (2008), 63–76.
- [4] J. E. Beasley, OR-Library: Distributing Test Problems by Electronic Mail, *J. Oper. Res. Soc.* 41(11) (1990), 1069–1072.
- [5] V. Boyer, M. Elkihel, and D. El Baz, Heuristics for the 0–1 Multidimensional Knapsack Problem, *Eur. J. Oper. Res.* 199(3) (2009), 658–664.
- [6] M. Chih, Three Pseudo-utility Ratio-inspired Particle Swarm Optimization with Local Search for Multidimensional Knapsack Problem, *Swarm. Evol. Comput.* 39 (2018), 279–296.
- [7] J. Choi and I.C. Choi, Identifying Redundancy in Multi-dimensional Knapsack Constraints based on Surrogate Constraints, *Int. J. Comput. Math.* 91(12) (2014), 2470–2482.
- [8] C. D’Ambrosio, S. Martello, and L. Mencarelli, Relaxations and Heuristics for the Multiple Non-linear Separable Knapsack Problem, *Comput. Oper. Res.* 93 (2018), 79–89.
- [9] S.M. Douiri, M.B.O. Medeni, S. Elbernoussi, and E.M. Souidi, A New Steganographic Method for Grayscale Image using Graph Coloring Problem, *Appl. Math. Inform. Sci.* 7(2) (2013), 521–527.
- [10] A. Freville, The Multidimensional 0–1 Knapsack Problem: An Overview, *Eur. J. Oper. Res.* 155(1) (2004), 1–21.
- [11] R.D. Galvao, L.G.A. Espejo, and B. Boffey, A Comparison of Lagrangean and Surrogate Relaxations for the Maximal Covering Location Problem, *Eur. J. Oper. Res.* 124(2) (2000), 377–389.
- [12] B. Gavish, F. Glover, and H. Pirkul, Surrogate Constraints in Integer Programming, *J. Inform. Optim. Sci.* 12(2) (1991), 219–228.
- [13] B. Gavish and H. Pirkul, Efficient Algorithms for Solving Multiconstraint Zero-one Knapsack Problems to Optimality, *Math. Program.* 31(1) (1985), 78–105.
- [14] F. Glover, Surrogate Constraint Duality in Mathematical Programming, *Oper. Res.* 23(3) (1975), 434–451.
- [15] F. Glover, Heuristics for Integer Programming using Surrogate Constraints, *Decision Sci.* 8(1) (1977), 156–166.
- [16] F. Glover, Tutorial on Surrogate Constraint Approaches for Optimization in Graphs, *J. Heuristics.* 9(3) (2003), 175–227.
- [17] H.J. Greenberg and W.P. Pierskalla, Surrogate Mathematical Programming, *Oper. Res.* 18 (1970), 924–939.
- [18] B. Haddar, M. Khemakhem, S. Hanafi, and C. Wilbaut, A Hybrid Quantum Particle Swarm Optimization for the Multidimensional Knapsack Problem, *Eng. Appl. Artif. Intel.* 55 (2016), 1–13.
- [19] S. Hanafi and F. Glover, Exploiting Nested Inequalities and Surrogate Constraints, *Eur. J. Oper. Res.* 179(1) (2007), 50–63.
- [20] S. Hanafi and C. Wilbaut, Improved Convergent Heuristics for the 0-1 Multidimensional Knapsack Problem, *Ann. Oper. Res.* 183(1) (2011), 125–142.

- [21] R.R. Hill, Y. K. Cho, and J. T. Moore, Problem Reduction Heuristic for the 0–1 Multidimensional Knapsack Problem, *Comput. Oper. Res.* 39(1) (2012), 19–26.
- [22] S. Karabati, P. Kouvelis, and G. Yu, A Min-max-sum Resource Allocation Problem and its Applications, *Oper. Res.* 49(6) (2001), 913–922.
- [23] T. Koch, T. Achterberg, E. Andersen, et al., MIPLIB 2010, *Math. Program. Comput.* 3 (2) (2011), 103–163.
- [24] X. Kong, L. Gao, H. Ouyang, and S. Li, Solving Large-scale Multidimensional Knapsack Problems with a New Binary Harmony Search Algorithm, *Comput. Oper. Res.* 63 (2015), 7–22.
- [25] S. Laabadi, M. Naimi, H. El Amri, and B. Achchab, The 0/1 Multidimensional Knapsack Problem and Its Variants: A Survey of Practical Models and Heuristic Approaches, *Amer. J. Oper. Res.* 8(05) (2018), 395–439.
- [26] M. Lama and D. Pinto, A Preprocessing that Combines Heuristic and Surrogate Constraint Analysis to Fix Variables in TSP, In *Mex Int. Conf. Artif. I. (727–734)*. Springer, Berlin, Heidelberg, 2004, April.
- [27] V.C. Li and G.L. Curry, Solving Multidimensional Knapsack Problems with Generalized Upper Bound Constraints using Critical Event Tabu Search, *Comput. Oper. Res.* 32(4) (2005), 825–848.
- [28] Y. Li, O. Ergun, and G.L. Nemhauser, A Dual Heuristic for Mixed Integer Programming, *Oper. Res. Lett.* 43(4) (2015), 411–417.
- [29] C.J. Lin, M.S. Chern, and M. Chih, A Binary Particle Swarm Optimization based on the Surrogate Information with Proportional Acceleration Coefficients for the 0-1 Multidimensional Knapsack Problem, *J. Ind. Product. Eng.* 33(2) (2016), 77–102.
- [30] A. Lokketangen, and F. Glover, Surrogate Constraint Analysis-New Heuristics and Learning Schemes for Satisfiability Problems, In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 35 (1997), 537–572.
- [31] L.A.N. Lorena and M.G. Narciso, Relaxation Heuristics for a Generalized Assignment Problem, *Eur. J. Oper. Res.* 91(3) (1996), 600–610.
- [32] L.A.N. Lorena and M.G. Narciso, Using Logical Surrogate Information in Lagrangean Relaxation: An Application to Symmetric Traveling Salesman Problems, *Eur. J. Oper. Res.* 138(3) (2002), 473–483.
- [33] S. Martello and M. Monaci, Algorithmic Approaches to the Multiple Knapsack Assignment Problem, *Omega* 90 (2020), Art. ID 102004.
- [34] R. Marti, A., Corberan, and J. Peiro, Scatter Search, In *Handbook of Heuristics*, 1–24 (2016).
- [35] T. Meng and Q.K. Pan, An Improved Fruit Fly Optimization Algorithm for Solving the Multidimensional Knapsack Problem, *Appl. Soft Comput.* 50 (2017), 79–93.
- [36] F. Molina, M.O.D. Santos, F. Toledo, and S.A.D. Araujo, An Approach using Lagrangian/surrogate Relaxation for Lot-sizing with Transportation Costs, *Pesquisa Oper.* 29(2) (2009), 269–288.
- [37] A. Moumen, M. Bouye, and H. Sissaoui, New Secure Partial Encryption Method for Medical Images using Graph Coloring Problem, *Nonlinear Dynam.* 82(3) (2015), 1475–1482.
- [38] A. Nagih and F. Soumis, Nodal Aggregation of Resource Constraints in a Shortest Path Problem, *Eur. J. Oper. Res.* 172(2) (2006), 500–514.
- [39] M.G. Narciso and L.A.N. Lorena, Lagrangean/surrogate Relaxation for Generalized Assignment Problems, *Eur. J. Oper. Res.* 114(1) (1999), 165–177.
- [40] S.M. Neuro and J.M. Pinto, Decomposition Techniques for the Long-range Production Planning of Oil Complexes, In *Comput. Aided Chem. Eng.* 18 (2004), 967–972.
- [41] M.A. Osorio, F. Glover, and P. Hammer, Cutting and Surrogate Constraint Analysis for Improved Multidimensional Knapsack Solutions, *Ann. Oper. Res.* 117(1-4) (2002), 71–93.

-
- [42] G.R. Raidl and J. Puchinger, Combining (Integer) Linear Programming Techniques and Metaheuristics for Combinatorial Optimization, In *Hybrid Metaheuristics (31–62)*, Springer, Berlin, Heidelberg, 2008.
- [43] J. Reese, Solution Methods for the p -median Problem: An Annotated Bibliography, *NETWORKS*. 48(3) (2006), 125–142.
- [44] C. Rego, RAMP: A New Metaheuristic Framework for Combinatorial Optimizatio, In *Metaheuristic Optimization via Memory and Evolution (441-460)*, Springer, Boston, MA, 2005.
- [45] C. Rego, F. Mathew, and F. Glover, Ramp for the Capacitated Minimum Spanning Tree Problem, *Ann. Oper. Res.* 181(1) (2010), 661–681.
- [46] G.M. Ribeiro and L.A.N. Lorena, Heuristics for Cartographic Label Placement Problems, *Comput. Geosci.* 32(6) (2006), 739–748.
- [47] A. Sakalliglu, Surrogate Constraint Applications to Network Models in Operations Research, Master Thesis, Department of Mathematics, Giresun University, 2019.
- [48] F. Taniguchi, T. Yamada, and S. Kataoka, Heuristic and Exact Algorithms for the Max–min Optimization of the Multi-scenario Knapsack Problem, *Comput. Oper. Res.* 35(6) (2008), 2034–2048.
- [49] J. Wang, T. Liu, K. Liu, B. Kim, J. Xie, Z. Han, Computation Offloading Over Fog and Cloud Using Multi-Dimensional Multiple Knapsack Problem, in: *2018 IEEE Global Communications Conference (GLOBECOM)*, IEEE, Abu Dhabi, United Arab Emirates, 2018: pp. 1–7.
- [50] B. Zhang, Q.K. Pan, X.L. Zhang, and P.Y. Duan, An Effective Hybrid Harmony Search-based Algorithm for Solving Multidimensional Knapsack Problems, *Appl. Soft. Comput.* 29 (2015), 288–297.