ISSN 1392-8619 print/ISSN 1822-3613 online
ŪKIO TECHNOLOGINIS IR EKONOMINIS VYSTYMAS
TECHNOLOGICAL AND ECONOMIC DEVELOPMENT OF ECONOMY
http://www.tede.vgtu.lt
2006, Vol XII, No 1, 18–22

LARGE-SCALE SET PARTITIONING PROBLEMS: SOME REAL-WORLD INSTANCES HIDE A BENEFICIAL STRUCTURE

Luca Coslovich[1], Raffaele Pesenti[2], Walter Ukovich[3]

[1,3]Department of Electrical Engineering, Electronics and Computer science, University of Trieste, Via A. Valerio 10, 34127 Trieste, Italy. E-mail: [coslovic, ukovich]@units.it
[2]Department of Computer Science Engineering, University of Palermo, Viale delle Scienze, 90128 Palermo, Italy. E-mail: pesenti@unipa.it

Received 3 October 2005; accepted 4 January 2006

Abstract. In this paper we consider large-scale set partitioning problems. Our main purpose is to show that real-world set partitioning problems originating from the container-trucking industry are easier to tackle in respect to general ones. We show such different behavior through computational experiments: in particular, we have applied both a heuristic algorithm and some exact solution approaches to real-world instances as well as to benchmark instances from Beasley OR-library. Moreover, in order to gain an insight into the structure of the real-world instances, we have performed and evaluated various instance perturbations.

Keywords: set partitioning, Lagrangian relaxation, real-world instances, container-trucking industry, OR-library, instance perturbations.

1. Introduction

In this work we consider large-scale set partitioning problems. Set partitioning is a fundamental model in combinatorial optimization: the main applications include truck delivery management, vehicle scheduling, bus driver scheduling, and airline crew scheduling. It is also well known that set partitioning is an NP-hard problem [1].

Many real life problems, such as vehicle routing and airline crew scheduling can be formulated as SPPs. SPP has been studied extensively (see Balas and Padberg [2] for a survey of some of its applications and solution methods). Although the best-known application of SPP is surely airline crew scheduling [3], several other applications exist, including truck delivery management [4] and vehicle scheduling [5]. Moreover, some of the largest SPP instances are generated by vehicle scheduling problems (see, *e.g.*, [5]).

We show that real-world set partitioning problems originating from the container-trucking industry are easier to tackle in respect to general ones. Such result is achieved through a computational test: both a heuristic algorithm and some exact solution approaches have been applied to 50 real-world instances as well as to 55 benchmark instances of Beasley OR-library [6]. We also believe that most of our conclusions can be extended to other transportation problems not related with container movements (see Section 5).

The computational results that we have obtained show a quite different behavior of the heuristic: real-world instances seem to be better solved, in general (*i.e.*, the obtained bounds are more accurate, on average, and the computation time is more stable). We have also used CPLEX (the most widely used mathematical programming optimizer) in order to obtain and compare to the optimal values, when possible.

In order to gain an insight into the structure of the real-world instances, we have performed and tested five different instance perturbations. To be more specific, we have slightly altered the structure of the tested instances and have analyzed the new behavior of the algorithms. In particular, the following perturbations have been analyzed (see Section 4): deletion of high density rows, column merge, random insertion of new unitary elements, displacement of unitary elements within the same columns, and density variation of OR-library instances.

2. Set partitioning problem (SPP)

2.1. Definition

First, we give a formal description of SPP. Let M be a non-empty and finite set and let F be the set of feasible subsets of M. Associated with each element j of F is cost c_j. Let A be *0-1* matrix with a row for each element in M and a column for every characteristic vector of a feasible subset in F. The problem is to find a collection of elements of F, which is a partition of M, where the cost sum of theses elements is minimal.

A mathematical formulation of SPP follows:

$$\min\left(c^T x\right)$$
$$Ax = 1$$
$$x \in \{0,1\}^n,$$

where x is the binary decision variable vector, A is *0-1* matrix and c is the column cost vector.

2.2. The adopted solution approach

In order to tackle large-scale SPPs we have implemented a heuristic algorithm (in this work, matrix A of the largest tested instances has hundreds of thousands columns). In particular, we have preferred a Lagrangian relaxation based heuristic that exploits the subgradient method, a well-known approach used to find good Lagrangian multipliers. To be more specific, the algorithm operates as follows. First, all the constraints of SPP are relaxed in a Lagrangian fashion. The determination of near-optimal Lagrangian multipliers corresponds in some sense to a heuristic solution of the dual of LP relaxation. A widely used approach for finding near-optimal multiplier vectors within short computing time uses the subgradient method. This approach generates a sequence of nonnegative Lagrangian multipliers, starting from an arbitrarily defined vector (all the multipliers have been initially set to 0, in our case). We have implemented the classical Held and Karp choice for updating the Lagrangian multipliers (see [7]).

We have implemented such a heuristic solution approach since it has proven to be quite effective in tackling large-scale covering and partitioning problems [8]. Although we have implemented only one particular heuristic algorithm, it is important to note that different solution approaches might be differently affected by the structure of the tackled instances.

3. Real-world and OR-library SPP instances

In this work we have tested real-world instances originating from the container-trucking industry. Such instances have been supplied by an Italian transportation company that operates in Italy and in the neighboring countries as well. Its fleet consists of hundreds of tractors and semi-trailers capable of handling all types of containers. Its main customers are shipping lines, maritime agencies, service companies, industrial firms as well as private individuals.

The considered real-world SPP instances model a container transportation problem. The rows of matrix A represent the transportation orders to be executed in a given time horizon (namely, a day) and the columns represent the *pairings*, *i.e.*, the sequences of orders that can be carried out by one vehicle.

The behavior of the real-world instances in respect to different solution approaches, both exact and heuristic, has been compared with OR-library SPP instances [6]. See Table 1 for some aggregated results that have motivated this work. In Table 1 the gap between the upper bounds obtained with the heuristic and the optimal values obtained with CPLEX is shown.

Table 1. Different behavior of real-world and OR-library SPP instances

	Real-world instances	OR-library instances
Average	0.42 %	17.13 %
Std. dev.	0.48 %	72.56 %

4. Instance structural analysis: computational experiments

Since we believed that some particular features of the instances do affect significantly the effectiveness and efficiency of a solution algorithm, we have designed and performed five specific tests that should either confirm or deny the soundness of our hypotheses. In four cases we have perturbed the real-world instances, whereas in one case we have performed a perturbation on OR-library instances. The description of such perturbations is presented in the following subsections together with some results.

4.1. Deletion of high density rows

In this perturbation we have discarded 5 % of rows of matrix A in the real-world instances. In particular, the eliminated rows are the ones with the highest number of unitary elements. Thus, in this test, from a practical point of view, the most *easily manageable* transportation orders are not taken into account (*i.e.*, we have discarded those orders that appear in many pairings). Our computational study shows that this instance structure alteration may not affect significantly the quality of the obtained solutions. Some results are reported in Table 2, where the gap between the upper bounds obtained with the heuristic and the optimal values obtained with CPLEX is shown. Table 3 reports the gaps for three particular instances (E1, E2, and E3), as an example. In the following, A, B, and C represent three different parameter settings of the adopted heuristic algorithm.

Table 2. Average results of the first test

	Before perturbation	After perturbation
Average	0.42 %	0.33 %
Std. dev.	0.48 %	0.42 %

Table 3. Results obtained for three particular instances; first test

		Original Instances	Modified Instances
A	E1	0.05	0.05
	E2	0.65	0.19
	E3	2.83	0.84
B	E1	0.08	0.13
	E2	0.79	0.21
	E3	1.50	1.44
C	E1	0.09	0.05
	E2	0.47	0.12
	E3	4.90	1.62

As it can be seen in Table 3, in one of the reported cases the gap gets worse (instance E1 with parameter setting B).

4.2. Column merge

In this perturbation we have merged together the columns of matrix A, in the real-world instances. In particular, the number of columns has been halved. Roughly, the applied merging rule works as the logical operator OR and the cost of the new column is given by the sum of the costs of the two columns that have been merged. From a practical point of view, this is similar to forcing half of the resources (*i.e.*, vehicles) to carry out all the transportation orders. Our computational study shows that the obtained instances are hardly solvable as effectively as before. See Table 4 for some results where the gap between the upper bounds obtained with the heuristic and the optimal values obtained with CPLEX is shown. Table 5 reports the gaps for three particular example instances E1, E2, and E3.

As it can be seen in Table 5, there are many exceptions to the general worsening of the gaps. As a matter of fact, in spite of the perturbation, instance E1 is solved to prove the optimality for each of the parameter settings.

4.3. Random insertion of unitary elements

In this perturbation a random insertion of unitary elements is performed on matrix A in the real-world instances. In particular, the density of matrix A is increased so that the perturbed instances become more similar to OR-library ones.

Table 4. Average results of the second test

	Before perturbation	After perturbation
Average	0.42 %	1.36 %
Std. dev.	0.48 %	1.05 %

Table 5. Results obtained for the three particular example instances; the second test

		Original Instances	Modified Instances
A	E1	0.05	0.00
	E2	0.65	0.32
	E3	2.83	1.99
B	E1	0.08	0.00
	E2	0.79	0.49
	E3	1.50	1.72
C	E1	0.09	0.00
	E2	0.47	0.68
	E3	4.90	1.79

In spite of the low increase of the density (less than 1%), the perturbed instances become *almost intractable* (CPLEX could not obtain a proven optimal solution within 30 minutes). This happens to be a very enlightening event. Table 6 reports some results where the gap between the upper and the lower bounds obtained with the heuristic is shown, since the optimal values were not obtainable in reasonable computation time, after perturbation. Table 7 reports the gaps for the three particular example instances E1, E2, and E3.

As it can be seen in Table 7 in a few cases it may happen that the heuristic algorithm finds upper bounds that are better than those obtained with CPLEX (instance E1 for parameter settings B and C).

4.4. Displacements of unitary elements within the same columns

In this perturbation for each column we have randomly displaced the unitary elements of matrix A in the real-world instances. By doing so, obviously, the density of A remains the same. Nevertheless, from a practical point of view, this perturbation heavily changes what is supposed to be the geographical distribution of the transportation orders over

Table 6. Average results of the third test

	Before perturbation	After perturbation
Average	0.42 %	181.38 %
Std. dev.	0.48 %	169.33 %

Table 7. Results obtained for the three particular example instances; the third test

		Original Instances	Modified Instances
A	E1	0.05	3.27
	E2	0.65	287.91
	E3	2.83	515.55
B	E1	0.08	-0.89
	E2	0.79	268.58
	E3	1.50	459.47
C	E1	0.09	-0.04
	E2	0.47	283.47
	E3	4.90	458.72

the served territory. In fact, we believe that the spatial distribution of the orders is one of the most important features defining the beneficial structure of a SPP instance (see Section 5). Table 8 reports some results where the gap between the upper and the lower bounds obtained with the heuristic is shown, since the optimal values were not obtainable in reasonable computation time, after perturbation. Table 9 reports the gaps for the three particular example instances E1, E2, and E3.

4.5. Density variation of OR-library instances

In this test we have perturbed OR-library instances. In particular, the density of matrix A has been lowered by bounding, for each column, the number of unitary elements. The chosen bound is the maximum number of unitary elements that are present in the columns of the real-world instances. Despite the decrease of the density, the perturbed instances are, in general, more difficult to tackle: the gaps between the upper bounds obtained with the heuristic and the optimal values slightly worsen, whilst the computation time needed to obtain an optimal solution increases unacceptably (up to about 200 times). Some results are reported in Table 10, where the gap between the upper and the lower bounds obtained with the heuristic is shown, since the optimal values were not obtainable in reasonable computation time, after perturbation. Table 11 reports the gaps for three particular OR-library instances (O1, O2, and O3), as an example.

Table 8. Average results of the fourth test

	Before perturbation	After perturbation
Average	0.42 %	19.50 %
Std. dev.	0.48 %	30.52 %

Table 9. Results obtained for the three particular example instances; the fourth test

		Original Instances	Modified Instances
A	E1	0.05	11.09
	E2	0.65	15.93
	E3	2.83	43.07
B	E1	0.08	5.31
	E2	0.79	14.47
	E3	1.50	46.59
C	E1	0.09	4.81
	E2	0.47	16.53
	E3	4.90	39.22

Table 10. Average results of the fifth test

	Before perturbation	After perturbation
Average	17.13 %	21.62 %
Std. dev.	72.56 %	73.38 %

Table 11. Results obtained for three particular OR-library instances; the fifth test

		Original Instances	Modified Instances
A	O1	0.00	72.23
	O2	7.18	0.00
	O3	8.67	6.38
B	O1	0.00	37.94
	O2	13.47	9.94
	O3	23.25	17.84
C	O1	0.00	27.30
	O2	13.47	124.69
	O3	8.37	15.10

5. Conclusions

In this work we intend to show that some real-world SPPs are easier to tackle in respect to general ones. We have decided to prove such a fact through some computational experiments. Since we believed that some particular features of the instances do affect significantly the effectiveness and efficiency of a solution algorithm, we have designed five specific tests that should either confirm or deny the soundness of our hypotheses. Such a study has been performed since, to the best of our knowledge, there is a lack of real-world SPP instance structural analysis in the literature; nevertheless, the results obtained may be useful in order to better tackle some practical applications, as described in the following.

The results obtained in this work suggest that, *e.g.*, low density does not always guarantee good performances (the fifth test), in general. In other words, a given decrease of the density of unitary elements is not sufficient to obtain a better behavior of a solution algorithm. Of course, this result holds as long as the density is not lowered too much, since very low density SPPs are always quite easy to be solved.

Another non-trivial finding is that reducing the size of an instance by merging columns together may worsen the behavior of a solution algorithm (second test). A very dreadful behavior of the heuristic is obtained when adding or displacing unitary elements in a random fashion: this seems to heavily modify the beneficial structure (the third and the fourth tests). From a practical point of view, the third and the fourth tests, in particular, suggest that the distribution of the transportation orders over the served territory is a crucial aspect. As it happens, pairings are usually made of orders that are physically close to one another. Moreover, note that this characteristic is not limited to container-transportation industry: as a matter of fact, many real-world vehicle delivery problems present the same peculiarity. Roughly, this feature may lead to SPPs that are in some sense constituted by several subproblems of smaller size.

The development that is currently being investigated is concerned with the determination of a *measure of tractabil-*

ity of a given general SPP instance, according to the results obtained in this work. Moreover, note that such results may then be used to develop solution algorithms specifically designed for tackling certain classes of SPPs.

References

1. Garey, M. R. and Johnson, D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: W. H. Freeman and Company, 1979.
2. Balas E. and Padberg M. Set partitioning: a survey. *SIAM Review*, Vol 18, 1976, p. 710–760.
3. Marsten, R. E. and Shepardson, F. Exact solution of crew problems using the set partitioning mode: recent successful applications. *Networks*, Vol 11, 1981, p. 165–177.
4. Kelly, J. P. and Xu, J. F. A set-partitioning-based heuristic for the vehicle routing problem. *Informs Journal of Computing*, Vol 11(2), 1999, p. 161–172.
5. Borndörfer, R. Aspects of Set Packing, Partitioning and Covering. PhD thesis, Technische Universität Berlin, Berichter aus der Mathematik, Shaker, Aachen, 1998.
6. Beasley, J. E. OR-library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, Vol 41(11), 1990, p. 1069–1072.
7. Held, M. and Karp, R. M. The traveling salesman problem and minimum spanning trees: part II. *Mathematical Programming,* Vol 1, 1971, p. 6–25.
8. Caprara, A.; Toth, P. and Fischetti, M. Algorithms for the Set Covering Problem. *Annals of Operations Research*, Vol 98, 2000, p. 353–371.

DIDELĖS MATEMATINĖS AIBĖS DALIJIMO PROBLEMŲ SPRENDIMAS, NAGRINĖJANT REALIUS PAVYZDŽIUS

L. Coslovich, R. Pesenti, W. Ukovich

Santrauka

Nagrinėjamos didelių matematinių aibių dalijimo problemos. Autorių tikslas – atskleisti praktines matematinių aibių dalijimo problemas, kurių pasitaiko konteinerių vežimo versle. Aprašomi du eksperimentai, atlikti kompiuteriu. Atliekant pirmąjį eksperimentą, realiai aibės perdalijimo problemai spręsti buvo pritaikyti euristinis ir keli tikslūs sprendimo metodai. Atliekant antrąjį, tie patys metodai pritaikyti naudojant duomenis iš Beasley operacijų tyrimų bibliotekos. Gauti rezultatai palyginti, ir gauta naudingos informacijos apie realią matematinių aibių struktūrą.

Pagrindiniai žodžiai: matematinės aibės dalijimas, konteinerių vežimo verslas, operacijų tyrimų biblioteka.

Luca COSLOVICH. Doctor of Science, Post-doc in Operations Research. DEEI, University of Trieste. In 2005, Doctor in Information Engineering University of Trieste (Italy). Degree in Electronic Engineering in 2001. Research interests: metaheuristic algorithms for transportation problems, Web service applications in the logistic chain and decision support systems for free electricity market operators.

Author of two papers which are going to be published in European Journal of Operational Research.

Raffaele PESENTI. Doctor of Science, Professor. Department of Computer Science Engineering. University of Palermo. Doctor in systems sciences at the University of Genova (Italy). Research interests: management and evaluation of complex systems in transportation/logistics, strategic and analytical elements of the systems design and the development of models and methods which may find application in the day by day operations of transportation firms.

Walter UKOVICH. Professor of Operations Research, Faculty of Engineering, the University of Trieste.

National coordinator responsible for Coordinate and Strategic Projects of CNR (Italian National Research Council) on Logistics, Member of the Evaluation Boards of the University of Trieste, of the University of Macerata (Italy), of the University Architecture Institute of Venezia (Italy), and of the Polytechnic of Turin (Italy). Research interests: study of partially unknown systems, vector, multicriteria and multiobjective optimization, management of distribution networks and systems, traffic control, public transportation systems, periodic phenomena, image processing, evaluation of diagnostic and therapeutic techniques, organization and management of health systems, production planning and control, logistics, innovation and evaluation problems.