Network Models with Convex Cost Structure like Bundle Methods

Christoph Helmberg[1]

Fakultät für Mathematik, Technische Universität Chemnitz
D-09107 Chemnitz, Germany
helmberg@mathematik.tu-chemnitz.de

Abstract. For three rather diverse applications (truck scheduling for inter warehouse logistics, university-course timetabling, operational train timetabling) that contain integer multi-commodity flow as a major modeling element we present a computational comparison between a bundle and a full linear programming (LP) approach for solving the basic relaxations. In all three cases computing the optimal solutions with LP standard solvers is computationally very time consuming if not impractical due to high memory consumption while bundle methods produce solutions of sufficient but low accuracy in acceptable time. The rounding heuristics generate comparable results for the exact and the approximate solutions, so this entails no loss in quality of the final solution. Furthermore, bundle methods facilitate the use of nonlinear convex cost functions. In practice this not only improves the quality of the relaxation but even seems to speed up convergence of the method.

Keywords. Lagrangian decomposition, large scale convex optimization, bundle methods, integer multi-commodity flow

1 Introduction

Integer multi-commodity flow problems arise naturally as modeling elements in many applications containing logistic aspects. In particular if these arise from time discretization the models are typically of very large scale. While the models lend themselves ideally to Lagrangian decomposition, they are mostly put into a standard linear programming solver as a whole because of convenience and because it is widely considered the only efficient environment for cutting plane approaches. The purpose of this paper is to point out that bundle methods with primal aggregation (like ConicBundle [1]) are a very attractive alternative in such cases, because they allow to use Lagrangian decomposition in combination with cutting plane approaches and even seem to profit in speed an quality from the use of more complex convex cost functions. Indeed, we present three real world applications where a bundle approach was able to generate reasonable starting points for rounding heuristics within several minutes while solving the full initial linear programming relaxation by a state of the art solver already took hours to days or failed due to memory requirements. In addition, the quality of

Dagstuhl Seminar Proceedings 09261
Models and Algorithms for Optimization in Logistics
http://drops.dagstuhl.de/opus/volltexte/2009/2190

the rounded solutions does not seem to depend on whether the exact solution of the relaxation or only a rough approximation to it is used for starting the rounding heuristics.

This extended abstract is structured as follows. Section 2 recapitulates the basic steps of a bundle method with primal aggregation and the use of cutting planes. In each of the following three sections we briefly describe a practical application, outline a model and give some numerical results. These are, in fact, short summaries of [2,3,4] and involve the work of many other people that I wish to acknowledge here with the affiliations of that time: F. Eckle (Fiege eCom), F. Fischer (TU Chemnitz), J. Janßen (Deutsche Bahn), B. Krostitz (Deutsche Bahn), A. Lau (TU Chemnitz), S. Röhl (ZIB Berlin), W. Rüstau (Fiege eCom).

2 Primal aggregation in bundle methods

The basic theory of bundle methods is explained in [5] or in more concise form in [6] and Lagrangian decomposition is a standard technique. Primal aggregation [7,8,9], however, and dynamic versions [10,11] (i.e., combined with cutting planes) are not yet common knowledge, so we sketch the main ideas shortly following the expositions in [2,10]. For the purpose of this presentation it is sufficient to consider Lagrangian relaxation of the linear constraints of the primal problem

$$\max\{c^\top x : Ax = b, x \in \mathcal{X}\}$$

where $\emptyset \neq \mathcal{X} \subset I\!R^n$ is a compact (*e.g.*, finite discrete or bounded polyhedral) set so that $\max\{\tilde{c}^\top x : x \in \mathcal{X}\}$ is efficiently solvable for any $\tilde{c} \in I\!R^n$ (like, *e.g.*, uncoupled minimum cost flow problem) and $A \in I\!R^{m\times n}$, $b \in I\!R^m$ describe additional linear constraints (*e.g.*, the coupling of the capacities in multi-commodity flow). Finding the best Lagrangian relaxation amounts to solving the (convex) dual problem

$$\min_{y\in I\!R^m} \psi(y) \quad \text{where} \quad \psi(y) := \max_{x\in\mathcal{X}} (c^\top x + (b - Ax)^\top y).$$

Given a starting point $y^0 = \hat{y}^0$, the bundle method iteratively determines the next candidate y^{k+1} as optimizer of a quadratic model,

$$y^{k+1} = \arg \min_{y\in I\!R^m} \max_{x\in\widehat{\mathcal{X}}^k} \left[c^\top x + (b - Ax)^\top y + \tfrac{1}{2}\|y - \hat{y}^k\|^2\right], \tag{1}$$

where $\hat{y}^k$ serves as current *center of stability* and the *model* $\widehat{\mathcal{X}}^k \subseteq \{x^0, \dots, x^k, \bar{x}^k\}$ is a subset of the previous optimal solutions $x^i \in \mathrm{Argmax}_{x\in\mathcal{X}}(c - A^T y^i)^\top x$ of the Lagrangian relaxations and a special *primal aggregate* $\bar{x}^k$ to be explained below. Next, in the oracle step, the Lagrangian relaxation $\psi(y^{k+1})$ is solved giving a corresponding primal optimizer x^{k+1}. If this value improves sufficiently upon $\psi(\hat{y}^k)$, the center is moved to this candidate, $\hat{y}^{k+1} = y^{k+1}$ (a *descent step*). Otherwise, a *null step* occurs, $\hat{y}^{k+1} = \hat{y}^k$, so the center is not changed but the next model $\widehat{\mathcal{X}}^{k+1}$ is improved in y^{k+1} by including the new x^{k+1}.

Instead of solving (1) directly, one may interchange min and max (strong duality holds by compactness if we switch to the convex hull of $\widehat{\mathcal{X}}^k$) and solve the new inner minimization over y explicitly giving $y(x) := \hat{y}^k - (b - Ax)$. By substituting this for y one obtains a primal convex quadratic subproblem in x equivalent to (1) and an optimal solution to this gives the next primal aggregate,

$$\bar{x}^{k+1} \in \text{Argmax}\{-\tfrac{1}{2}\|Ax - b\|^2 + (c - A^T\hat{y}^k)^\top x + b^\top \hat{y}^k : x \in \text{conv}\widehat{\mathcal{X}}^k\}. \quad (2)$$

Note, $\bar{x}^{k+1} \in \text{conv}\mathcal{X}$ because it is a convex combination of the points in $\widehat{\mathcal{X}}^k$, and the next candidate is $y^{k+1} = y(\bar{x}^{k+1})$. To guarantee convergence the next model only needs to satisfy $\{x^{k+1}, \bar{x}^{k+1}\} \subseteq \widehat{\mathcal{X}}^{k+1}$. If the convex hull of the primal feasible set is nonempty and compact and a dual optimal solution exists the method generates a subsequence y^k, $k \in K$, of dual feasible points that converges to an optimal y^* and satisfies $\|y^k - \hat{y}^{k-1}\| \to 0$ and $c^\top \bar{x}^k + (b - A\bar{x}^k)^\top y^k \to \psi(y^*)$ (see, e.g., [12]; $K = \{k : \hat{y}^k \neq \hat{y}^{k-1}\}$ in the case of infinitely many descent steps, otherwise, ignoring the finite case, $K = I\!N$). On this subsequence $b - A\bar{x}^k \to 0$ and $c^\top \bar{x}^k \to \psi(y^*)$, so any cluster point x^* of the $\bar{x}^k$, $k \in K$, satisfies $x^* \in \{x \in \text{conv}\mathcal{X} : Ax = b\}$ and $c^\top \bar{x}^* = \psi(y^*)$ and is therefore an optimal solution to the relaxation.

Thus, the primal aggregates $\bar{x}^k$ (2) serve as successively better approximations to the primal optimal solution of the relaxation and as the basis for rounding heuristics or for cutting plane approaches. When adding constraints violated by $\bar{x}^k$ and reoptimizing, the following two issues have to be considered. The dimension of the dual problem changes on the fly and the next aggregate $\bar{x}^{k+1}$ will, in general, still violate the newly added inequalities. However, as this violation goes to zero over time, both aspects can be taken care of by dynamic bundle methods [10,11] if an appropriate violation measure is used in the separation procedure.

In all our experiments we use the ConicBundle callable library [1]. It is designed for Lagrangian relaxation and supports primal aggregation, primal cutting planes as well as the use of independent bundle models for sums of convex functions. Like for subgradient algorithms, the user only has to provide an oracle implementing the evaluation of $\psi(y)$, *i.e.*, a routine that finds $x^* \in \text{Argmax}_{x\in\mathcal{X}}(c - A^T y)^\top x$ for given y and returns the value $(c^\top x^*) + (b - Ax^*)^\top y$, the primal violation (subgradient) $b - Ax^*$, and x^* itself for primal aggregation. In principle, primal aggregation is also possible for pure subgradient algorithms [9], but while requiring exactly the same oracle data, the quadratic term in (2) enhances convergence towards feasibility as well as the stability of the optimization process. We illustrate the advantages of the bundle approach in comparison to state of the art linear programming solvers on three applications from practice.

3 Truck scheduling for inter warehouse logistics [2]

Problem Description. A company operates several warehouses within the same city. In each warehouse articles in stock are commissioned into shipments but part of the stock may currently be located in other warehouses. In order to complete the shipments within one working day several trucks transfer pallets between the warehouses. The total transportation time of a pallet between automatic storage systems of different warehouses is roughly six hours (with up to one hour actual driving time on the truck), each pallet carries a standard number of items of only one type of article, and a truck can carry up to 27 pallets in one ride. The task is to determine, within roughly ten minutes, a schedule of the truck rides for the next 4 hours together with their loads of pallets so that with good probability all current and future shipments can be completed in each warehouse on time. Our scenarios from practice included two and three warehouse, roughly 1000 pallets were transferred each day, and, based on current demand for ordered shipments and a stochastic model of future demand, one or several pallets of roughly 1000 different articles of a total of 40000 had to be considered for transfer.

The model. For use in a rolling horizon technique the planning period is the full next day based on a time discretization of ten minutes. For each article a separate network represents the flow of the pallets of this article between the warehouses (staying in the warehouse, getting ready for transfer, and the transfer itself). For each truck class (*i.e.*, trucks that need not be discerned in capacity) a separate network models the actual loading, transport ride, unloading, waiting, and empty transfers of the trucks. A flow of one along a transport ride in a truck graph raises the joint capacity of the corresponding transfer arcs in the article graphs by the truck capacity. Further coupling capacities are induced by the storage capacity of each warehouse and the number and capacity of the loading and unloading platforms in each warehouse. The selection of appropriate pallets is guided by a convex, piecewise linear cost function (a separate one for each warehouse and time-step in each article graph) that maps the probability of a shortage on account of current stock of this article to a penalty value.

Results. The instances stem from half a year of real world data for two warehouses (2-WH in Table 1) and are also mapped to three warehouses (3-WH). Computing schedules at 6:00, 9:00, 12:00, and 15:00 hours each day yields a total of 942 instances ranging between of 200,000 to 1,100,000 arcs and 1500 to 4500 multipliers (in the case of 2-WH). Table 1 is taken from [2] and compares

Table 1. Average and deviation of running time and relative precision for truck scheduling instances

		LP by simplex		bundle		
scenario	#inst.	time(sec)	heur-gap(%)	time(sec)	relax-gap(%)	heur-gap(%)
2-WH	942	2138 (1445)	5.29 (5.90)	229 (95)	3.37 (3.44)	5.75 (6.13)
3-WH	942	6629 (5724)	19.4 (12.8)	312 (129)	6.45 (4.34)	18.8 (11.7)

the dual simplex code of CPLEX 9.1 [13] to stopping after at most 2000 oracle evaluations of a ConicBundle implementation using MCF [14] for computing the minimum cost flow subproblems, where

$$\text{relax-gap} = 100 \cdot \left(1 - \frac{\text{bundle-sol.}}{\text{LP-sol.}}\right), \quad \text{heur-gap} = 100 \cdot \left(1 - \frac{\text{LP-sol.}}{\text{heuristic-sol.}}\right).$$

Computation times refer to a Linux PC with Pentium 4, 3.2 GHz processor, 1 GByte RAM, and 1 MByte level 2 cache. Some preliminary experiments with CPLEX 12.1 suggest that the new version is faster by a factor of two, so the bundle approach is clearly faster also today. Note, there is no significant difference in the quality of solutions after rounding heuristics have been applied.

4 University-course timetabling [3]

Problem description. Setting and results are based on the current situation at TU Chemnitz and have been worked out in the diploma thesis of Anja Lau [3]. The basic requirements of university-course timetabling are quite the same in most universities, so we will only give the two special features arising at TU Chemnitz. First, the campus is distributed over four distinct locations within the city so that transfer times pose a significant constraint in setting up timetables. Second, the study programs at TU Chemnitz include a significant number of semiobligatory and optional courses, so the goal is to find a timetable that keeps many options open to the students.

The Model. For each distinct study group (*i.e.*, for each semester of each branch of a study program) and each day of the week a separate network represents the flow of the students through the courses at the various locations and time slots (one arc per slot for obligatory courses, one for each semiobligatory and optional course, arcs for transferring, for waiting, for going to no course at all, . . .). Course arcs are opened by separate assignment variables, assigning for each course a time slot on some day at some location (the room problem is skipped, similar in style to [15]. A convex piecewise linear reward structure motivates students to visit as many semiobligatory and optional courses as possible so that in the case of several available options (and in lack of better data) they try to split up so that each course is visited by at least 10% of the study group. But of course they still prefer fewer waiting periods and transfers between locations. Further convex piecewise linear terms penalize deviations from the average daily course load of each study group.

Results. Instances are constructed by taking various subsets of increasing size (A to E in Table 2) of study programs at TU Chemnitz for the winter term 2007/08 and the summer term 2008. The number of primal variables ranges from roughly 200,000 for A to 1,300,000 for E. Table 2 compares the computation time for solving the relaxations with CPLEX 9.1 barrier [13] (simplex variants performed abysmal here) and with a ConicBundle implementation using MCF [14] for computing the minimum cost flow subproblems. All times are in minutes and refer to a computer with Intel Xeon Dual Core 3.0 GHz processor with 64 GB RAM. Again, an acceptable approximate solution is quickly computed by the bundle approach. Note, the differences in quality of the solutions produced

Table 2. running time and relative precision for university-course timetabling

		LP by barrier		bundle		
scenario	term	time(min)	heur-gap(%)	time(min)	relax-gap(%)	heur-gap(%)
A	W07/08	6.1	3.0	0.4	4.3	3.8
	S08	4.9	2.3	0.3	8.2	3.1
B	W07/08	10.1	6.3	0.7	3.6	7.8
	S08	7.2	5.8	0.6	3.0	6.0
C	W07/08	77.4	7.1	2.1	2.5	7.2
	S08	176.5	5.2	2.5	2.5	6.1
D	W07/08	> 720	-	3.5	-	-
	S08	169.5	5.4	3.6	2.4	5.7
E	W07/08	572.8	6.8	6.8	1.6	7.8
	S08	258.6	4.8	7.0	1.5	4.8

by the rounding heuristic slightly favor the exact relaxation solution as a starting point. These differences, however, are of little relevance considering the many uncertainties in the data concerning the expected number of students in the audience.

5 Operational train timetabling [4]

Problem description. For long term simulation studies concerning network capacities, Deutsche Bahn is interested in the following setting. Given passenger and freight trains with fixed routes and desired stopping intervals at selected stations along their routes, find an operational timetable for all trains so that actual train stops satisfy the interval constraints and technical restrictions like stopping dependent running times, headway times and station capacities are observed. Note, neither is any connection data available nor is periodicity required other than specified by the stopping intervals.

The Model. Time is discretized into steps of one minute. For each train the progress along its route is represented by a time discretized network that encodes stopping dependent running times and can be solved as a shortest path problem. The capacities of the arcs are coupled via station capacities and headway time restrictions. In the computations reported in [4] both constraint types are implemented in a cutting plane approach. While separation of the station capacities is trivial, separation of headway conflicts is implemented by a heuristic searching for maximal cliques in a conflict graph (in contrast to [16], exact separation is impractical here).

Results. Instances are based on a real timetable over a time span of six hours for south-western Germany. Instance 1 includes only those trains that run over the five most frequently used track segments (242 Passenger trains, 9 freight trains, 317336 variables), instance 2 is the main traffic route along the river Rhine (50 passenger and 67 freight trains, 2,448,842 variables), and instance 3 considers the entire subnet (2501 passenger, 659 freight trains, 8,990,060 variables).

Table 3. running times for operational train timetabling

Instance	CPLEX	ConicBundle
1	33s	12s
2	1945s	341s
3	–	2512s

Table 3 compares the running times of the cutting plane algorithms implemented with CPLEX 9.1 [13] and with ConicBundle, where both use the same basic relaxation and separation procedures. Times refer to an Intel Xeon Dual Core 3 GHz computer with 16 GB RAM, the largest problem could not be finished by CPLEX due to insufficient memory. For both approaches the development of the bound over time is illustrated in Figure 1 for instance 2. Even considering the fact that the code was carefully tuned for ConicBundle while CPLEX is used in default setting, the differences in running time are surprising as cutting plane applications are considered a classical domain of the simplex algorithm.

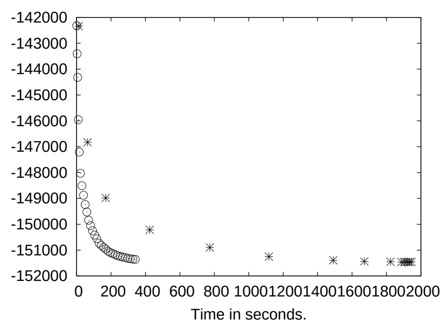

Fig. 1. Development of the bound for instance 2: $*$ CPLEX, $\circ$ ConicBundle

References

1. Helmberg, C.: ConicBundle 0.3. Fakultät für Mathematik, Technische Universität Chemnitz. (2009) http://www.tu-chemnitz.de/~helmberg/ConicBundle.
2. Helmberg, C., Röhl, S.: A case study of joint online truck scheduling and inventory management for multiple warehouses. Operations Research **55** (2007) 733–752
3. Lau, A.: Erstellen von wegeoptimierten Stundenplänen mit Diskreten Methoden. Master's thesis, Technische Universität Chemintz, Fakultät für Mathematik (2008) In German, http://www.tu-chemnitz.de/mathematik/discrete/diplom/anja_lau.pdf.

4. Fischer, F., Helmberg, C., Janßen, J., Krostitz, B.: Towards solving very large scale train timetabling problems by Lagrangian relaxation. In Fischetti, M., Widmayer, P., eds.: ATMOS 2008 - 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems, Dagstuhl, Germany, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2008)
5. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms II. Volume 306 of Grundlehren der mathematischen Wissenschaften. Springer, Berlin, Heidelberg (1993)
6. Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizaábal, C.A.: Numerical Optimization. Springer (2003)
7. Feltenmark, S., Kiwiel, K.C.: Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems. SIAM J. Optim. **10** (2000) 697–721
8. Helmberg, C., Rendl, F.: A spectral bundle method for semidefinite programming. SIAM J. Optim. **10** (2000) 673–696
9. Anstreicher, K.M., Wolsey, L.A.: Two "well-known" properties of subgradient optimization. Math. Programming **120** (2009) 213–220
10. Helmberg, C.: A cutting plane algorithm for large scale semidefinite relaxations. In Grötschel, M., ed.: The Sharpest Cut. MPS-SIAM Series on Optimization. SIAM/MPS (2004) 233–256
11. Belloni, A., Sagastizábal, C.: Dynamic bundle methods. Math. Programming **120** (2009) 289–311
12. Helmberg, C., Kiwiel, K.C.: A spectral bundle method with bounds. Math. Programming **93** (2002) 173–194
13. ILOG S.A. 9 Rue de Verdun, 94253 Gentilly Cedex, France: ILOG AMPL CPLEX System, Version 9.1, User's Guide. (2005) Information available at `http://www.ilog.com`.
14. Löbel, A.: MCF Version 1.2 – A network simplex Implementation. Konrad-Zuse-Zentrum für Informationstechnik Berlin. (2000) Available at http://www.zib.de/Optimization/Software/Mcf (free of charge for academic use).
15. Lach, G., Lübbecke, M.E.: Optimal university course timetables and the partial transversal polytope. In: Experimental Algorithms. Volume 5038/2008 of Lecture Notes in Computer Science., Springer (2008) 235–248 7th International Workshop, WEA 2008 Provincetown, MA, USA, May 30-June 1, 2008 Proceedings.
16. Lukac, S.: Holes, antiholes and maximal cliques in a railway model for a single track. ZIB-Report ZR 04-18, Konrad-Zuse-Institut für Informationstechnik Berlin, Takustrae 7, 14195 Berlin, Germany (2004)