A Strengthened Model for the Web Search Optimization Problem

GRAÇA MARQUES GONÇALVES
Departamento de Matemática, FCT, UNL
CMA, FCT, UNL
P 2829-516 Monte da Caparica
Portugal
gmsg@fct.unl.pt

LÍDIA LAMPREIA LOURENÇO
Departamento de Matemática, FCT, UNL
CMA, FCT, UNL
P 2829-516 Monte da Caparica
Portugal
lll@fct.unl.pt

Abstract: In this article we investigate the Web Search Optimization Problem, a NP-hard combinatorial optimization problem arising from Software Design. This is a new problem in the combinatorial optimization area. We develop a natural mixed integer linear programming formulation for this problem. The natural model is strengthened by including in the model valid inequalities. Computational experiments show that, in most cases, the strengthened model gives an integer solution for the problem. The lower bounds obtained by the strengthened model relaxation of the considered formulation improve upon those obtained by the natural model relaxation.

Key–Words: Web Search Optimization, Natural Strengthened Formulation, Valid Inequalities.

1 Introduction

When searching for a product on the web, groups of other similar products are proposed in advertising windows, to the user, based on previous search. These products are grouped by a similarity measure into groups with fixed size. This problem of software design for web search, named Web Search Optimization Problem (WSOP), is a combinatorial problem that consists of finding K disjoint groups with fixed size from a set of N items. The objective is to maximize the overall similarity among the items selected to belong to the same group. This is a NP-hard [4] clustering type problem with fixed cardinality constraints.

To the best of our knowledge, the WOSP has never been addressed in the scope of combinatorial optimization. We can find some literature about web search results in the field of statistics [7] and the use of genetic meta-heuristics for effective web search using ranking function optimization is introduced by Fan et al. [3].

A survey of another combinatorial optimization problems with fixed cardinality constraints, are described in [2], namely the k-cardinality tree problem, the k-cardinality TSP and related routing problems, the k-cardinality sub-graph problem, graph partitioning problems with a cardinality constraint, location problems and packing problems with cardinality constraints. Also a survey of mathematical programming models for clustering problems can be found in [5]. In this work, types of clustering and criteria are presented, also algorithms for hierarchical, partitioning, sequential, and additive clustering are addressed. Solution methods like dynamic programming, graph theoretical algorithms, branch-and-bound, cutting planes, column generation and heuristics are applied.

In order to solve the WSOP, in the scope of Combinatorial Optimization, we developed a natural mixed-integer formulation to obtain groups of products similar to the product previously searched. Next, we strengthened the natural formulation with a small subset of valid inequalities, the star inequalities, resulting in a better approximation of the convex hull of the problem's feasible region. With the purpose of testing the models, we made some computational experiments with random generated instances. The results showed that, in most cases, the strengthened model gives a feasible or the optimal solution for the problem.

The paper is organized as follows: in Section 2, we propose a natural mixed-integer formulation for the Web Search Problem as well as the strengthened model. Next, in Sections 3 and 4 some computational experiments and results are given. Finally, in Section 5, we present some conclusions.

2 Natural Formulation

Consider the following notation for the mixed-integer linear formulation for the Web Search Optimization Problem:

N - number of items ($N \in \mathcal{N}$)
K - number of clusters ($K \in \mathcal{N}, K < N$)
i, j - items indexes ($i, j \in \{1, \ldots, N\}$)
k - cluster index ($k \in \{1, \ldots, K\}$)
M_k - number of items per cluster$k$ ($M_k \in \mathcal{N}, \sum_k M_k < N$)
s_{ij} - similarity between items i and j, element of a symmetric matrix with diagonal elements equal to zero ($0 \leq s_{ij} < 1$)

Consider now the following binary decision variables:

x_{ik} - binary variable which indicates whether item i is in cluster k (=1) or not (=0), ($i = 1, \ldots, N$; $k = 1, \ldots, K$)
y_{ij} - binary variable which indicates whether items i and j are in the same cluster k (=1) or not (=0) ($i = 1, \ldots, N-1; j = i+1, \ldots, N$).

According with the parameters and the variables defined above, the WSOP can be formulated as a natural mixed integer linear programming problem (MILP) as follows, denoted by F:

$$max \quad \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} y_{ij} \tag{1}$$

$$s.t. \quad y_{ij} \geq x_{ik} + x_{jk} - 1 \tag{2}$$
$$1 \leq i < j \leq N; k = 1, \ldots, K$$

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} y_{ij} = \sum_{k=1}^{K} \frac{M_k!}{2!(M_k - 2)!} \tag{3}$$

$$\sum_{k=1}^{K} x_{ik} \leq 1 \tag{4}$$
$$i = 1, \ldots, N$$

$$\sum_{i=1}^{N} x_{ik} = M_k \tag{5}$$
$$k = 1, \ldots, K$$

$$x_{ik} \in \{0, 1\} \tag{6}$$
$$i = 1, \ldots, N; k = 1, \ldots, K$$

$$0 \leq y_{ij} \leq 1, \tag{7}$$
$$1 \leq i < j \leq N$$

The objective function (1) gives the total similarity, which is the sum of the similarity values between pairs of items placed in the same cluster. Constraints (2) relate the variables y_{ij} and x_{ik}. Constraints (3) give the exact number of variables y_{ij} which are equal to 1. The set of constraints (4) assures that each item belongs to one cluster at most. Cardinality constraints (5) do not allow violation of the number of items in each cluster. Finally, constraints (6)-(7) are the variables' domain.

Note that, although the variables y_{ij} are continuous, between 0 and 1, they are equal to 0 or 1 in the optimal solution, due to the maximization objective and the constraints (2)-(3).

In order to strengthen the model F, the following star inequalities are included in the model, resulting in the problem F_{star}:

$$\sum_{i=1}^{j-1} y_{ij} + \sum_{i=j+1}^{N} y_{ji} = \sum_{k=1}^{K} (M_k - 1) x_{jk},$$
$$j = 1, \ldots, N$$

These constraints force, for each j, $(M_k - 1)$ variables y_{ij} to be equal to 1, when j is in cluster k.

The star inequalities are presented in the paper of Park, Lee and Park [6] as facets for the convex hull of the set of feasible solutions for the edge-weighted maximal clique problem.

3 Computational Experience - Test Instances

This section reports the computational experience performed with the strengthened natural formulation for the Web Search Optimization Problem. As no benchmark instances exist for the WSOP, a set of instances for this problem was generated.

The instances of the WOSP used in the computational experiments were generated with $N = 40$, based on the instances for the k-cluster Problem, reported in the CEDRIC's Library of instances (http://cedric.cnam.fr/ lamberta/Library/k-cluster.html) (Billionnet [1]). Each one is defined by a graph, by its density (d) and by the number M_1 of items in the cluster, which is equal to 10 ($\frac{1}{4}N$), 20 ($\frac{1}{2}N$) or 30 ($\frac{3}{4}N$).

Three different graph densities, with values 0.25, 0.50 and 0.75 were considered. For each M_1 fixed and each density d fixed, there are 5 different graphs, with all edge weights equal to 1. The total number of different graphs is 15, and the set of 15 graphs considered for different M_1 values is always the same. Note that, for each graph, 3 different instances exist, each one with a different number M_1 of items in the cluster. The final number of instances is 45, as can be seen in table 1.

Table 1: Instance Parameters.

K	M_1	Graph density (d)	N.Inst
1	10	$(0.25, 0.50, 0.75)$	15
	20	$(0.25, 0.50, 0.75)$	15
	30	$(0.25, 0.50, 0.75)$	15

The WSOP test instances were then obtained by considering $N = 40$ and they were based on the 15 graphs mentioned above. For each edge graph $[i, j]$, a positive weight s_{ij} was randomly generated strictly between 0 and 1, defining the similarity between items i and j. The remaining data of these instances were defined as follows.

Instances with $K = 1, 2$ and $\sum_{k=1}^{K} M_k = 10, 20, 30$ were generated. For $K = 1$ we generated 45 instances, 15 for each M_1 value. For $K = 2$, 90 instances were generated, 15 for each value of $\sum_{k=1}^{K} M_k$ and for each choice of M_k, one with balanced and another one with unbalanced values of M_k, according with table 2.

Note that, for $K = 2$, for d fixed and $\sum_k M_k$ also fixed, there are 2 different instances relative to the same graph. Then, for those d and $\sum_k M_k$ fixed, there are 10 instances because 5 different graphs exist for each density.

Table 2: Instance Parameters with $N = 40$.

K	$\sum_{k=1}^{K} M_k$	M_k		N.Inst
1	10	$M_1 = 10$		15
	20	$M_1 = 20$		15
	30	$M_1 = 30$		15
2	10	$M_1 = 5\ (\frac{1}{2})$	$M_2 = 5\ (\frac{1}{2})$	15
		$M_1 = 2\ (\frac{1}{5})$	$M_2 = 8\ (\frac{4}{5})$	15
	20	$M_1 = 10\ (\frac{1}{2})$	$M_2 = 10\ (\frac{1}{2})$	15
		$M_1 = 4\ (\frac{1}{5})$	$M_2 = 16\ (\frac{4}{5})$	15
	30	$M_1 = 15\ (\frac{1}{2})$	$M_2 = 15\ (\frac{1}{2})$	15
		$M_1 = 24\ (\frac{4}{5})$	$M_2 = 6\ (\frac{1}{5})$	15

4 Computational Results

The models F and F_{star} were solved by using the standard mathematical software CPLEX. The algorithm provided by the Ilog CPLEX 12.6, ran on a i7 computer with 3.60 GHz processor and 8 GB RAM. In all tests the following CPLEX parameters were considered: time limit=7200 seconds, clocktype=1, mip tol absmipgap=0.0, mip tol mipgap=0.0, mip tol integrality=0.0, feasopt tolerance=0, threads=8, while the other standard CPLEX parameters were used. The computational tests were made for the instances described in the previous section.

In Tables 3 and 4 presented below, the first two columns are the graph density and the values of $\sum_k M_k$, respectively. The Gap and the CPU time of F and F_{star} models are the average values for 5 instances in Table 3 and average values for 10 instances in Table 4. The Gap at the root node of the search tree is equal to $\frac{v(\bar{P})-v(P)}{v(P)} * 100\%$, where $v(P)$ is the optimum value of the problem and $v(\bar{P})$ is the linear relaxation optimum value of the same problem. When there is no optimal solution available, the best integer solution was considered for the gap computation. Columns 3 and 5, headed by Average gap, give the average gap in percentage. In columns 4 and 6, headed by Average CPU time, we find the average CPU time (in seconds) required to solve the formulations F and F_{star}, respectively.

Table 3: Results for models F and F_{star} for K=1.

d	$\sum_k M_k$	F	F	F_{star}	F_{star}
		Average	Average	Average	Average
		gap (%)	CPU time (s)	gap (%)	CPU time (s)
25	10	188	368	177	46
	20	373$^{(1)}$	7200	299	3296
	30	51	50	35	13
50	10	88	70	85	29
	20	167	5834	148	2288
	30	21	8	16	7
75	10	54	35	52	19
	20	101	2649	91	1307
	30	13	7	10	6

(1) - 5 instances not solved.

For $K = 1$, from Table 3, we note that the model F_{star} with valid inequalities substantially improves the linear relaxation relative to the model F. This strengthened model gives the optimal solution for all instances in the time limit of 7200 sec, as opposed to model F which did not solve 5 instances in the same time limit. For both models, the average gap for the set of instances corresponding to $d = 25$, $d = 50$, $d = 75$ and $\sum_k M_k = 30$, is lower than the gap obtained for $d = 25$, $d = 50$, $d = 75$ and $\sum_k M_k = 10, 20$. Then, the F and F_{star} models perform better for instances with higher elements in the group. As for the CPU time, it is verified that F_{star} model has smaller computational times than F model, although it has one more set of constraints.

For $K = 2$, from Table 4, the model F_{star} gave the optimal value for the instances $d = 25$, $\sum_k M_k = 10$, $d = 50$, $\sum_k M_k = 10$ and for the instances with $d = 75$ and $\sum_k M_k = 10$. For the other instances the CPLEX did not obtained the optimal solution in the time limit of 7200 seconds, or gave the message ”out of memory”.

The model F_{star} with the valid inequalities for $K = 2$ also improves the linear relaxation relative to the model F. We remember that for the unsolved instances, the gap values are obtained from the best given feasible solution. This is why there are two average gap values for F_{star} higher than the corresponding values for F.

Note that the number of messages ”out of mem-

Table 4: Results for models F and F_{star} for K=2.

		F	F	F_{star}	F_{star}
d	$\sum_k M_k$	Average	Average	Average	Average
		gap (%)	CPU time (s)	gap (%)	CPU time (s)
	10	104	288	101	39
25	20	$265^{(1)}$	7200	$212^{(6)}$	7200
	30	$192^{(2)}$	7200	$222^{(7)}$	7200
	10	53	796	48	29
50	20	$177^{(3)}$	6003	$123^{(8)}$	6926
	30	$101^{(1)}$	7200	$107^{(7)}$	7200
	10	31	38	30	18
75	20	$77^{(4)}$	4392	$70^{(9)}$	4412
	30	$70^{(5)}$	6333	$65^{(5)}$	6343

(1) - 3 out of memory and 7 instances not solved
(2) - 8 out of memory and 2 instances not solved
(3) - 5 out of memory and 5 instances not solved
(4) - 4 out of memory and 2 instances not solved
(5) - 8 instances not solved
(6) - 4 out of memory and 6 instances not solved
(7) - 10 instances not solved
(8) - 5 out of memory and 1 instance not solved
(9) - 3 out of memory and 2 instances not solved.

ory" substantially decreased from model F to model F_{star}. Also, the number of solved instances is bigger for F_{star} model.

Although some instances were not solved for $K = 2$, the model F_{star} performs well with high densities and a greater number of elements in the groups.

As for the CPU times, we observe that the F_{star} model has, in general, lower computational times. For some groups of instances, the average CPU times are bigger for F_{star} because more instances were solved or ran in the time limit, thus influencing the average values.

5 Conclusions

This paper presents a natural formulation for the web search optimization problem. This formulation is strengthened with valid inequalities, the star inequalities, in order to improve the linear relaxation bound.

Computational experiments show that the CPLEX solver with the strength model, solved all the instances of the problem, for $K = 1$. For $K = 2$, the CPLEX solved all instances with $\sum_k M_k = 10$ and for all instances with $\sum_k M_k = 30$, the model F_{star} gave a feasible or the optimal solution.

We conclude that the model F_{star} performs better with high densities and a greater number of elements in the groups.

Research should be continued to developed heuristics in order to obtain lower bounds for all size instances. The study of other formulations and valid inequalities should also be made to reinforce linear relaxations to get better upper bounds for the optimal value.

6 Acknowledgments

This work was partially supported by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the project UID/MAT/00297/2019 (Centro de Matemática e Aplicações).

References:

[1] Billionnet A. (2005). "Different formulations for solving the heaviest k-subgraph problem". *Information Systems and Operational Research* 43 (3): 171-186.

[2] Bruglieri, M., Ehrgott, M., Hamacher, H. and Maffioli, F. (2006). "An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints". *Discrete Applied Mathematics*, 154, pp. 1344-1357.

[3] Fan, W., Gordon, M., Pathak, P., Xi, W., Fox, E. (2004). "Ranking Function Optimization For Effective Web Search By Genetic Programming: An Empirical Study". Proceedings of the 37th Hawaii International Conference on System Sciences, pp. 1-8.

[4] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the teory of NP-completeness*, San Francisco: W. H. Freeman and Company, 1979.

[5] Hansen, P. and Jaumard, B. (1997). "Cluster analysis and mathematical programming". *Mathematical Programming*, 79, pp. 191-215.

[6] Park, K., Lee and Park, S. (1996). "An extended formulation approach to the edge-weighted maximal clique problem". *European Journal of Operational Research*, 95, pp. 671-682.

[7] Zeng, H., He, Q., Chen, Z., Ma, W., Ma, J. (2004). "Learning to Cluster Web Search Results", SIGIR '04 Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 210-217.