THE MINIMUM WEIGHT SPANNING STAR FOREST MODEL OF THE OPTIMAL DIVERSITY MANAGEMENT PROBLEM

AGOSTINHO AGRA$^{(A)}$, DOMINGOS M. CARDOSO$^{(A)}$, J. ORESTES CERDEIRA$^{(B)}$,
MIGUEL MIRANDA$^{(C)}$, EUGÉNIO ROCHA$^{(A)}$
$^{(A)}$UNIVERSITY OF AVEIRO; $^{(B)}$ TECHNICAL UNIVERSITY OF LISBON (TULISBON);
$^{(C)}$ YAZAKI

Abstract. In this paper we describe a real application project concerned with the implementation of optimization algorithms to solve a car industry problem known, in the literature, as the Optimal Diversity Management Problem. The minimum weight spanning star forest model for this problem is introduced, its complexity is analyzed, the implemented optimization techniques are presented and the computational results for real data instances are reported. Since the practical problem particularities had strongly conditioned the implemented optimization algorithms, these practical specifications are explored in the paper. Based on the results of this project, the manufacturer currently solves instances that are more than twenty times larger than the largest instances reported for this problem.

keywords: p-median; star covering

1. Introduction

This paper reports a practical implementation of optimization techniques to solve a particular problem of car industry. This problem is known as the *Optimal Diversity Management Problem* (ODMP). Although the project was developed without the knowledge of previous work, both the problem formulation and the optimization techniques implemented are much similar to those already presented in previous papers about the subject. The problem was first introduced in [3] and [4]. A more recent paper on the subject with similarities with our work is [1] (see also [2]).

Despite the previous contributions for this problem we believe this paper still has two major contributions:

i) We present a different proof of NP-hardness for the optimal diversity management problem (ODMP) that, in particular, proves the stronger result that the ODMP with constant option costs and constant demands is NP-hard. This also motivates the introduction of a new model for the ODMP: the *minimum weight spanning star forest* problem.

Date: February 16, 2007.
The research of the first, second and third authors was supported by *Centre for Research on Optimization and Control (CEOC)* from the "Fundação para a Ciência e a Tecnologia" FCT, cofinanced by the European Community Fund FEDER/POCI 2010.

ii) This paper focus on a real application and intends to a) tackle very large real instances (we run instances whose dimensions are greater than twenty times de greatest dimensions previously reported); b) provide real problem specifications which may motivate researchers to study different approaches to this problem; c) report preliminar results on real data which also may influence research for new optimization techniques.

In Section 2 we describe the practical problem and we also point out particular specifications of the problem. In section 3 we revisit the standard mathematical model for the ODMP, introduce the minimum weight spanning star forest model for this problem and present a new proof of its NP-hardness. In Section 4 we describe the implemented optimization techniques. The practical computational results are presented in Section 5. Finally, in Section 6, we make some final remarks and summarize the main questions raised in the paper.

2. Problem description

Cars are purchased with a set of active *options* (airbags, air conditioned, radio car, etc). Active option means that the car is prepared with the connections necessary to include such option (that is, the car has all the necessary material for that connection). A set of active options is called a *configuration.* For technical reasons, it is not possible to produce a large variety of different configurations. Therefore, in general, cars are produced with more active options than the ones asked by the clients. Since the global cost of option connections is very high in automobile industry, it is essential to choose a set of, say p, different configurations to be produced in order to minimize the total cost of the unnecessary option connections.

Problem particularities. Next we describe the particularities of this real application problem which have great influence on the choice of the optimization techniques and may motivate the study of alternative models.

(1) Some pairs of different configurations associated with certain special options are incompatible, such as the configurations related with left-hand side and right-hand side wheel. This incompatibility relation produce classes of compatible configurations. For large instances the number of classes of compatible configurations can be greater than twenty.

(2) Each class of compatible configurations consider a set of options and in each of these classes, the configuration with all options (considered in that class) active should be produced.

(3) There are two type of instances: instances with real data, where the demand for each configuration is known in advance, and forecasted instances were the demand is estimated. This last case occurs, for instance, when a new car model is to be produced. In the first case the number of different configurations is small even when the total demand is very large. The reason is that the clients usually choose the same type of options. In the second case however the number of configurations can be very large (more than two million). This follows from the fact that the company doesn't

consider the possibility of dropping any feasible configuration, and therefore every configuration has a strictly positive (although sometimes very small) probability of being produced.

(4) The maximum number of configurations to be produced (value of p) is small (usually, at most sixty).

(5) It is desirable to provide estimative costs for several values of p.

(6) Car manufacture companies (clients of the cable manufactures) usually pay a percentage of the additional cost. The measure of this cost is called the give-away percentage. This is computed as $\%\ give{-}away = \frac{additional\ cost}{ideal\ cost} \times 100$, where the *ideal cost* is the cost of the demanded configurations and the *additional cost* is the total cost of solution minus the ideal cost.

3. Mathematical model and complexity

Consider the *inclusion relation configurations digraph* $G = (V, A)$ where the vertices are configurations (that is, sets of active option connections) and an arc $(i, j) \in A$ means that the configuration i includes configuration j (each active option in configuration j is also active in configuration i). Note that this digraph has no direct cycles (that is, it is acyclic) and it is arc transitive (that is, if $(i, j), (j, k) \in A$, then $(i, k) \in A$). A *spanning star forest* (SSF) of G is spanning subdigraph where each component is a star. Here, we consider that a star has a center, which is a vertex with zero in-degree, and all other vertices, if any, have zero out-degree and one in-degree. Given any acyclic arc transitive digraph (in particular, an inclusion relation configurations digraph), to determine a SSF of minimum cardinality is very easy. In fact, since every acyclic arc transitive digraph $G = (V, A)$ is the comparability digraph of the partially ordered relation set (poset), $P = (V, \succeq)$, such that $x \succeq y$ iff $x = y$ or $(x, y) \in A$, the following proposition holds.

Proposition 3.1. *Let $G = (V, A)$ be an acyclic arc transitive digraph. Then there is a SSF of G where each star center is a maximal element for the poset defined by G. Furthermore, there is no other SSF of G with less number of stars.*

Proof. Since it is immediate to conclude that from the maximal elements of the poset defined by $G = (V, A)$ we may construct a SSF with these maximal elements as star centers, we just prove the second part of the theorem. Let $M = \{v_{j_1}, \ldots, v_{j_m}\} \subseteq V$ be the set of maximal elements of the poset defined by G and let $C = \{v_{i_1}, \ldots v_{i_p}\} \subseteq V$ be the set of centers of a SSF of G. Since there is no arc $(x, y) \in A$ with $x \in C$ and $y \in M$, it follows that $M \subseteq C$ and then $|M| \leq |C|$. □

A SSF with minimum number of stars is designated *minimum size* SSF. Let c_v be the unit production cost of configuration v (i.e, the sum of the costs of all active options in configuration v) and let n_v denote the expected number of cars with configuration v that will be sell. To each each arc (i, j) of G we assign the weight $w_{ij} = n_j(c_i - c_j)$. Note that $w_{ij} > 0$ since every active option has a positive cost, configuration i strictly contains configuration j, and it is assumed that $n_i > 0$, otherwise configuration i would not be considered. The weight of a

SSF is the sum of the weights of its arcs. The p-ODMP is to choose a SSF with p centers (configurations) of minimum weight. This is usually seen as a version of the $p-$median problem and, unlike minimum size SSF problem, it is NP-hard. The first proof of NP-hardness of the ODMP was given in [3]. Here we present a different proof for a model of p-ODMP where the weight of each arc of the inclusion relation configurations digraph is the difference between the cardinality of the configurations represented by the vertices connected by the arc. This particular model is called the minimum weight *spanning star forest* with p stars (minimum weight p-SSF) of an acyclic arc transitive digraph, and consists on the determination of a minimum weight arc sum spanning star forest with p stars. Let us consider this problem in the following a different perspective (covering sets by sets):

Let $N = \{1, 2, \ldots, n\}$ (corresponding to the set of options) and $\mathcal{F} \subseteq 2^N$ a collection of subsets of N (corresponding to the set of configurations, that is, the set of vertices of G). A *spanning star forest* of $\mathcal{F}$ is a partition $\{F_1, F_2, \ldots, F_k\}$ ($F_j \neq \emptyset$) of $\mathcal{F}$ such that in each F_j there exists an element X_j containing all other elements $X \in F_j$ (configuration with all active options included in the configurations of F_J). We call X_j the *center* of the *star* F_j. Each arc in the star F_j is defined by the ordered pair of sets (X_j, X) and has weight $|X_j| - |X|$, for all $X \in F_j \setminus X_j$.

Note that every maximal (with respect to inclusion) element of $\mathcal{F}$ must be the center of some star.

Theorem 3.2. *Deciding whether a collection $\mathcal{F}$ of subsets of $N = \{1, 2, \ldots, n\}$ has a p-SSF of weight not greater than L is NP-complete.*

Proof. The problem is clearly in NP. We transform the 3-minimum cover problem which is NP-complete (see [5]) to the minimum weight p-SSF decision problem. Note that the 3-minimum cover asks whether a collection $\mathcal{C}$ of subsets with cardinality 3 of $N = \{1, 2, \ldots, n\}$ contains a cover of size p or less, i.e., a subset $\mathcal{C}'$ of $\mathcal{C}$ with $|\mathcal{C}'| \leq p$ such that every element of N occurs in at least one member of $\mathcal{C}'$. Then, given an instance $\mathcal{C} \subseteq 2^N$ (with $|N| = n > 5$) of the 3-minimum cover of N of size at most p, let us define $\mathcal{F} := \{N, \{1\}, \ldots, \{n\}\} \cup \mathcal{C}$ and $L := 2n + (|\mathcal{C}| - p)(n - 3)$. We show that $\mathcal{C}$ contains a cover of N of size at most p iff $\mathcal{F}$ has a $(p + 1)$-SSF of weight arc sum not greater than L.

If $\mathcal{C}'$ is a cover of size p of $\mathcal{C}$, then it is possible to assign to each singleton $\{j\}$, $j = 1, \ldots, n$, of $\mathcal{F}$ a unique member of $\mathcal{C}'$. This defines a set of k pairwise disjoint stars of $\mathcal{F}$ with total weight equal to $2n$. The remaining star of $\mathcal{F}$ has center N and includes all the $|\mathcal{C}| - p$ sets of $\mathcal{C} \setminus \mathcal{C}'$. The weight of this star is $(|\mathcal{C}| - p)(n - 3)$, and thus the SSF consisting of these $p + 1$ stars has weight equal to L (see Figure 1). Suppose $\mathcal{C}$ has no cover of size p. Any choice of p elements of $\mathcal{C}$ to became the centers of stars will leave at least one singleton $\{j\}$ of $\mathcal{F}$ out from those stars. The weight of any such $(p + 1)$-SSF is at least $3(n - 1) + (|C| - p)(n - 3) > L$.

We complete the proof noting that the weight of every $(p + 1)$-SSF that uses an arbitrary singleton $\{j\}$ to be the center of any star is no less than $2n - 2 + (|C| - p)(n - 3) + n - 3$, and therefore is greater than L. □

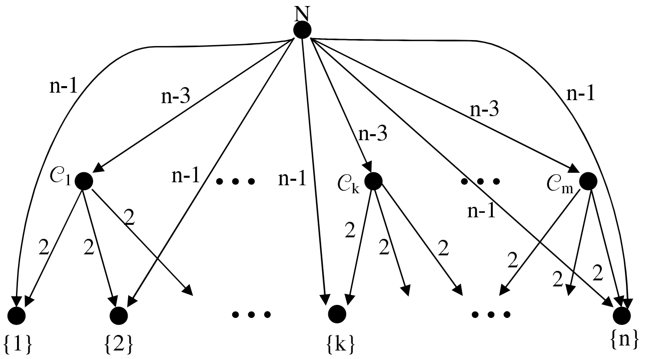

Figure 1. Comparability diagram

In real world applications, the inclusion relation configurations digraph has several connected components. Let us assume that the inclusion relation configurations digraph $G = (V, A)$ has t components, $G_1, \ldots, G_t$, and also that for each component G_k, the minimum weight spanning star forest with s stars has weight W_k^s, for $s = 1, \ldots, m$. Thus, the determination of the weight of the minimum weight spanning star forest with p stars for the the inclusion relation configurations digraph G, with at least one star in each connected component and such that $p = s_1 + s_2 + \cdots + s_t \geq t$, is equivalent to choose the t best entries, $W_1^{s_1^*}, \ldots, W_t^{s_t^*}$, of the table $W = [W_k^s]$, one entry in each column, such that

$$\sum_{k=1}^{t} W_k^{s_k^*} = \min \left\{ \sum_{k=1}^{t} W_k^{s_k} : \sum_{k=1}^{t} s_k = p \right\}.$$

For instance, considering the Table 1, obtained for an inclusion relation configurations digraph G with connected components $G_1, \ldots, G_4$, the best entries for the determination of the minimum weight arc sum spanning star forest of G with 8 stars are the framed ones.

s	G_1	G_2	G_3	G_4	
1	12	$\boxed{12}$	13	12	
2	9	11	$\boxed{10}$	$\boxed{8}$	
3	$\boxed{7}$	8	9	6	
4	6	7	6	4	
5	4	5	3	2	
$s_k^* \rightarrow$	3	1	2	2	$\sum_{j=1}^{4} W_k^{s_k^*} = 7 + 12 + 10 + 8 = 37$

Table 1. Weights of the minimum weight spanning star forests of the components of G.

In the next section, this problem will be formulated as binary linear programming problem which, in general, is NP hard too.

4. Optimization techniques

Based on the specifications of the problem stated in Section 2, the choice of the optimization techniques is strongly limited.

First, the use of standard commercial software to solve the complete problem must be excluded, since we are dealing with instances with millions of vertices.

Second, from items 1 and 2, it follows that the number of connected components is usually large. However, in each component, there is a vertex just with forward arcs connecting it with the remaining vertices of that component. Therefore, since these vertices (one for each component) are star centers in the final solution, a large number of vertices can be immediately chosen.

Third, since the number of components is large and p can not be very large, the average number of configurations (vertices) chosen in each component is, in general, not larger than three or four.

These two final observations together with the observations that the greedy choice of the second configuration in each component is optimal (for this type of graphs) and considering item 5 of particularities, strongly suggests the use of the greedy algorithm.

Next we present the optimization techniques implemented during the project.

Greedy Algorithm: Consider an instance with t components and consider an integer number p, $p > t$. Take the initial solution S with the t vertices, one for each component, with null in-degree. For $s = t + 1$ until p, choose a vertex not in S whose inclusion in S produces the greatest reduction of the total cost.

For the large dimension real world problems this algorithm has the drawback that it may imply reading the adjacency matrix (or part of it) of each component several times and, since the reading and writing operations are too expensive from the perspective of computational time consumption, an alternative is to run the greedy algorithm for each component from 1 to $p - t + 1$ configurations and then choose the best possible feasible combination of p configurations (vertices). This idea was independently used in [1]. Next we describe in more detail this second algorithm.

Two-Phase Algorithm: Consider an instance with t components and an integer number p, $p > t$.

Phase 1: Using the Greedy algorithm for each component G_k separately, $k = 1, \ldots, t$, obtain a feasible solution and the corresponding weight, W_k^s, of the minimum weight spanning star forest for the component G_k with s stars, for selecting s, from $s = 1$ to $s = p - t + 1$.

Phase 2: Given the table of weights W_k^s,, solve (heuristically) the problem

$$\text{(4.1)} \qquad \min \quad \sum_{k=1}^{t} \sum_{s=1}^{p-t+1} W_k^s z_{sk}$$

$$\text{(4.2)} \qquad \text{s. t.} \quad \sum_{s=1}^{p-t+1} z_{sk} = 1, \qquad (k = 1, \ldots, t),$$

$$\text{(4.3)} \qquad \sum_{k=1}^{t} \sum_{s=1}^{p-t+1} s z_{sk} = p,$$

$$\text{(4.4)} \qquad z_{sk} \in \{0, 1\}, \qquad (i = 1, \ldots, p-t+1,\ k = 1, \ldots, t),$$

where $z_{sk} = 1$ if the s-th entry of the k-th column of the table W is chosen, with $s = 1, \ldots, p-t+1$ and $k = 1, \ldots, t$. Constraints (4.2) ensure that for each component just one number of stars is selected and (4.3) ensure that the total number of selected stars is exactly p.

The problem was solved by a Genetic Algorithm. Briefly, genetic algorithms (GAs) are search procedures that reproduce the mechanics of natural selection and crossover in biology genetics. First developed by John H. Holland in the 1960's, they allow computers to find solutions (or approximations to them) for difficult problems by using evolutionary techniques, based on function optimization and artificial intelligence. The base elements of GAs are *chromosomes* which are (usually integer vectors) representations of solutions. GAs can operate on chromosomes in different ways. In what follows, we describe what is known as the *steady state* GA procedure: (1) generate a population of chromosomes; (2) choose a set of best chromosomes by applying a *selection function*; (3) recombine (*crossover*) chromosomes to obtain new ones; (4) change (*mutation*) some of them; (5) replace a percentage of weakest chromosomes by the new ones; and (6) repeat the steps (2)-(5) until it gives a good solution, in a precise sense, or stop after a given number of iterations. For a comprehensive guide to this subject and an extended list of applications see [6].

However, for an effective application of GAs, and considering their intrinsic natura, a good tuning of parameters, choice of the selection function, crossover and mutation mechanisms is required. Therefore, we describe the choices we found to give the best results for this problem. The size of the initial population is 300 chromosomes, the probability of a crossover is 0.6, the probability of a mutation is 0.1, the percentage of substitutions is 0.25, and the iteration limit number is 2000. The crossover uses the *two point crossover* mechanism, i.e. two cuts are made in both father and son chromosomes, the mutation is a transposition of two genes, i.e. the so called *swap mutator*, and the selection function is the linear objective function given by the expression (4.1). Let us emphasize that we did not register any significant improvement by, for example, defining the size of the initial population as a function of p and t.

The current implementation of the GA made use of the free C++ genetic algorithm library (GAlib), developed by Matthew Wall from MIT.

5. Computacional results

Next we present the computational result, obtained in a PIV 3.2 Ghz computer with 4Gb of memory and 300Gb of hard disk, where #*vertices* denotes the number of vertices, t the number of connected components, p the number of stars, #*cars* the total demand of configurations (total number of cars, and *% g-a* the give-away percentage. The last columns report results obtained using the Greedy or the Two-Phase algorithms.

				Greedy		Two-phase	
# vertices	t	p	# cars	time (sec.)	% g-a	time (sec.)	% g-a
166	33	65	38.910	5	0,68	2	0,68
239	24	60	38.910	5,5	1,72	3	1,72
317	47	65	38.733	3	1,32	1	1,32
584	16	20	30.500	3	1	1	1

Table 2. Results for known demand instances.

				Greedy		Two-phase	
# vertices	t	p	# cars	time (hr.)	% g-a	time (hr.)	% g-a
2.354.688	16	150	1.844.000	234	14,28	80	14,28
1.118.208	14	150	1.800.000	57	13,07	21	13.07
829.440	6	150	1.800.000	96	10,62	38	10,62
393.216	16	60	1.800.000	1	5,17	0,5	5,17

Table 3. Results for forecasted demand instances.

6. Conclusions

The real application study has several drawbacks from a theoretical perspective. Probably, the major one, is the lack of any more precise information about the quality of solutions obtained since, for practical reasons, no lower bounding technique was implemented. Another drawback is the fact that the output information obtained is conditioned by the practical interests of the company. Related with the first drawback it is relevant to refer that other algorithms had been tested in earlier test versions. We briefly mention an algorithm based on the work [7]: for each vertex compute the minimum of the weights of the incident arcs (take $+\infty$ if the vertex has in-degree equal to zero). Then choose the p vertices corresponding to the p greatest values obtained. Although the performance of these other algorithms was, in general, better for randomly generated data (ignoring some particularities of the problem), for the preliminary real instances, the greedy algorithm had, in general, very good performances. An intuitive explanation for that behavior can be found in Section 4. Also, local search algorithms, tested in previous versions, were dropped once the largest instances were considered.

This study also raised several questions.

- How to deal with instances with more than two million vertices? With the technological advances of the automobile industrie this number will certainly raise (exponentially) in the next years.
- The minimum weight arc sum spanning star forest model was considered for the ODMP. Can we derive new results for the ODMP by studying the structure of this model?
- Different practical perspectives of the problem were presented and different problem versions can be stated. Namely, considering item 5 of particularities, we may state the following problem: Find the minimum number of configurations such that the $\%\ give - away$ is lower than a given value.
- The particular structure of the inclusion relation configurations digraph could be explored. Until now, up to the best of our knowledge, only two major issues were explored: the fact that the graph has several connected components [1] and using some properties of the optimal solutions it is possible to fix the value of several variables [4].

Currently we are studying the last issue in order to improve the implemented algorithms.

References

[1] P. Avella, M. Boccia, C. D. Martino, G. Oliviero, A. Sforza, "A decomposition approach for a very large scale optimal diversity management problem," *4OR* **3**(1), 23–37, 2005.

[2] P. Avella, A. Sassano, I. Vasil'ev, "Computational study of large-scae p-Median problems, " *Mathematical Programming* **109**, 89-114, 2007.

[3] O. Briant, "Étude théorique et numérique du problème de la gestion de la diversité," PhD thesis, Institut National Polytechnique de Grenoble, 2000.

[4] O. Briant, D. Naddef, "The optimal diversity management problem," *Operations Research* **52**(4), 515–526, 2004.

[5] M. Garey, D. Johnson, "Computers and intractability: a guide to the theory of NP-completeness," Freeman, San Francisco, 1979.

[6] M. Gen, R. Cheng, "Genetic algorithms and engineering optimization," New York - John Wiley & Sons, 2000.

[7] P. Jarvinen, J. Rajala, H. Sinervo, "A branch and bound algorithm for seeking the p-median," *Operations Research*, **20**, 173–178, 1972.