Ann. Data. Sci. (2014) 1(1):15–23
DOI 10.1007/s40745-014-0003-y

Exact Solutions to the Capacitated Clustering Problem: A Comparison of Two Models

Mark Lewis · Haibo Wang · Gary Kochenberger

Received: 5 December 2012 / Revised: 19 March 2013 / Accepted: 20 April 2013 /
Published online: 12 April 2014

Abstract In this paper, we investigate a natural nonlinear alternative to a standard linear model for CCP and compare the two models on a set of test problems. Our results show that moderate sized instances of CCP can in fact be solved optimally with modern exact methods in modest amounts of time and that the quadratic model generally outperformed its equivalent linear alternative in terms of quickly finding optimal or near optimal solutions.

Keywords Capacitated clustering · Branch-and-Cut · Exact methods · Metaheuristic

1 Introduction

The capacitated clustering problem (CCP), in its various forms, is a well-known NP-hard combinatorial optimization problem that has attracted a lot of attention in the literature. In general, the idea is to group similar items together into clusters provided that some aggregate distance metric is optimized while satisfying capacity constraints

M. Lewis
Department of Management, Craig School of Business, Missouri Western State University,
St. Joseph, MO 64507, USA
e-mail: mlewis14@missouriwestern.edu

H. Wang (✉)
IB&TS Division, Sanchez School of Business, Texas A&M International University,
Laredo, TX 78041, USA
e-mail: hwang@tamiu.edu

G. Kochenberger
Department of Decision Science, School of Business Administration, University of Colorado at Denver,
Denver, CO 80217, USA
e-mail: gary.kochenberger@cudenver.edu

on the clusters formed. Applications have been reported in such diverse areas as sales force territory design, the design of garbage collection zones, telecommunications, computational biology, design of mail delivery systems, and pattern recognition. These and other applications are reported in recent papers by Negeriros and Palhano [1], Deng and Bard [2], and Chou et al. [3].

Due to the computational burden posed by CCP, most of the solution methods reported in the literature for CCP are heuristics of one kind or another. The papers by Negeriros and Palhano [1], Deng and Bard [2], and Chou et al. [3], for instance, give an overview of solution methodologies that include simulated annealing, Lagrangian relaxation, neural networks, genetic algorithms, variable neighborhood search, tabu search, and GRASP. In each case the method advanced was specially crafted for CCP.

The adoption of heuristic methodologies for CCP is motivated by the well-known limitation posed by exact methods when it comes to solving NP-hard problems. But modern exact methods, as represented by the branch and cut algorithms embedded in CPLEX [4] and Gurobi [5], continue to evolve in their capabilities. Today these algorithms can solve small to medium sized instances of CCP with certain characteristics to optimality in reasonable amounts of computing time. Moreover, the size of CCP instance that can be efficiently solved may depend on the modeling choice made to represent a given problem. For our work here, we compare the linear 0/1 model presented by Deng and Bard [2] with its natural (alternative) quadratic formulation. Our motivation is derived from earlier work [6] where nonlinear models have outperformed equivalent linear models in head to head testing for certain classes of problems.

Rather than testing special purpose methods, our computations are carried out by the general purpose branch and cut methods of CPLEX and Gurobi without any specialization for the problem structure under study. Thus the focus here is on the relative performance of the linear and nonlinear models and the extent to which state of the art commercial optimizers are capable of solving these difficult problems to optimality. We also provide a comparison of the performance of these exact methods with the performance of the heuristic method used by Deng and Bard [2].

2 Capacitated Clustering with Limits on Cluster Weights

In this section we consider the capacitated clustering problem recently studied by Deng and Bard [2]. We first present their linear 0/1 model for this problem followed by an equivalent nonlinear model that we suggest as an alternative to their linear model.These models are compared in section three.

2.1 The Linear Model of Deng and Bard

Consider a graph G = (V, E) with n nodes and both positive edge and node weights. We'd like to partition the graph into P clusters such that the sum of the edge weights within clusters is maximized provided the sum of the node weights in any given cluster is bounded. A standard linear model for this problem, as used by Bard and Deng [2], is:

Lin CCP:

$$Max \sum_{k=1}^{p} \sum_{e \in E} c_e x_{ek} \tag{1}$$

st

$$\sum_{k=1}^{p} y_{ik} = 1, \quad \forall i \in V \tag{2}$$

$$x_{ek} \leq y_{ik}, \quad x_{ek} \leq y_{jk} \; \forall e = (i, j) \in E, \; k = 1, \ldots \tag{3}$$

$$x_{ek} \geq y_{ik} + y_{jk} - 1 \; \forall e = (i, j) \in E, \; k = 1, \ldots, p \tag{4}$$

$$C^{\min} \leq \sum_{i \in V} w_i y_{ik} \leq C^{\max}, \quad \forall k = 1, \ldots, p \tag{5}$$

$$x_{ek} \in \{0, 1\}, \; y_{ik} \in \{0, 1\} \; \forall i, \quad e = (i, j) \in E, \; k = 1, \ldots, p \tag{6}$$

where $x_{ek} = 1$ if edge e has both endpoints in cluster k, 0 otherwise; $y_{ik} = 1$ if node i is included in cluster k, 0 otherwise; c_e = weight of edge $e \in E$, ; w_i = weight of node $i \in V$; p = number of clusters to be created; $C^{\max}$, $C^{\min}$ = upper and lower bounds on aggregate cluster node weight.

Note that even for modest sized graphs, this model explodes in size with many variables and constraints. Note also that due to the nature of the objective function, as remarked by Deng and Bard[2], the constraints of (4) are redundant. Moreover the binary restriction on thex_{ek}can be relaxed and x_{ek}treated as continuous in the range [0,1].

2.2 Equivalent Nonlinear Model

It's easy to see that the above linear model is a straight-forward linearization of an equivalent quadratic model. By substituting the product y_{ik}, y_{jk} for x_{ek}, all the edge variables (x_{ek}) and the constraints of (3) and (4) can be eliminated yielding the equivalent quadratic model:

Quad CCP:

$$Max \sum_{e \in E} c_e \sum_{k=1}^{p} y_{ik} y_{jk} \tag{7}$$

st

$$\sum_{k=1}^{p} y_{ik} = 1, \; \forall i \in V \tag{8}$$

$$C^{\min} \leq \sum_{i \in V} w_i y_{ik} \leq C^{\max}, \; \forall k = 1, \ldots p \tag{9}$$

$$y_{ik} \in \{0, 1\} \tag{10}$$

For any graph G, Quad CCP is much smaller that Lin CCP yet they are equivalent representations of the same problem. Despite a growing literature on the potential attractiveness of quadratic binary combinatorial models, as summarized in the paper by Alidaee et al. [6], the dominate practice, when confronted with a nonlinear model, is first try to find an equivalent linear representation. There are of course good reasons for this. Notably, this preference for linearity is rooted in the high quality of the various solution methodologies available for linear models and the conventional wisdom they have collectively encouraged.

Nonetheless, methods for solving certain nonlinear binary models, particularly quadratic binary models, have improved substantially. Many of these advances are embedded in exact solvers available in commercial codes and they provide an alternative to the more standard propensity to convert a quadratic model to an equivalent linear model. That is, it may be for certain problem classes, solving the quadratic representation of the problem may be computationally preferred solving the equivalent linear representation. We explore this in the computational testing that follows.

3 Computational Testing

In this section we provide a comparison of the Lin CCP and Quad CCP models for the capacitated clustering test problems used by Deng and Bard (D&B) in their recent paper [2]. The problems provided by D&B are of size 30, 40 and 50 nodes all with 5 clusters with a minimum of 5 nodes in a cluster and a maximum of 8, 9 and 12 nodes respectively There are 10 instances for each problem size. The thirty problems were randomly generated based on the recurrent US Postal Service problem of allocating zones to be serviced by mail carriers. Sixteen of the thirty problems were proven to be optimal to within 0.001 by Deng and Bard. We prove the remaining 14 problems optimal either using Cplex directly or by jump-starting a linear model of the problem with the more quickly found quadratic solution.

D&B ran their problems using Ubuntu Linux on a Dell Poweredge 2950 workstation with 2 dual core hyperthreading 3.73 GHz Xeon processors and 8 GB memory. Their results were obtained by a GRASP-related heuristic with path re-linking that was specially crafted for CCP and was compared to Cplex version 11. The results we report here were obtained via a Dell Precision desktop computer running Windows 7 with a single 3.4 GHz Intel i7 hyperthreaded processor having 4 cores and 12 GB memory. Both of these computers and the software used are similar in computing power, although the Poweredge workstation with 2 Xeon processors should be faster than a single quad-core i7 processor due to differences in bus and memory sharing [7].

To carry out our comparison of the linear and quadratic models we used two state of the art commercial optimizers, CPLEX version 12 and Gurobi version 5.0. Both of these codes have advanced branch and cut implementations leveraging multi-threaded computer architectures and are designed to optimize both linear and quadratic models with 0/1 variables. Gurobi 5.0, released in May 2012, provides enhanced support for solving quadratically modeled problems. Thus, an additional contribution of our paper is to provide a head-to-head comparison of these two packages on these problem sets.

Table 1 Summary of results using default settings for CPLEX and Gurobi

Problem group	Solver used	Avg. (median) time to solution	Avg. optimality gap (%)
30 nodes	Gurobi LIN	15 (11)	
	Gurobi QUAD	14 (1)	
	CPLEX LIN	16 (16)	
	CPLEX QUAD	226 (33)	
40 nodes	Gurobi LIN	1,298 (830)	
	Gurobi QUAD	785 (15)	0.9
	CPLEX LIN	2017 (1,765)	0.3
	CPLEX QUAD	2,917 (3,283)	0.3
50 nodes	Gurobi LIN	3,380 (3,102)	1.1
	Gurobi QUAD	813 (343)	1.2
	CPLEX LIN	5,720 (6,550)	1.9
	CPLEX QUAD	6,134 (6,578)	0.7

We compare CPLEX and Gurobi on the 30 problems modeled linearly (LIN) and quadratically (QUAD). Each problem had a time limit of 2 h. For the following discussion, in order to summarize the distribution of times, we present times in the following format: mean (median). Thus, if the median is dramatically different than the mean, then there were outliers in the distribution of times, e.g. Gurobi QUAD problem 40_4 was 4× longer than the average time, but 210× the median time. Table 1 summarizes our results that indicate that default Gurobi was faster than default CPLEX and that Gurobi QUAD was noticeably faster than Gurobi LIN, while CPLEX QUAD was slightly slower than CPLEX LIN.

CPLEX was able to prove optimality for the 30-node LIN problems in less than 1 minute and generally did so shortly after finding the optimal solution. Optimal solutions were found for 8 of the 10 40-node LIN problems with the results proven to be optimal for 5 of the 8 solutions. None of the 50-node LIN problems were solved to optimality with in the 2-h time limit. Across all CPLEX LIN problems, the best solutions found, for non-optimal solutions, were on average 98.3 % of the optimal solutions with a range of 97.1–99.6 %.

For the 30-node Gurobi LIN problems, the results are very similar to the CPLEX results. Namely, solutions were found and proven to be optimal in an average of approximately 15 (11) s. For the 40-node problems, optimal solutions were found by Gurobi LIN in less than half the time of CPLEX and proven to be optimal for all 10 problems in contrast to the CPLEX results where solutions for only 5 of the 40-node problems were proven to be optimal. For the 50-node problems, optimal solutions were found (but not proven) for 3 of the 10 problems while CPLEX found none of the optimal solutions and took on average about twice as long. Gurobi times for the linear model were about the same for the smaller problems, but were about twice as fast for the larger problems

For the 30-node problems, CPLEX QUAD optimal solutions were found but not proven within the allotted time limit. For the 40-node CPLEX QUAD problems, opti-

mal solutions were found (but not proven) for 9 of the 10 problems. In contrast to the linear model results, optimal solutions were found for 6 of the 10 50-node problems compared to none for the linear model! Across all CPLEX QUAD problems, for non-optimal solutions, the best solutions found were on average 99.3 % of the optimal solution with a range of 98.3–99.8 %. The median for the 30-node CPLEX QUAD problems is much less than the mean, indicating that most of the times are much less than the mean.

Using Gurobi with the QUAD model, for the 30-node problems, optimal solutions were quickly found and proven for 4 of the 10 problem instances. Optimal solutions were quickly found (but not proven) for 8 of the 10 40-node problems and for 5 of the 10 50-node problems. Across all problems where the best solution found was not optimal, the solutions found were on average 99.4 % optimal with a range of 98.5–99.9 %. Gurobi's average and median QUAD time to optimality was significantly faster than CPLEXs QUAD time. For the 50-node problems, Gurobi QUAD median time to optimality was about 20x faster than CPLEX QUAD.

For the 30-node QUAD problems, CPLEX spent 266 (33) s to find the optimal solution, but was unable to prove optimality on any problem, whereas Gurobi found the optimal in 14 (1) s and proved optimality for four of the ten problems in 1790 (3) s, with three of the 30-node problems proven optimal in less than 5 s. This performance illustrates that quadratic models were competitive with their linear alternatives.

For the 40-node LIN problems, Gurobi's average (median) time to the optimal solution was 1298 (830), about the same as CPLEX, 2017 (1765). In the 2 h time limit, Gurobi's time to best solution using the QUAD model was 785 (15), while CPLEX's was 2917 (3283), although CPLEX found nine out of the ten optimal solutions and Gurobi found seven with the other three being within 0.3–1 % of optimal. Gurobi's median time of 15 s reflects that optimal solutions were found for 6 out of the 10 problems in under 15 s.

For the 50-node LIN problems, Gurobi's time to optimal solution was 3380 (3102), about half that of CPLEX, 5720 (6550). CPLEX did not find any of the optimal solutions, but was within 2 % of optimality, while Gurobi found three optimal solutions and was within 1.1 % for the others. Gurobi's time to best solution using the QUAD model was 813 (343), finding five optimal out of ten, the rest being within 1.2 % of optimal. CPLEX's QUAD time was 6134 (6578), finding six out of the ten optimal solutions, the others being within 1 % of optimal. Gurobi's average and median times for the 50-node problems indicate it was faster to optimal solutions and that the QUAD model is faster than the LIN to optimal or near optimal solutions.

In general, comparing Gurobi's performance between the linear and quadratic model, we see that the quadratic model had more difficulty proving optimality but was considerable faster to finding good, often optimal, solutions. This is most noticeable as the problem size increases. The results of our testing suggest that both CPLEX and Gurobi, and both the linear and the quadratic models, were reasonably successful in finding high quality solutions to the set of test problems considered here. In fact, for many of the smaller test instances, solutions were found and proven to be optimal in a very reasonable amount of time. A difference in the results overall, however, has to do with the quality of solutions generated and the time required to find high quality solutions. Generally, the quadratic model produced high quality solutions faster than

the linear model. This was particularly true for the Gurobi results. On the other hand, the quadratic model took longer to prove optimality or failed to prove optimality in the allotted time.

It's interesting to compare the results presented here with those given in the original paper by Deng and Bard [2]. Their results, which they obtained from a metaheuristic specially crafted for CCP, generated optimal solutions for the 30/40/50 node problems in 5/12/24 s. Since their approach was heuristic in nature, it is expected that it would be faster than the default general purpose, exact approaches we employed here. Nonetheless, for many of the problems our results compare nicely with theirs both in terms of solution quality and solution time.

As noted, for comparison purposes between CPLEX and Gurobi, our results are based on default settings, that is, no parameter tuning. However, it should also be noted that if the CPLEX parameter is set to frequently use solution polishing, which tends to trade-off finding good feasible solutions quickly as opposed to proving optimality, then the comparison to the D&B metaheuristic becomes even closer. For example, with CPLEX set for frequent solution polishing, average times to the optimal solutions for the QUAD problems are 5 (3) s for the 30-node problems, 10 (10) s for the 40-node and 835 (35) s for the 50-node. However, comparable dramatic improvements by changing a single search or emphasis parameter were not found for Gurobi.

4 Summary and Conclusions

In this paper we considered a recently published version of the capacitated clustering problem. Using a published set of test problems, we compared the linear model of the literature with an alternative quadratic model. This comparison was carried out using the branch-and-cut methods of CPLEX and Gurobi, two state of the art, general purpose, exact optimizers. Our results not only permit a comparison of the linear and quadratic models, but also allow a comparison of the two optimizers, CPLEX and Gurobi. Furthermore, our results enable a comparison to be made of the performance of the exact methods with that of the specially crafted heuristic method of Deng and Bard.

Overall, the results suggest that while both models and both optimizers performed well, there are performance differences based on the problem set considered here. The quadratic model generally found high quality solutions more quickly than the linear model but was slower to prove optimality. Gurobi, on both the linear and the quadratic models, showed a performance edge over CPLEX in terms of both finding and proving optimal solutions. Finally, the exact methods employed here, while not quite as efficient as the heuristic of Deng and Bard across the entire test bed, was nonetheless successful in finding high quality solutions in a reasonable amount of time, and in some cases in times comparable to the heuristic times of Deng and Bard.

These conclusions should be viewed with a full understanding that they are derived from only one set of test problems. Nonetheless, they suggest that the quadratic model is a viable alternative to the better known linear model and that the exact methods tested here are in fact viable alternatives to the heuristic method of Deng and Bard for capacitated clustering problems of the type and size considered.

References

1. Negreiros M, Palhano A (2005) The capacitated centered clustering problem. Comput Oper Res 33:1639–1663
2. Deng Y, Bard J (2011) A reactive GRASP with path re-linking for capacitated clustering. J Heuristics 17:119–152
3. Chou C, Chaovalitwongse WA, Berger-Wolf TY, DasGupta B, Ashley M (2012) Capacitated clustering problem in computational biology: combinatorial and statistical approach for sibling reconstruction. Comput Oper Res 39:609–619
4. ILOG Inc. (2011) ILOG IBM CPLEX 11 user's manual. ILOG Inc., Philadelphia
5. Gurobi Optimization Inc. (2011) Gurobi manual. Gurobi Optimization Inc., Houston
6. Alidaee B, Kochenberger G, Lewis K, Lewis M, Wang H (2009) Computationally attractive nonlinear models for combinatorial optimization. Int J Math Oper Res 1:9–19
7. IBM (2012) Comparison of CPLEX performance on dual core machines with true multiple processor machines. IBM Technical Note

Mark Lewis has a BS in Electrical Engineering from the University of Kansas and worked as an engineer in avionic development at Lockheed Martin for twelve years before getting a Ph.D in Operations Research from Southern Methodist University in Dallas, Texas, USA. He is currently a professor teaching information systems and operations management. His research interests include mathematical programming and heuristics.

Haibo Wang received a Ph. D. in Production Operations Management, 2004 from The University of Mississippi. He is currently Killam Distinguished Professor of decision sciences in the Division of International Business and Technology Studies, Sanchez School of Business at Texas A&M International University. He has received the scholar of the year award in 2011 and the global scholar of the year award in 2013. He is guest professor and visiting professor of a number of institutions in China and guest editors of several international journals. He is co-founder of technology consulting company and senior partner of international logistics company. He has publications in such outlets as European Journal of Operational Research, Computers and Operation Research, IEEE transactions on Control System Technology, IEEE transactions on Automation Science and Engineering, Journal of Operational Research Society, Computers and Industrial Engineering, Journal of Applied Mathematical Modeling, International Journal of Flexible Manufacturing Systems, International Journal of Production Research, Journal of Human and Ecological Risk Assessment, Journal of Heuristics, Communications in Statistics, International Journal of Information Technology and Decision Making, Journal of Combinatorial Optimization, etc.

Gary Kochenberger has a BS in electrical engineering and a PhD in Management Science from the University of Colorado. He is currently a professor of Business Analytics at the University of Colorado at Denver. His research interests include combinatorial optimization, resource allocation, pattern classification, data mining, and related areas.