Management Science and Financial Engineering
Vol 18, No 2, November 2012, pp.13-17
ISSN 2287-2043 │ EISSN 2287-2361 │

http://dx.doi.org/10.7737/MSFE.2012.18.2.013
© 2012 KORMS

General Set Covering for Feature Selection in Data Mining

Zhengyu Ma
School of Industrial and Management Engineering, Korea University
Hong Seo Ryoo*
School of Industrial and Management Engineering, Korea University

(Received: May 26, 2012 / Revised: June, 14 2012 / Accepted: June 19, 2012)

ABSTRACT

Set covering has widely been accepted as a staple tool for feature selection in data mining. We present a generalized version of this classical combinatorial optimization model to make it better suited for the purpose and propose a surrogate relaxation-based procedure for its meta-heuristic solution. Mathematically and also numerically with experiments on 25 set covering instances, we demonstrate the utility of the proposed model and the proposed solution method.

Keywords: General Set Covering, Feature Selection, Surrogate Relaxation

* Corresponding Author, E-mail: hsryoo@korea.ac.kr

1. INTRODUCTION

Let us define a general set covering problem (henceforth, (gSC)) with multiplicity k as the problem of covering each of the m 0-1 constraints k times at minimum cost. That is, we consider

$$\text{(gSC)} \qquad \min\left\{\mathbf{c}^{\mathrm{T}}\mathbf{x} : \mathbf{A}\mathbf{x} \geq \mathbf{k},\ \mathbf{x} \in \{0, 1\}^n\right\},$$

where $\mathbf{c}$ is a positive integer vector of order n, $\mathbf{A}$ is a matrix of 0s and 1s, and $\mathbf{k}^{\mathrm{T}} = [k, k, \cdots, k]$, where k is an integer larger than or equal to 1.When $k = 1$, (gSC) reduces to a standard set covering (SC) problem, which has been extensively researched in the literature. When $\mathbf{k} \in \boldsymbol{R}^m_+$ and $\boldsymbol{c} \in \boldsymbol{R}^n_+$, (gSC) generalizes to a problem known as the multi-covering problem (e.g., Dantzig, 1954; Hall and Hochbaum, 1979).

In data mining, a need arises for identifying a small subset of the features, called support features, that describes the data on hand without contradiction. In brief, the features in data mining take the role of decision variables when the classification problem is formulated as an optimization problem. Therefore, the degree of difficulty associated with finding an accurate classification theory is closely related to the number of support features used. For support feature selection, which is a combinatorial optimization problem in nature, set covering has widely been accepted as a staple tool (e.g., Boros *et al*., 1997; Boros *et al*., 2000), and the common practice is to apply SC k times in succession to ensure selecting small but enough number of features for an accurate analysis of data on hand (e.g., Boros *et al*., 2000; Kim and Ryoo, 2008; Ryoo and Jang, 2009). In short, support features are the building blocks of a classification theory and, therefore, the quality of a classification theory can be said to be only as good as the quality of the features selected for use. We refer the reader to Boros *et al*. (1997) for the importance and a more background on feature selection in data mining and the utility of SC for this combinatorial optimization problem.

Since the introduction in Dantzig (1954), multiset covering has received little attention from the OR society. As seen, however, (gSC) selects support features based on combinatorial interplay among all variables, hence is better suited for the purpose of support feature selection in data mining than the standard approach of using SC k times successively (e.g., Boros *et al*., 2000; Kim and Ryoo, 2008; Ryoo and Jang, 2009).

In this short paper, we propose (gSC) for support feature selection and illustrate its utility over the standard approach (henceforth, (sSC)) in Section 2. For solving (gSC), we propose the standard subgradient optimization scheme, based on surrogate relaxation techniques, and show that the surrogate dual provides tighter lower bounds than the Lagrangian counterpart in Section 3. In Section 4, with experiments on 25 set covering instances from the OR Library (e.g., Beasley, 1990),we demonstrate the utility of (gSC) over (sSC) and also the use of the surrogate relaxation-based solution method for solving (gSC).

2. UTILITY OF (GSC) FOR FEATURE SELECTION

Support feature selection in data mining is a combinatorial optimization problem (e.g., Boros *et al*., 1997). In this section, with two small examples, we illustrate the usefulness of (gSC) for support feature selection, in comparison with (sSC), the standard approach in the literature (e.g., Boros *et al*., 2000; Kim and Ryoo, 2008; Ryoo and Jang, 2009).

2.1 Optimal vs. Non-optimal Solutions

Consider the feature selection instance in Table 1 with $k = 2$ and the unicost vector $\mathbf{c} = \mathbf{1}$.

Table 1. SC Matrix for Example in Section 2.1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Const. 1	0	0	0	0	1	1	1	1	0	0
Const. 2	1	1	0	1	0	0	1	0	0	0
Const. 3	1	1	0	0	1	0	0	0	1	0
Const. 4	0	1	1	1	0	0	0	1	0	0
Const. 5	0	0	1	0	0	1	1	0	1	1

When the corresponding (gSC) instance is solved, we obtain an optimal solution $x_2 = x_3 = x_5 = x_7 =1$ with the objective value 4.To help the reader with understanding, $x_j = 1$ indicates that feature j is selected. When (sSC) is used, the SC instance with the right-hand side vector of 1s is solved the first time with all features considered. For this SC instance, we obtain $x_2 = x_6 =1$ as an optimal solution. When the SC instance is solved the second time, only the columns 1, 3, 4, 5, 7, 8, 9 and 10 are considered, and we obtain $x_1 = x_3 = x_7 =1$ as an optimal solution. Now, putting the two solutions together, we obtain via (sSC) a solution $x_1 = x_2 = x_3 = x_6 = x_7 = 1$ with the objective value 5, which is just a feasible solution for (gSC). From this, and also comparing the solutions by (gSC) and (sSC) while paying close attention to the dissimilarity between them, one notes that the standard feature selection approach in data mining is not truly combinatorial optimization-based and can easily see the benefit of using (gSC) for support feature selection in data mining.

2.2 Feasibility vs. Infeasibility

Consider the instance in Table 2 with the objective coefficient vector $\mathbf{c}^{\mathrm{T}} = [2\,1\,1\,1\,1\,1\,1]$ and $k = 2$.

When the corresponding (gSC) instance is solved, we obtain $x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = 1$ with the objective value 7. When (sSC) is used, we obtain $x_2 = x_3 = x_4 = 1$ from the first instance of SC but find that the second SC instance without these three columns admits no feasible solution.

Table 2. SC Matrix for Example in Section 2.2

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Const. 1	1	0	0	1	0	0	0
Const. 2	1	0	0	1	0	0	0
Const. 3	0	1	0	1	0	0	0
Const. 4	0	1	0	0	1	0	0
Const. 5	0	0	1	0	0	1	0
Const. 6	1	0	1	0	0	0	1

In summary, the two examples of this section illustrate the utility of (gSC) for feature selection in data mining well.

3. LAGRANGIAN AND SURROGATE RELAXATIONS

3.1 Review and Duality Results

The Lagrangian and surrogate relaxations are respectively defined for (gSC) for any $\boldsymbol{\lambda} \in \mathbf{R}_{+}^{m}$ as

$$\mathrm{L}(\boldsymbol{\lambda}) = \min\left\{(\mathbf{c}^{\mathrm{T}} - \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{A})\mathbf{x} + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{k} : \mathbf{x} \in \{0, 1\}^{n}\right\}$$

and

$$\mathrm{S}(\boldsymbol{\lambda}) = \min\left\{(\mathbf{c}^{\mathrm{T}}\mathbf{x} : \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{A}\mathbf{x} \geq \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{k},\ \mathbf{x} \in \{0, 1\}^{n}\right\}.$$

For any $\boldsymbol{\lambda} \in \mathbf{R}_{+}^{m}$, L($\boldsymbol{\lambda}$) is fast solved by examining the signs of $(\mathbf{c}^{\mathrm{T}}-\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{A})$ and fixing $x_j = 1$ if $(c_j-\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{A}^j) \leq 0$ and 0 otherwise, where $\mathbf{A}^j$ denotes the column in $\mathbf{A}$ corresponding to x_j. S($\boldsymbol{\lambda}$) is also fast solved by fixing columns in the increasing order of $c_j/\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{A}^j$ values until the condition $\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{A}\mathbf{x} \geq \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{k}$ is satisfied (e.g., Beasley, 1990; Lorena and Lopes, 1994). Therefore, any of the two relaxations can be used for (gSC).

After a close examination, however, one notes that S($\boldsymbol{\lambda}$) handles (gSC) more relevantly than L($\boldsymbol{\lambda}$) for the general case $k > 1$. In short, when $L(\boldsymbol{\lambda})$ is used for (gSC), the same columns are selected and fixed at 1 regardless of the value of k, as $(\mathbf{c}^T-\boldsymbol{\lambda}^T\mathbf{A})$ is not affected by the value of k. This is not the case with $S(\boldsymbol{\lambda})$ as additional columns are needed to satisfy the feasibility condition $\boldsymbol{\lambda}^T\mathbf{Ax} \geq \boldsymbol{\lambda}^T\mathbf{k}$ when a larger value of k is used. This shows that the surrogate relaxation is better suited for (gSC) than the Lagrangian relaxation.

Furthermore, it is well-known that the surrogate relaxation provides tighter lower bounds than the Lagrangian counterpart (e.g., Greenberg and Pierskalla, 1979). To see this, let $v(\bullet)$ denote the optimal value of an optimization problem $\bullet$ and define the Lagrangian dual of (gSC) as

$$v(L) = \max\{L(\boldsymbol{\lambda}):\ \boldsymbol{\lambda} \in \mathbf{R}_+^m\}$$

and the surrogate dual of (gSC) as

$$v(S) = \max\{S(\boldsymbol{\lambda}):\ \boldsymbol{\lambda} \in \mathbf{R}_+^m\}.$$

Proposition: For any $\boldsymbol{\lambda} \in \mathbf{R}_+^m$, $S(\boldsymbol{\lambda}) \geq L(\boldsymbol{\lambda})$.

Proof: We have

$$\begin{aligned} L(\boldsymbol{\lambda}) &= \min\ \{\mathbf{c}^T\mathbf{x}+\boldsymbol{\lambda}^T(\mathbf{k}-\mathbf{Ax})\} \\ &\leq \min\ \{\mathbf{c}^T\mathbf{x}+\boldsymbol{\lambda}^T(\mathbf{k}-\mathbf{Ax}):\ \boldsymbol{\lambda}^T(\mathbf{k}-\mathbf{Ax}) \leq \mathbf{0}\} \\ &\leq \min\ \{\mathbf{c}^T\mathbf{x} : \boldsymbol{\lambda}^T(\mathbf{k}-\mathbf{Ax}) \leq \mathbf{0}\} \\ &= S(\boldsymbol{\lambda}) \end{aligned}$$

for (gSC) that is in minimization form. ■

Corollary: The surrogate dual provides a tighter lower bound for (gSC) than the Lagrangian dual; that is, $v(S) \geq v(L)$.

3.2 Proposed Algorithm

Based on results of the previous subsection, we propose the surrogate method for (gSC). For comparative experiments of the following section, however, we sketch two algorithms based on the Lagrangian as well as surrogate relaxations for lower bounding but with the same subgradient optimization framework and a primal heuristic procedure for building a cover (a feasible solution to (gSC)) on the dual solution provided by the two relaxation methods. These algorithms are summarized below.

```
procedure Lagrangian and surrogate (proposed) method
  - initialize UB = +∞, LB = -∞ and $\lambda_i = \min_{j\in J_i}\{c_j/|I_j|\}$, $\forall i$
  - set iteration counter to 1.
while (UB–LB > 1 or iteration time < 5,000) do
  - solve L(λ) or S(λ).
  - if L(λ) or S(λ) is larger than LB, update LB.
  - construct a cover with by the primal-heuristic.
  - if the cover found provides a lower UB, update UB.
  - update λ by the subgradient method.
  - increase the iteration counter to 1 larger.
end while
```

In the algorithms above, UB and LB indicate the upper and lower bounds on the optimum of (gSC), J_i denotes the index set of the features that covers constraint i, and I_j denotes the index set of the constraints that selecting feature j (that is, setting $x_j = 1$) covers. The initialization scheme for $\boldsymbol{\lambda}$ owes to Beasley (1987) and Caprara *et al.* (1999), and the primal heuristic we use is (a simple modification of) the textbook greedy heuristic(e.g., Chvatal, 1979; Nemhauser and Wolsey, 1999) that selects additional columns one at a time by the minimum ratio of $c_j/(I_j \cap M_u)$ until every constraint is covered at least k times, where M_u denotes the index set of currently uncovered constraints. For more details on the Lagrangian and surrogate duals and/or the subgradient optimization, we refer the reader to the aforementioned references. Last, we recall that, when $k = 1$, (gSC) reduces to the standard SC model, hence the procedure above become one for solving SC instances.

4. NUMERICAL EXPERIMENTS: PRELIMINARYRESULTS AND DISCUSSIONS

In this section, we demonstrate the efficacy of (gSC) over (sSC),the standard feature selection approach in the literature, and compare the performance of the Lagrangian and surrogate methods for (gSC).For experiments, we used 25 small to medium size test SC problems from Beasley (1990); namely, scp41-scp410, scp51-scp510 and scp61-scp65.We randomly chose and used $k = 2$ for (sSC) and (gSC) in these experiments and solved all instances of SC (for (sSC)) and (gSC) on an Intel i5 2-core notebook with 2.6 GB of memory. C++ was used for all implementations. We refer the reader to Beasley(1990) for more information on the 25 SC problems.

4.1 (sSC) vs. (gSC)

Table 3 summarizes the results of by the surrogate method for (sSC) and (gSC) and presents the better of the two results by the two set covering approaches for UB and the gap (UB-LB) for each instance in bold numbers for easy referencing.

In this table, 'Iter'. indicates when the best solution yielding the UB was found, 'Gap' is the difference between the best UB and LB, and 'Time' is the CPU seconds spent for solving the corresponding problem. Here, we mean by 'solving' when the gap is closed or the maximum 5000 iteration limit is reached for the problem.

The results in Table 3 support that (gSC) is superior to (sSC) by all measures used to compare their performance. The only exception is the time; as one can see, (gSC) requires a little more time, but the difference does not seem to be significant, statistically.

Table 3. Comparison of (sSC) and (gSC)

Data	(sSC)					(gSC)				
	LB	UB	Iter.	Gap	Time	LB	UB	Iter.	Gap	Time
Scp41	1301.56	1316	650	14	1.4	1139.71	**1150**	2417	**10**	1.7
Scp42	1377.90	1394	3687	16	1.4	1204.65	**1205**	301	**0**	0.2
Scp43	1376.76	1403	174	26	1.4	1205.84	**1215**	2997	**9**	1.6
Scp44	1340.00	1360	1213	19	1.7	1182.69	**1185**	983	**2**	1.7
Scp45	1415.02	1464	2941	48	1.4	1260.53	**1267**	2402	**6**	1.6
Scp46	1539.51	1616	1807	76	1.7	1343.05	**1349**	1257	**5**	1.7
Scp47	1277.77	1310	3428	32	1.5	1114.06	**1115**	123	**0**	1.0
Scp48	1368.60	1404	3222	35	1.7	1210.13	**1232**	3548	**21**	1.6
Scp49	1608.57	1628	1423	19	1.7	1484.02	**1485**	251	**0**	0.2
Scp410	1522.66	1533	4274	10	1.7	1353.41	**1356**	1302	**2**	1.6
Scp51	656.74	675	1449	18	3.2	577.32	**579**	2875	**1**	3.1
Scp52	745.79	759	1102	13	3.2	666.44	**685**	2519	18	3.1
Scp53	614.49	626	3934	11	2.7	569.44	**579**	2239	**9**	3.2
Scp54	663.25	685	1962	21	3.2	576.46	**590**	4148	**13**	3.1
Scp55	627.21	629	186	1	0.2	548.01	**550**	249	**1**	3.1
Scp56	633.94	642	638	8	2.7	556.16	**561**	3044	**4**	3.2
Scp57	810.34	818	3770	7	3.2	691.33	**696**	1578	**4**	3.1
Scp58	739.57	754	4018	14	3.1	659.95	**664**	1542	**4**	3.1
Scp59	778.75	785	1237	6	3.2	679.76	**690**	809	10	3.1
Scp510	729.36	747	3673	17	2.6	669.16	**677**	2200	**7**	3.1
Scp61	344.84	370	1015	25	2.2	276.20	**283**	663	**6**	2.2
Scp62	359.44	387	1645	27	2.2	296.47	**302**	143	**5**	2.3
Scp63	372.08	406	1555	33	2.2	308.92	**313**	1333	**4**	2.3
Scp64	345.36	365	1287	19	2.2	285.86	**294**	56	**8**	2.3
Scp65	390.51	419	2311	28	2.2	346.43	**359**	1718	**12**	2.3

4.2 Lagrangian vs. Surrogate Methods

We summarize the results by the Lagrangian and surrogate methods for (gSC) in Table 4 and indicate the better of the results by the two relaxation methods for LB, UB and the gap between them for each instance in bold numbers.

Table 4. Comparison of Lagrangian and Surrogate Methods

Data	(sSC)					(gSC)				
	LB	UB	Iter.	Gap	Time	LB	UB	Iter.	Gap	Time
Scp41	1137.86	1154	2267	16	1.0	**1139.71**	1151	1788	**11**	2.0
Scp42	1203.44	1205	2371	1	1.1	**1204.59**	1205	216	**0**	0.3
Scp43	1203.65	**1213**	4817	9	1.0	**1205.68**	1214	726	**8**	2.0
Scp44	1181.04	1185	1234	3	1.0	**1182.70**	1185	167	**2**	2.0
Scp45	1259.35	1267	1639	7	1.0	**1260.49**	1267	670	**6**	2.0
Scp46	1340.58	1350	1550	9	1.1	**1343.05**	**1349**	1694	**5**	2.0
Scp47	1112.59	1115	894	2	1.0	**1114.02**	1115	75	**0**	1.4
Scp48	1208.93	1231	3514	22	1.1	**1210.06**	**1230**	2451	**19**	2.0
Scp49	1482.46	1485	1203	2	1.0	**1484.12**	1485	267	**0**	0.4
Scp410	1351.36	1356	1179	4	1.0	**1353.18**	1356	718	**2**	2.0
Scp51	576.86	579	1159	2	1.6	**577.32**	579	1578	**1**	3.6
Scp52	665.82	683	876	17	1.7	**666.44**	**681**	3882	**14**	3.5
Scp53	568.78	577	3552	8	1.7	**569.34**	**575**	1308	**5**	3.7
Scp54	576.17	**588**	1045	**11**	1.7	**576.46**	590	3162	13	3.7
Scp55	546.79	550	684	3	1.4	**548.02**	550	60	**1**	3.6
Scp56	555.84	**560**	2130	4	1.6	**556.16**	561	898	**4**	3.7
Scp57	690.61	695	2314	4	1.5	**691.33**	695	1049	**3**	3.7
Scp58	659.22	664	971	4	1.5	**659.95**	664	682	**4**	3.9
Scp59	679.02	688	3997	8	1.6	**679.76**	**687**	2455	**7**	3.7
Scp510	668.40	673	996	4	1.6	**669.16**	**672**	2082	**2**	4.4
Scp61	**276.24**	283	1785	6	1.9	**276.20**	283	287	6	2.5
Scp62	296.46	302	1697	5	1.9	**296.46**	302	104	5	3.5
Scp63	308.93	313	2145	4	1.8	**308.91**	313	553	4	2.8
Scp64	285.73	294	4505	8	1.8	**285.86**	294	299	8	2.8
Scp65	346.35	354	3389	7	2.0	**346.39**	354	63	7	2.6

First, we see in this table that the surrogate method provides better lower bounds for 23 (gSC) instances and better upper bounds for 7 (gSC) instances while the Lagrangian method provides better UB for 3 instances. When the gaps are compared, the surrogate method performs better or equally well on all but one instance. Mathematically, we showed in Section 3.1 that the surrogate relaxations provide tighter lower bounds than the Lagrangian counterparts. The numerical evidence in Table 4 supports this and illustrates other advantages of the surrogate method that are related to this.

In terms of CPU time, the Lagrangian method seems to be about twice more efficient than its counterpart with an exception of the four instances for which the surrogate dual closed the gap, hence terminated early. As for the time efficiency of the Lagrangian method, we wish to numerically investigate it in our future research.

In terms of CPU time, the Lagrangian method seems to be about twice more efficient than its counterpart with an exception of the four instances for which the surrogate dual closed the gap, hence terminated early. As for the time efficiency of the Lagrangian method, we wish to numerically investigate it in our future research.

Another advantage of the surrogate dual is that it reduces the gap between the upper and lower bounds much faster and after a much smaller number of iterations. To show this, we plot in Figure 1~Figure 3 the gap against the number of iterations for instances scp46, scp54 and scp65 by the two relaxation methods. Note in Table 4 that the gap was smaller for the surrogate method for scp46, smaller for the Lagrangian method for scp54 and the same for both for scp65. In short, regardless of which relaxation performed better, the surrogate dual seems to bring the gap closer much faster than its counterpart. This may indicate that, by means of a more effective use of the surrogate relaxation and/or, perhaps, by using it as a prelude to using the Lagrangian relaxation, one may develop a scheme that can solve the (general) set covering problems more efficiently. This investigation makes up an interesting topic of study for our future research.

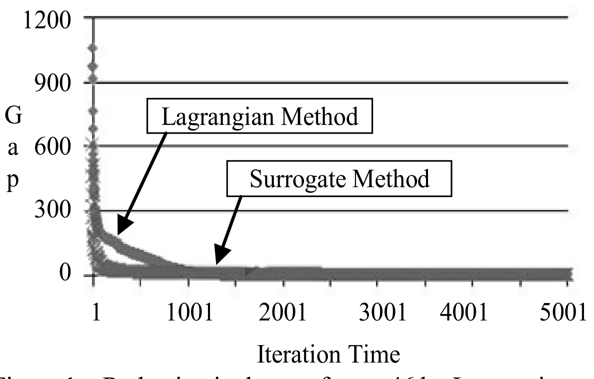

Figure 1. Reduction in the gap for scp46 by Lagrangian and surrogate methods

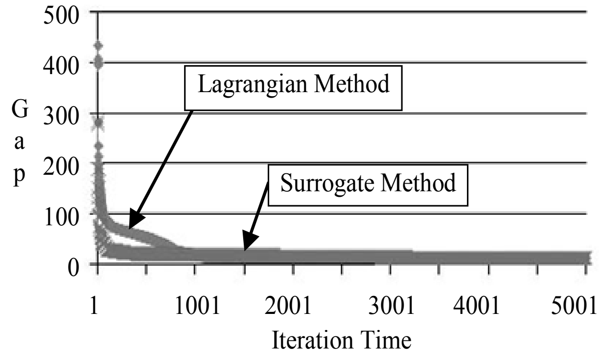

Figure 2. Reduction in the gap for scp54 by Lagrangian and surrogate methods

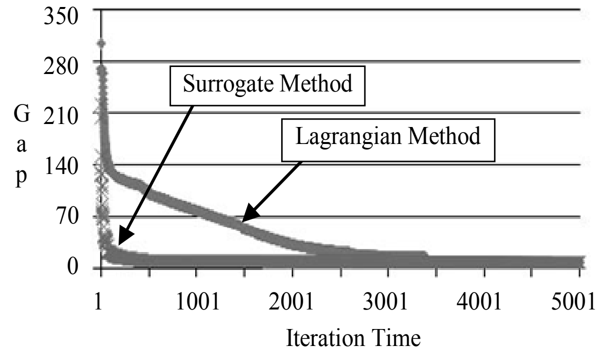

Figure 3. Reduction in the gap for scp65 by Lagrangian and surrogate methods

5. CONCLUSIONS

Set covering has extensively been studied in the literature for its applications in real-life decision. In particular, in data mining, the selection of a small number of important features (variables) holds a key to a successful analysis of the data on hand, and set covering has been accepted as a staple device for this purpose.

In this paper, we considered a generalized version of set covering called the general set covering and showed that it can be usefully used for feature selection in data mining. Next, we developed a surrogate relaxation-based method for meta-heuristic solution of the general set covering problems and illustrated the utility of the general set covering and the proposed meta-solution procedure both mathematically and numerically.

As for future research, we will investigate how the surrogate relaxation may be used as a stand-alone method or within a Lagrangian scheme to develop a more efficient and effective meta-solution procedure for general set covering problems.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (Grant Number: 2010-0016571).

REFERENCES

Beasley, J. E., "An Algorithm for Set Covering Problem," *European Journal of Operational Research* 31 (1987), 85-93.

Beasley, J. E., "A Lagrangian Heuristic for Set-covering Problems," *Naval Research Logistics* 37 (1990), 151-164.

Beasley, J. E., "OR-Library: Distributing Test Problems by Electronic Mail," *Journal of the Operational Research Society* 41 (1990), 1069-1072.

Boros, E., P. L. Hammer, T. Ibaraki, and A. Kogan, "Logical Analysis of Numerical Data," *Mathematical Programming* 79 (1997), 163-190.

Boros, E., P. L. Hammer, T. Ibaraki, and A. Kogan, "An Implementation of Logical Analysis of Data," *IEEE Transactions on Knowledge and Data Engineering*, 12, 2 (2000), 292-306.

Caprara, A., M. Fischetti, and P. Toth, "A Heuristic Method for the Set Covering Problem," *Operations Research* 45 (1999), 730-743.

Chvatal, V., "A Greedy Heuristic for the Set-covering Problem," *Mathematics of Operations Research* 4 (1979), 233-235.

Dantzig, G. B., "A Comment on Edie's Traffic Delay at Toll Booths," *Journal of the Operations Research Society of America* 2, 3 (1954), 339-341.

Greenberg, H. J. and W. P. Pierskalla, "Surrogate Mathematical Programming," *Operations Research* 18, 5 (1979), 924-939.

Hall, N. G. and D. S. Hochbaum, "A Fast Approximation Algorithm for the Multicovering Problem," *Discrete Applied Mathematics* 15 (1986), 35-40.

Kim, K. and H. S. Ryoo, "A LAD-based Method for Selecting Short Oligo Probes for Genotyping Applications," *OR Spectrum: Special Issue on OR and Biomedical Informatics* 30 (2008), 249-268.

Lorena, L. A. N. and F. B. Lopes, "A Surrogate Heuristic for Set Covering Problems," *European Journal of Operational Research* 79 (1994), 38-150.

Nemhauser, G. L. and L. A. Wolsey, "Integer and Combinatorial Optimization," A Wiley-Interscience Publication, John Wiley and Sons INC., USA, 1999.

Ryoo, H. S. and I.-Y. Jang, "MILP Approach to Pattern Generation in Logical Analysis of Data," *Discrete Applied Mathematics* 157 (2009), 749-761.